

Enhanced Simulation-Based Verification and Validation of Automotive Electronic Control Units

Thomas Herpel, Thomas Hoiss and Jan Schroeder

Abstract Modern cars comprise of increasingly complex electronic devices. Especially for electronic control units (ECUs) in safety-critical application areas, testing must be effective and efficient in the potentially conflicting area of increasing system complexity and shorter development cycles. As a significant enhancement to well-known development efforts according to the V-Model, the use of ECU simulation models at the early stages of development were proposed. These models closely resembled the behavior and functionality of a real ECU in a state-based implementation compliant to modeling standards like UML (Unified Modeling Language). The approach was seamlessly embedded in automotive hardware in loop-based ECU integration and system testing, which allowed for a thorough verification and validation of basic system requirements and test case implementations.

Keywords Automotive electronics · Verification and validation · Simulation · System modeling · State charts · Hardware-in-the-Loop testing

1 Introduction

Modern luxury vehicles are equipped with up to one hundred electronic control units (ECUs) serving various purposes like driver assistance, occupant protection, and on-board entertainment. Thus, the more software functions implemented in these

T. Herpel (✉) · T. Hoiss
Automotive Safety Technologies, Gaimersheim, Germany
e-mail: thomas.herpel@astech-auto.de

T. Hoiss
e-mail: thomas.hoiss@astech-auto.de

J. Schroeder
Department of Computer Science and Engineering,
University of Gothenburg, Gothenburg, Sweden
e-mail: jan.schroder@gu.se

ECUs, the more complex the hardware becomes. Furthermore, the broad variety of vehicle options, shorter production cycles, and faster times-to-market yield challenges during the development of the automotive electronic systems. Especially for safety-critical systems, like airbag control or emergency braking systems, safe and reliable development must be assured without compromise. Electronic development in the automotive domain typically follows a process depicted in the “V-Model”, as shown in Fig. 1, starting from required specifications (upper left) to hardware/software implementation (lower tip) to final application tests (upper right). The key advantage of this developmental process model is a direct connection of each development phase on the left hand side to its counterpart validation activity at the corresponding layer on the right hand side. However, this means that all development and test activities have to be performed consecutively, one after the other. This can sometimes lead to significant delays when tests at a certain level are performed and systematic faults, if they exist, are discovered. To overcome this, this study proposed that the information encoded in the system requirements be used to derive simulation models of the under-development ECUs at the very early stages. The proposed simulation component is called “Software-ECU” (Soft-ECU) and resembles the functional behavior of a real ECU according to the system architecture, e.g. controller area network (CAN), FlexRay bus communication, failure handling, fault tolerance, and diagnosis. However, non-functional aspects, such as memory consumption, should still be validated using real ECU hardware samples in order to obtain realistic and unbiased results. By applying a suitable simulation environment, the Soft-ECU can be exposed to test cases that are derived from the same system requirements for later use in the verification and validation process steps. Hence, effectiveness and efficiency in ECU development are enhanced on two fronts. First, test cases can reveal basic systematic faults at the very beginning of hardware development without the need for a physically present hardware sample. Second, test

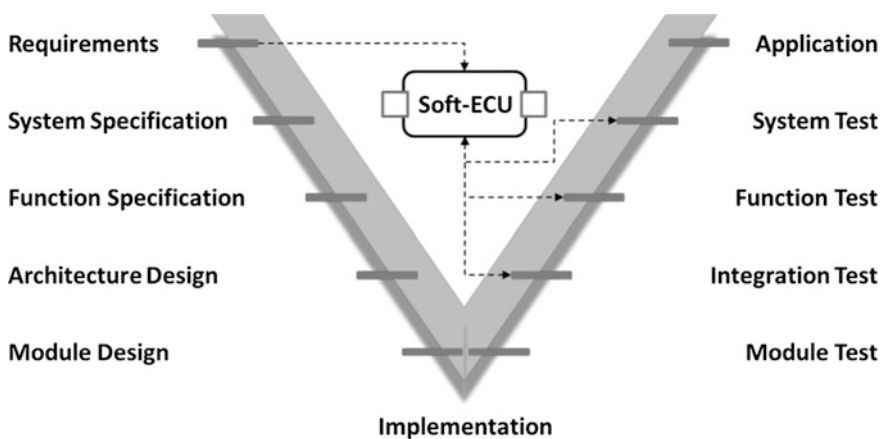


Fig. 1 Electronics development according to V-Model with use of Soft-ECU

case validation is achieved without obstructing test system resources like hardware-in-the-loop simulators (HiL), i.e. the correct implementation of the system tests can be validated. In Fig. 1, these two aspects are resembled by the block “Soft-ECU” in the middle portion connected to both the requirements on the left and the different test activities on the right.

Because design and production of the first prototype hardware samples and the validation of up to a few thousand test cases are typically very time-consuming during developmental stages, significant benefits at the early stages are expected by front-loading these activities.

This paper is organized as follows: Sect. 2 presents related work in the said area of research. In Sect. 3, the most common testing approaches for automotive electronics are presented. The proposed process enhancements are introduced in Sect. 4, together with some first analytical results from the Soft-ECU modeling in a real vehicle development project. Finally, Sect. 5 summarizes the paper and gives an outlook on future work.

2 Related Work

Herpel et al. [1] have shown how to improve ECU testing by applying combined electric/electronic and functional HiL simulator testing. Challenges in integration testing of automotive electronics were identified and strategies for sound validation and verification were proposed by Schroeder et al. [2]. Suh et al. and Schuette et al. [3, 4] proposed a HiL- or PC-based simulation environment for hardware testing, while requiring a real hardware sample for validation purposes. Himmler [5] introduced virtualization in ECU testing from the field of aircraft system development. Model-based development of ECU software and validation were employed by Vora et al. [6] at various stages of the development of powertrain control strategies. Witter et al. [7] combined a HiL test of ECUs with a virtual environment for the generation of driving scenarios, which required a real ECU. Caraceni et al. [8] proposed to deploy a real-time model of an engine ECU in software and HiL tests to shift as much of the development as possible to off-vehicle testing. However, for HiL testing, a prototype sample of the ECU is needed. Andrianarison and Piques [9] employed the systems modeling language (SysML) for modeling the system capabilities and system interaction on various levels of architecture.

3 Testing Approaches for Automotive Electronics

In order to put safe and reliable systems to market, thorough testing activities for both hardware and software are performed during automotive ECU development. From the system requirements, which encode the functional and non-functional

specifications and are typically denoted in a semi-formal or formal manner, test cases are derived. Depending on system complexity, this yields up to thousands of test cases to cover all the specification aspects. The V-Model encodes a stepwise testing process for various consecutive stages, which is well known and commonly applied with respect to developmental standards like ISO 26262 [10].

3.1 Module Tests

A module test is the very first testing activity after implementation of ECU software. It is performed either as model-in-the-loop (MiL), where interfaces of model-based ECU software components are stimulated to validate correct implementation of requirements, or as software-in-the-loop (SiL), where ECU software is compiled according to the target hardware platform, and MiL test cases are applied to the compilation for purposes of back-to-back testing. Additionally, processor-in-the-loop (PiL) testing aims to assess the compiled software integrated into the target hardware platform.

All approaches—MiL, SiL, and PiL—serve to validate fulfillment of the functional requirements. While MiL and SiL testing are typically performed on a desktop PC and cover no ECU hardware aspects like memory consumption or runtimes, PiL testing is carried out on a target hardware platform processor boards.

3.2 Integration Tests

The aim of integration testing is to prove successful hardware and software integration for an ECU sample. To this end, the ECU sample is connected to a HiL simulator, which is capable of stimulating the ECU and tracking the corresponding reaction. It is typically a black-box test, where automated test cases trigger the HiL simulator. The focus is on electrical and electronic aspects, like bus communication, diagnosis data or handling of peaks, leakages, or toggling of the power supply [1].

3.3 Function Tests

A proposal for effective validation of safety-critical functionality of an airbag ECU was shown by Herpel et al. [1] using an enhanced HiL simulator, which allows for direct stimulation of the ECU-internal sensor values. In general, functional testing aims to estimate the functional performance of an ECU in terms of both correct and on-time reaction to a given set of input values.

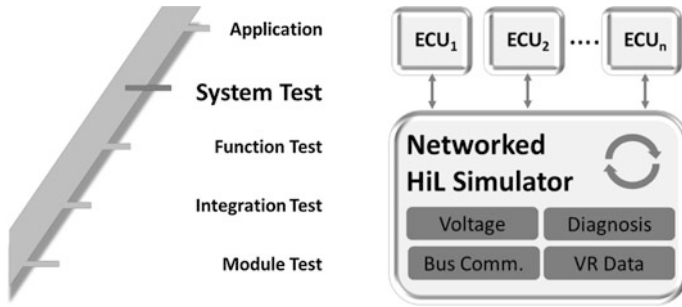


Fig. 2 Networked HiL simulator at ECU system test level

3.4 System Tests

System tests focus on the validation of plausible behavior of an ECU when interacting with sensors, other ECUs, or actuators inside a car. A suitable test system for that purpose is a networked HiL simulator (Net-HiL). As depicted in Fig. 2, the Net-HiL can connect all ECUs in a car and control data exchange on the communication bus level. Hence, each connected ECU is connected to the detailed in-car electronic infrastructure and performs on the level of vehicle function.

3.5 Application

Application is the final step of the verification and validation. ECUs in a series-production status are integrated into a real car and tested on the vehicle function level in dedicated test campaigns or driving maneuvers on proving grounds. With respect to the observed behavior, ECU function parameters can be applied towards an optimal system performance. Of course, these testing activities are both costly and time consuming, and reproducibility is difficult, especially for complex driving scenarios.

4 Test Process Enhancements

ECU testing depends heavily on the availability of real ECU hardware components. Any delay in hardware development inevitably causes delays in test system operations. On one hand, test cases cannot be validated with respect to correctness, coverage, plausibility, or completeness. This holds for testing activities at all stages of verification and validation. On the other hand, system testing with Net-HiL simulators suffers from late hardware availability, as joint in-car functionality relying on data from several ECUs cannot be validated in cases where single ECUs

are not available. Actually, it can be assumed that test case validation must not be done as part of a test campaign, since valuable time of test system availability can be used more effectively once real ECU hardware is available. Likewise, the Net-HiL must not be kept in a waiting mode until a real hardware sample has been delivered for each connected ECU.

4.1 Simulation Approach

It is suggested that Soft-ECUs be employed for validation in order to decrease idle-times of test systems, which will increase test system availability for actual testing.

Soft-ECU: A Soft-ECU mimics a real ECU as a software component. Interfaces, structure, and functionality are derived from system requirements, allowing for the simulation of both physical and functional behaviors. Examples of ECU requirements to be captured by a Soft-ECU are “Message XY is sent to FlexRay every 10 ms” or “Detection of a voltage drop is written to error memory.” For implementation of a Soft-ECU with communication ports, internal states, and function blocks, as depicted in Fig. 3, a model-based approach was proposed using Matlab and Simulink.

Simulation Framework: Soft-ECUs cannot be operated alone, which means a higher-level entity is required for timing, data generation, and scheduling, as shown in Fig. 4. Simulation must either be capable of test case validation by the application of HiL test cases to Soft-ECUs, or of enabling system testing by integrating Soft-ECUs in a given hardware environment with the Net-HiL.

4.2 Concept for Test Case Validation

HiL test cases for ECUs are derived from the ECU requirement specifications. To achieve a high degree of automation, various modeling paradigms are available in

Fig. 3 Structural model of a Soft-ECU

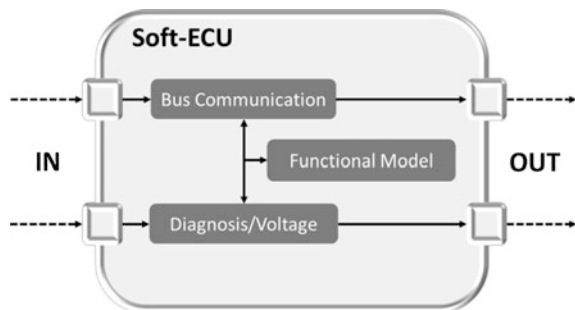
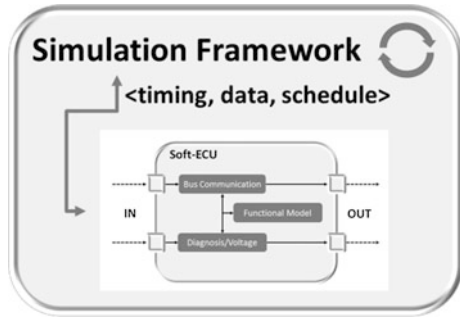


Fig. 4 Simulation framework for Soft-ECU operation



order to obtain a machine-readable test case implementation, e.g. based on Python scripts, sequence diagrams, and state charts according to the unified modeling language (UML, <http://www.uml.org/>). The set of test cases has to be validated with respect to completeness and correct implementation of the test goals as defined by the requirements. Instead of using valuable resources of the HiL simulator for that purpose, an “offline validation” approach is proposed, which employs the capabilities of the Soft-ECU. This process is depicted in Fig. 5.

Both test cases and Soft-ECUs are derived from ECU requirements. Applying the test cases to the Soft-ECU in the simulation framework allows for the validation of the correctness of test case implementation such as for time outs, system values, parameters, or error-handling strategies. The scope of the test cases applicable to the Soft-ECU depends on the level of detail in the Soft-ECU model. It can range from simple request/response state charts in bus communication to complex physical models that resemble thermal behaviors of real ECUs. To avoid obstructing the use

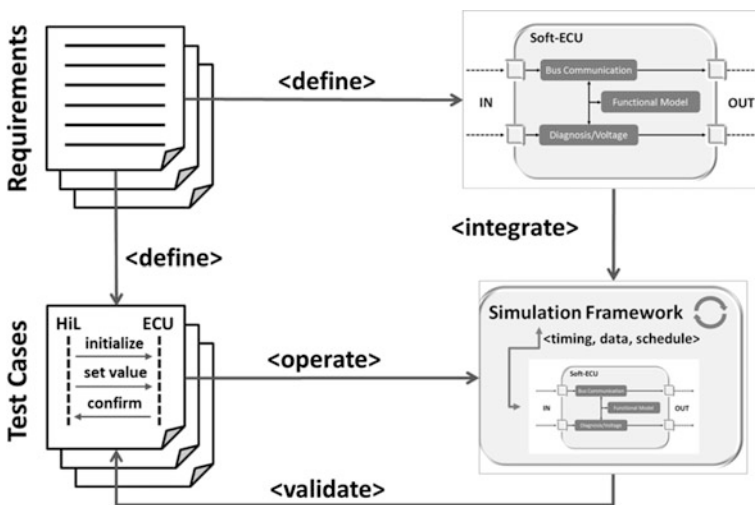


Fig. 5 Process of test case implementation and validation using Soft-ECU

of the HiL simulator for test case validation, a HiL-independent implementation for the simulation framework is currently being developed in a PhD project. Initial requirements and markets analyses have shown that a framework based on SysML appears to be appropriate due to standardized support of state-based artefacts and possible enhancements with respect to timing and scheduling towards real-time testing via profiles like MARTE (<http://www.omgmarTE.org/>).

4.3 System Test with Networked HiL Simulator

Soft-ECUs can be employed with the Net-HiL for validation on system test levels at early stages of vehicle development, where ECUs are typically not yet available as real hardware samples. Each non-present ECU can be replaced by its counterpart Soft-ECU in order to complement the set of electronic peripherals of the Net-HiL and to enable a meaningful HiL test operation. This is illustrated in Fig. 6 for both early and late stages of vehicle development, with ECUs and Soft-ECUs shown grayed or highlighted according to availability and utilization, respectively. Since a Soft-ECU is capable of resembling functional behavior on a bus communication level, it is particularly suited to bridge the potential hardware gap at the Net-HiL.

According to the experiences of this study, control software packages of modern HiL simulators, such as ControlDesk (<http://www.dspace.com/>) or LabCar (<http://www.etas.com/>), show a high degree of compatibility with the Soft-ECUs modeled in Matlab and Simulink, and allow for a straightforward integration of Soft-ECUs into the simulator environment and test automation. Hence, HiL control software intrinsically serves as a simulation framework for Soft-ECUs, with no external standalone simulation framework at the Net-HiL required as in test case validation.

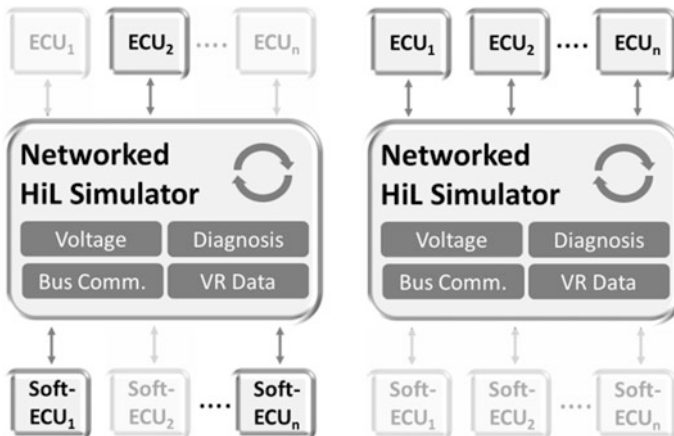


Fig. 6 ECU integration testing at networked HiL simulator using Soft-ECUs at early stage in development (*left*) and at late stage (*right*)

4.4 Analytical Results for Soft-ECU Development

During prototype ECU modeling activities for operation with a Net-HiL in a vehicle development project, the results of Soft-ECU modeling and model behavior along several revision cycles of Soft-ECUs implemented in Matlab and Simulink were determined, including the following performance measures:

- Block Count: Number of sub-blocks inside a Soft-ECU
- Structural Complexity (S): Connections to other Soft-ECUs (Coupling)
- Local Complexity (L): Work effort within one Soft-ECU (Cohesion)

The number of blocks inside a model was derived using a Matlab internal function for block counting. Structural complexity S and local complexity L were calculated according to the metrics following the approach as proposed by Card et al. [11] in Eqs. 1 and 2, where f_i remarks the fan out of module i , n is the number of modules on one layer and v_i are the I/O (input/output) variables of module i :

$$S = \frac{\sum f_i^2}{n} \tag{1}$$

$$L = \frac{\sum \frac{v_i}{f_i+1}}{n} \tag{2}$$

Figures 7 and 8 illustrate these measures for two exemplary Soft-ECUs that were modeled for a real electronic stability control ECU (ESC) and a real engine ECU (Engine), respectively. The plots resembled modeling activities along 50 model

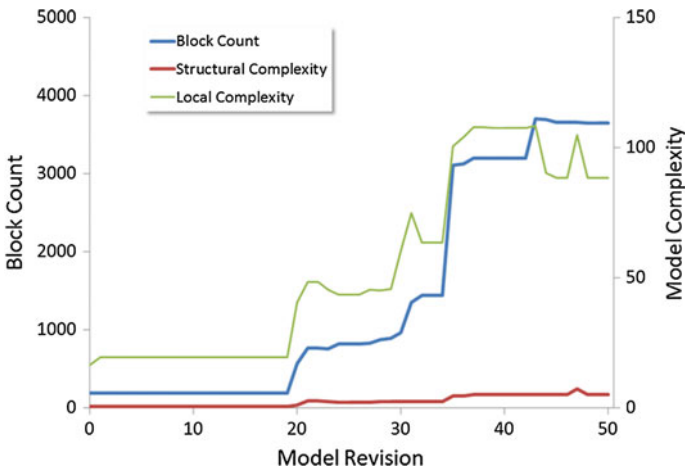


Fig. 7 Model size and complexity measures for ESC Soft-ECU along model revisions during vehicle development

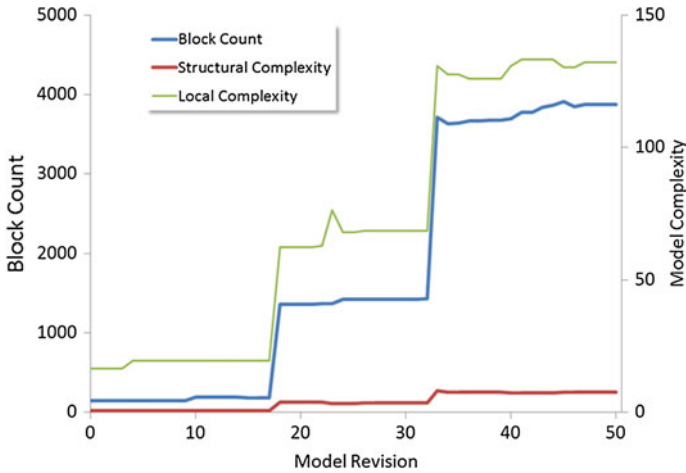


Fig. 8 Model size and complexity measures for engine Soft-ECU along model revisions during vehicle development

revision cycles and allow for an at-a-glance estimation of quantitative and qualitative parameters.

Obviously, both the extent and complexity of the models continuously increase during vehicle development, as more and more ECU system requirements are implemented in Soft-ECUs. Consequently, increased sizes yield increased requirements coverage with respect to the ECU system specifications, and increasing model complexity resembles the step-by-step extended functional scope and performance of respective Soft-ECUs. If the complexity measures exceed certain well-defined thresholds, a potential refactoring or redesign of a Soft-ECU can be initiated.

5 Conclusions and Outlook

This study proposed a simulation-based improvement for ECU testing during the early stages of vehicle development. With a model-based simulation component called Soft-ECU, the gap between rising electronics complexity and tight time schedules for verification and validation can be bridged with more effective and efficient HiL testing. Soft-ECUs were implemented in Matlab and Simulink for real ESC and engine ECUs, and quantitative and qualitative performance measures for model size and complexity were derived and presented. For test case validation, a simulator-independent simulation framework is being developed, in which a SysML-based approach shows potential for testing conformity to modeling standards and real-time capabilities. Consequently, the future work on this topic aims to accomplish implementation of the standalone simulation framework and integrate

the Soft-ECU-based validation of test cases into existing test processes. In addition, metrics, which allow for the numbering of test effort improvements, will be defined in a detailed case-study for the development of an airbag ECU. It is believed that virtualization is an indispensable part of verification and validation in the development of sound and secure automotive electronics.

References

1. Herpel, T., Hoiss, T., Sghair, A.: Hardware-in-the-loop-based systematic testing of automotive safety electronics. In: Proceedings of 4th International Conference on Electronics, Communications and Networks (CECNET2014), pp. 1425–1429. CRC Press, Boca Raton (2015)
2. Schroeder, J., Berger, C., Herpel, T.: Challenges from integration testing using interconnected hardware-in-the-loop test rigs at an automotive OEM. In: Presented at 1st International Workshop on Automotive Software Architectures, pp. 39–42, Montreal, Canada (2015)
3. Suh, M.W., Chung, J.H., Seok, C.S., Kim, Y.J.: Hardware-in-the-loop simulation for ABS based on PC. *Int. J. Veh. Des.* **24**(1), 157–170 (2000)
4. Schuette, H., Waeltermann, P.: Hardware-in-the-loop testing of vehicle dynamics controllers—a technical survey. Technical Report, SAE Technical Paper, 2005-01-1660 (2005)
5. Himmler, A.: From virtual testing to HIL testing—towards seamless testing. Technical Report. SAE Technical Paper, 2014-01-2165 (2014)
6. Vora, A., Wu, H., Wang, C., Qian, Y.: Development of a SIL, HIL and vehicle test-bench for model-based design and validation of hybrid powertrain control strategies. Technical Report, SAE Technical Paper, 2014-01-1906 (2014)
7. Witter, H., Heiden, M., Talwar, K.: ABS/ESP ECU testing with sophisticated HIL simulation methods. Technical Report, SAE Technical Paper, 2009-26-0079 (2009)
8. Caraceni, A., De Cristofaro, F., Ferrara, F. et al.: Benefits of using a real-time engine model during engine ECU development. Technical Report, SAE Technical Paper, 2003-01-1049 (2003)
9. Andrianarison, E., Piques, J.-D.: SysML for embedded automotive systems: a practical approach. In: 2010 Embedded Real Time Software and Systems, ERTS. http://web1.see.asso.fr/erts2010/Site/0ANDGY78/Fichier/PAPIERS%20ERTS%202010/ERTS2010_0057_final.pdf (2010)
10. International Organization for Standardization, ISO 26262: Road Vehicles—Functional Safety, October 2015
11. Card, D., Agresti, W.: Measuring software design complexity. *J. Syst. Softw.* **8**(3), 185–197 (1988)