Submission for:
O Software Quality Management (SQM) **primarily in German**
X Software Testing (ICSTEST) **in English**
O Software Validation for Healthcare (CSVHC) **in English**

Title of Paper

Automated Closed-Loop Testing of Embedded Engine Control Software

Presenter

Presenter: Dipl.-Ing. Sven Rebeschieß, IAV GmbH Berlin, Germany

Co-Presenter: Dipl.-Ing. Thomas Liebezeit, TU Berlin / MDT, Germany

Further Authors: Dipl.-Ing. Uzmee Bazarsuren, TU Berlin / MDT, Germany

Session Format

O Workshop **O** Tutorial **X** Conference Presentation

Instructional Level

O Introductory **X** Intermediate **O** Advanced

Target Group

Test Practitioners and Engineers, Software and Test Managers, QA Managers, Design and Development Engineers

Keywords

- Software-in-the-Loop (SiL)
- Test Workflow
- Automated Test
- Embedded Software

Tools

- MATLAB/Simulink
- TestDirector, Testmaster
- StarTeam, Subversion

Period of the SiLEST joint project: March 2004 – Dec. 2006

Abstract

1. Introduction

The workload and cost of validating software systems embedded in a technical environment (embedded software) often account for as much as 50% of the total development budget. Yet despite this expenditure imperfections in mechatronic systems of this type cannot be ruled out completely, particularly, for example, in cases where age-related wear alters the hardware environment (confer the rising number of recalls in the automotive sector). Improving the process for testing the software of embedded systems, therefore, not only permits high potential savings in terms of development costs, it also enhances product quality.
Part of the SiLEST (Software in the Loop for Embedded Software Test) joint project looks at the potential of testing the software of embedded systems within a simulated environment. Attention focuses on identifying the possible applications and limits of the software-in-the-loop (SiL) test method with the aim of conducting “in the loop” simulation at an early stage on a cost-effective, reproducible, retraceable and automatic basis. This process is also compared with the established hardware-in-the-loop (HiL) test method in order to reveal and contrast the benefits and drawbacks associated with both methods.
This paper describes the SiL test environment, the SiL test model as well as the automated SiL test itself. These are the main constituents of the test workflow defined within the SiLEST project for the automotive sector.

2. SiL test environment

The SiL test environment is made up of the test management system, the test execution system and the version management system: from the **test management system** comes the instruction to conduct new tests. It also provides access to individual tests and their results. It thus answers the question of who tested what, when and with which result. The **test execution system** comprises the test sequence controller, which permits testing automation, and the test model. Automatic test workflow control ensures adherence to a defined test process. The **version management system** is responsible for versioning all files used within the test workflow in order to ensure comprehensive reproducibility.
The test system defined in SiLEST integrates standard tools. In the automotive industry, for example, TestDirector, Matlab/Simulink and StarTeam are used. Our project partner DLR (German Aerospace Center) employs Testmaster, Matlab/Simulink and Subversion. The open interface concept facilitates the incorporation of further tools.

3. SiL test model

The SiL test examines the behavior of the engine control unit (ECU) software that is implemented. Using a test model, the ECU code undergoing validation is integrated in a closed-loop model made up of ECU and environment. As a result, the algorithms under test take up information from their environment that they themselves then influence. The test model presented in Fig. 1 comprises:

- the **ECU model** with engine control algorithms (e.g. idle speed control or two-stage supercharging that are provided in the form of an executable specification and/or ECU code), ECU characteristics (e.g. scheduler) and interfaces to the environment (e.g. I/O) as well as
- the **environment model** itself with models for vehicle (e.g. engine, transmission), driver, ambient conditions (e.g. road surface, incline, wind), sensors (S) and actuators (A).

The SiLEST test model is distinguished by the following three particular aspects:

- The model incorporates the algorithms under test in two different ways: firstly through the executable specification, and secondly through its actual implementation, i.e. the ECU code.
- The ECU, including its I/O interfaces, is simulated in model form.
- Sensors and actuators exhibiting nominal as well as off-nominal behavior are also modeled.

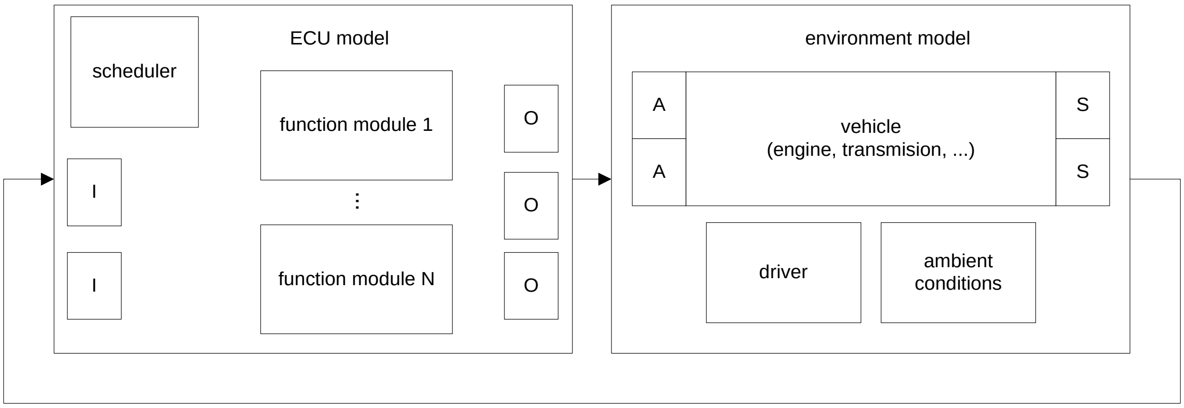

Fig. 1: SiL test model

4. SiL test

The aim of the automated SiL tests is to obtain early validation of software components that are already integrated. This assumes that, unlike subsequent hardware-in-the-loop testing, it is not yet possible to draw on any fully implemented functionality. Simulation, therefore, only takes place with the components already coded in conjunction with the executable specification and an environment model. This way, it is possible at an early stage in the development process to carry out automated regression tests in which the behavior of the current software status is examined from the aspect of integration.
The SiL testing procedure encompasses **test generation** in a SiLEST-specific XML test case format, **test selection**, automatic **test execution** and automatic **test evaluation**, providing a discrete test verdict (e.g. 'passed', 'failed', 'not run') and also generating test reports.

Our presentation will look in detail at the entire SiLEST workflow as well as the various stages involved in the actual SiL test. Based on an example, it familiarizes the listener with the procedure involved in this early, systematic and automated closed-loop test of engine control software and points out the benefits of the proposed method, e.g. cost cutting, retraceability and reproducibility.

Biography

After leaving school in 1987 qualifying for university entrance, Dipl.-Ing. Sven Rebeschieß (born in 1968) studied electrical engineering at the Technical University of Berlin (TUB), majoring in software engineering/robotics. He then worked as a research assistant at the Institute for Microelectronics (1994-1995) as well the Control Engineering Department (1995-2000) at the TUB where, among other things, he developed the MIRCOS real-time simulation system that executes Simulink models on a C166-based microcontroller system [1].
Since the end of 2000, he has been employed with IAV GmbH Berlin in the field of algorithm development and algorithm testing for engine control units. He has been in charge of the “SiLEST” project at IAV since mid-2004 (cf. “www.silest.de”). This is co-funded by the German Federal Ministry of Education and Research (BMBF) as part of the “Software Engineering 2006” research initiative.

Reference

[1] Sven Rebeschieß. “MIRCOS – Microcontroller-Based Real Time Control System Toolbox for use with Matlab / Simulink”. Proceedings of the 1999 IEEE International Symposium on CACSD. Hawaii, USA, 1999, p. 243–248

Contact information of Presenter

Dipl.-Ing. Sven Rebeschieß
IAV GmbH / Abt. MS-E33, Carnotstr.1, D-10587 Berlin
Email: sven.rebeschiess@iav.de
Tel.: +49 30 39978-9550
Fax: +49 30 39978-9299

Dipl.-Ing. Thomas Liebezeit
TU Berlin / MDT, Einsteinufer 17, D-10587 Berlin
Email: thomas.liebezeit@tu-berlin.de
Tel.: +49 30 314-23950
Fax: +49 30 314-22120

Dipl.-Ing. Uzmee Bazarsuren
TU Berlin / MDT, Einsteinufer 17, D-10587 Berlin
Email: uzmee.bazarsuren@tu-berlin.de
Tel.: +49 30 314-25612
Fax: +49 30 314-22120