

Submission for:

- Software Quality Management (SQM)
primarily in German
- Software Testing (ICSTEST) in English
- Software Validation for Healthcare (CSVHC)
in English

Title of Paper

XML Format for Automated Software-in-the-Loop Tests

Presenter

Presenter: Dipl.-Ing. Thomas Liebezeit, TU Berlin / MDT, GER
Co-Presenter: Dipl.-Inf. (FH) Holger Schumann, DLR / SISTEC, GER
Further Authors: Dipl.-Ing. Sven Rebeschieß, IAV GmbH Berlin, GER
Further Authors: Prof. Dr.-Ing. Clemens Gühmann, TU Berlin / MDT, GER
Further Authors: Dipl.-Ing. Uzmee Bazarsuren, TU Berlin / MDT, GER

Session Format

- Workshop Tutorial Conference Presentation
-

Instructional Level

- Introductory Intermediate Advanced
-

Target Group

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

Keywords

- Software-in-the-Loop (SiL)
- XML (Extensible Markup Language) Test Description
- Simulation-Based Test
- Automated Test

Period

SiLEST Project March 2004 – Dec 2006

Abstract

1. Introduction

This paper presents a new XML test format developed during the SiLEST project. The SiLEST project¹ examines the suitability of the Software-in-the-Loop (SiL) technology for testing embedded software.

The aim of the project is to develop a SiL test environment and subsequently compare the obtained results with those from Hardware-in-the-Loop (HiL) tests. Automated regression tests are the main focus of interest for the automotive industry and also for the aerospace industry. The aim is to achieve cost-effective, earlier "in the loop" testing with excellent reproducibility, traceability and good documentation.

¹ "Software in the Loop for Embedded Software Test", www.silest.de; supported with funds from the German Ministry of Education and Research (BMBF) as part of the research initiative "Software Engineering 2006".

2. SiL testing

The SiL test is characterized by the fact that the whole test system is available in a simulation environment. A simulation model for the development of engine control unit (ECU) algorithms, for example, breaks down into

- an **ECU model** with the software algorithm under test as model and/or C-code, a model of the ECU's operating system (scheduler) and control unit I/O components, and
- an **environment model** with engine and transmission models, vehicle (longitudinal) dynamic features, actuator and sensor models (nominal and off-nominal behavior) and driver and ambient conditions.

3. SiLEST SiL test environment

The SiLEST test environment consists of the test management system, test execution system (test automation and test system) and version management system.

The **test management system** provides access to the individual tests and their results. It administers all corresponding data in a database. This means it is possible to reconstruct at any time who tested what, when and with which results. The test management system also instructs new tests. The **test execution system** consists of the test sequence controller, which allows for test automation, and the test system with its test components. The **version management system** is responsible for versioning all files used in the test in order to guarantee comprehensive reproducibility.

3.1 Files

The SiLEST test environment works with **test case** files which are independent of a specific test method (e.g. SiL) but defined for a certain system being developed. This means that the test cases can be reused for other test methods, so that the test results are comparable (e.g. SiL vs. HiL). In order to carry out tests with a version of the system (e.g. SiL variant), the corresponding structure is described in the **test configuration**, and the test case is mapped to this description. The obtained test results are recorded in the **test report** for further processing.

All three data formats are each defined with an XML scheme (Extensible Markup Language, www.w3.org). The advantages of an XML data format consist above all in the clear, open structure of the data, so that it is easy to reuse the data in conjunction with a transformation. In addition, a scheme guarantees simple validation of the data structure.

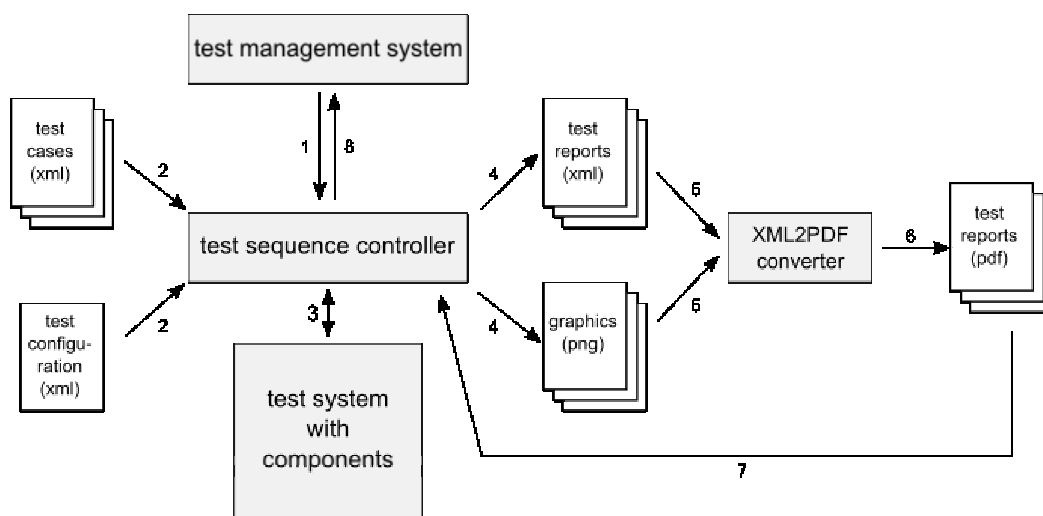


Fig. 1: Files and test procedure

Submission for:

- Software Quality Management (SQM)
primarily in German
- Software Testing (ICSTEST) in English
- Software Validation for Healthcare (CSVHC)
in English

3.2 Procedure

The files basically used in the test procedure are illustrated in Fig. 1. The processing of test files is triggered by the test management system (1). The test sequence controller takes on automatic processing of the test cases on the test system, also using the test configuration (2, 3).

The result consists of XML test reports, graphics (4) and the test result (binary test verdict). An XML2PDF converter transforms the test reports and graphics into a clearer format (.pdf) (5, 6). The test sequence controller sends these test reports and test results back to the test management system where the results of all tests are administered in a clearly organized structure (7, 8).

4. Describing the XML format

The SiLEST test environment uses three kinds of files whose basic structure and contents are explained below.

4.1 XML test case format

The test case format consists of various parts. Firstly, formal information is noted, such as a description of the test case, the author, the version, a project name, etc. This is followed by a definition of the specific test parameters (deviations from the basic calibration) of the system and test environment.

The next part defines analog input profiles or command sequences. The analog signals can be defined in intervals as vectors, functions or by means of basic mathematical functions (e.g. line, ramp). For automated evaluation of the test case, the expected results are defined in the last part (limits, epsilon environment, dynamic time warping) which permits generation of a binary result. The individual test results are linked into a total test result using logical elements (AND, OR, NOT).

All parts offer the possibility of extensive commentaries. This is important because the commentaries are used to generate the PDF test report.

4.2 XML test configuration format

The test configuration breaks down into two main parts. The first contains an abstract description of the test system. It depicts the individual components of the test system in such a way that the test sequence controller is provided with all necessary information (module name, designation, inputs and outputs). The second part maps the component inputs with the test signals defined in the test case and the component outputs with the defined test signals/port.

The test configuration also includes formal information, such as the current version or the author. Another essential component consists of the commentaries which are used for the PDF test report for documentation purposes.

4.3 XML test report format

The XML test report contains the test result and the individual results from the tests as defined in the test case. It also notes formal information (e.g. date, computer). The corresponding test case and the test configuration are connected by links. In addition, the test report contains version information for the individual test components and all corresponding files in order to ensure reproducibility of the test.

5. Conclusion and outlook

The XML test format defined for the SiLEST project was presented in terms of structure and contents. It consists of an XML description of the test cases, the test configuration (system being tested and mapping the test case to the system) and the test reports, and acts here as data format in the automated SiLEST test process. Work with the test cases and test configuration is carried out by a



test case editor, so that the test engineer does not have to write directly in the XML format.

The presentation will show the possibility of using XML as a flexible format in an extensive test environment. It will describe how test cases can be described regardless of the test method, illustrated with a consistent example relevant to practice.

Biography

Dipl.-Ing. Thomas Liebezeit (born 1976) graduated in electrical engineering from the Technische Universität Ilmenau in 2001, having specialized in automation and system engineering. He then worked as a research assistant for three years at the system and control theory/system analysis department of the TU Ilmenau. As part of the DeepC joint project funded by the German Ministry of Education and Research (BMBF), he was involved in the Mission Level Design for the autonomous underwater vehicle DeepC [1, 2]. He wrote a dissertation on this topic which is currently being brought to a successful conclusion.

Since early 2005, he has been a research assistant at the department of measurement and diagnostic technology of the Technical University of Berlin and works on the SiLEST joint project co-funded by the German Ministry of Education and Research (BMBF) on the aspects of software-in-the-loop and automated testing.

Reference

[1] Liebezeit, Thomas ; Zerbe, Volker: A Validation Environment for Mission Level Design of Autonomous Underwater Vehicles. In: Proceedings of the 49. Internationales Wissenschaftliches Kolloquium, Ilmenau, Germany, 2004

[2] Liebezeit, Thomas: Mission Level Design of Autonomous Underwater Vehicles. In: Proceedings of the First International ICSC Congress on Autonomous Intelligent Systems, ICAIS 2002, Geelong, Australia NAISO, ICSC-NAISO Academic Press, 2002.

Contact information of Presenter

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.-Inf. (FH) Holger Schumann
DLR / SISTEC, Lilienthalplatz 7, D-38108 Braunschweig
Email: holger.schumann@dlr.de
Tel.: +49 531 295 2776
Fax: +49 531 295 2767

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

Prof. Dr.-Ing. Clemens Gühmann
TU Berlin / MDT, Einsteinufer 17, 10587 Berlin
Email: clemens.guehmann@tu-berlin.de
Tel.: +49 30 314-29393
Fax: +49 30 314-22120

Dipl.-Ing. Uzmeë Bazarsuren



Submission for:

- Software Quality Management (SQM)
primarily in German
- Software Testing (ICSTEST) in English
- Software Validation for Healthcare (CSVHC)
in English

TU Berlin / MDT, Einsteinufer 17, D-10587 Berlin

Email: uzmee.bazarsuren@tu-berlin.de

Tel.: +49 30 314-25612

Fax: +49 30 314-22120
