

MiLEST Toolbox

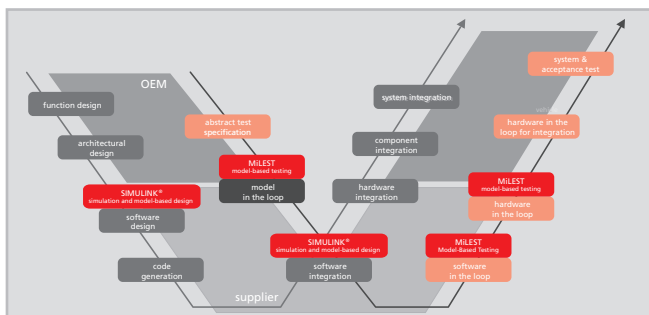
Model-in-the-Loop for Embedded System Testing



At the early stage of new system functionalities development a model serves as a primary means for including the novelties, yet there is no code, no hardware, and thus no real reference signals for testing. In MiLEST FOKUS proposes a new method for the stimulation and evaluation of embedded hybrid systems behavior which breaks down requirements into characteristics of specific signal features. A novel understanding of a signal is defined that enables us to describe it in an abstract way based on its properties – e.g., decrease, constant, maximum.

MiLEST Integration in the W-Model

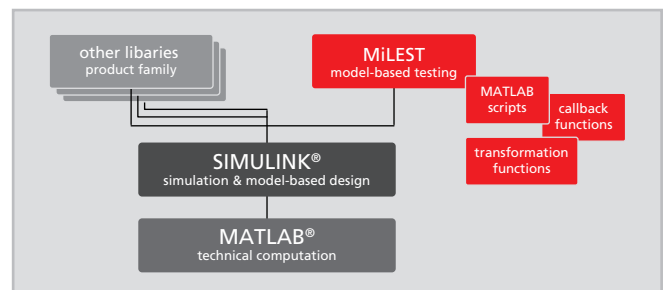
MiLEST applies to the software built into embedded systems. In particular, it refers to the software models from which systems are built. A model-based approach to functional black-box testing is used to provide a test model. This is in marked contrast to current test methods that are generally dedicated solutions specialized for specific testing contexts.



The test method FOKUS proposes is realized in MATLAB®/Simulink®/Stateflow® environment.

MiLEST Architecture

Technically, MiLEST is a Simulink add-on built on top of the MATLAB engine that represents an extension towards model-based testing activities. MiLEST consists of a library including callback functions, transformation functions, and other scripts.



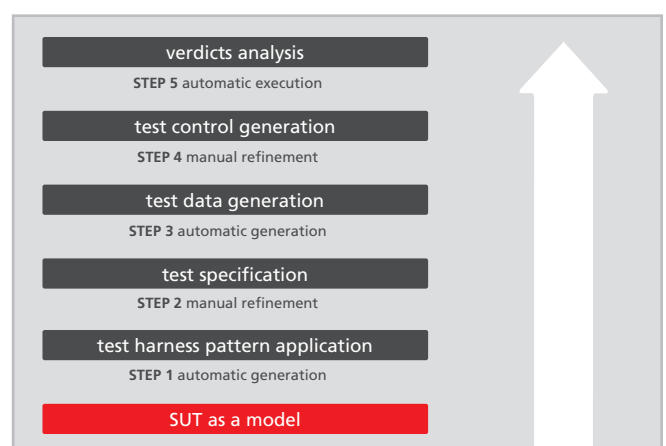
The testing library is divided into four composite parts.

Model-based Testing

MiLEST Model-based Testing (MBT) is testing in which the entire test specification is derived in whole or part from both the requirements and an SUT model that describes selected functional aspects of the SUT. In this context, the term entire test specification covers both the abstract test scenarios substantiated with concrete sets of test data and expected SUT outputs. It is organized in a set of test cases.

Test Development Process

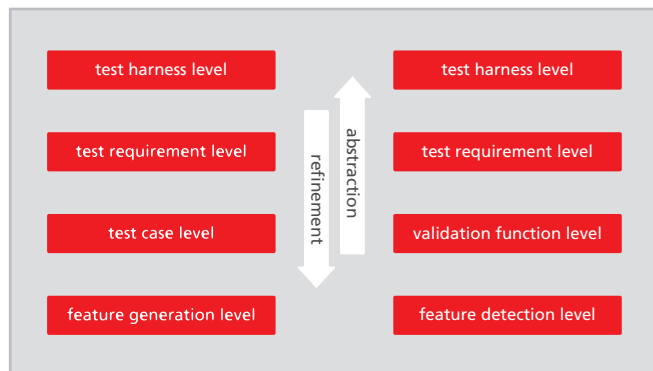
The Model-based Development paradigm assumes that the System Under Test (SUT) model is already available and that the input/output interfaces are clearly defined and accessible.



Besides the analysis of the SUT specification, proper functional, dynamic testing also requires systematic selection of test stimuli, appropriate test evaluation algorithms, and obviously an execution or simulation environment. If the above assumptions hold, a pattern for generation of a test harness model can be applied on the SUT model as denoted in step I. This is done automatically with a MiLEST transformation function, giving a concrete frame for test specification. Further on, the test engineer refines the test specification design (step II) using the concept of validation function patterns. Structures for test stimuli and concrete test signals are then generated. This step (step III) occurs automatically with application of the transformations. The test control design can be added automatically too (step IV). Finally, the tests may be executed and test results obtained in the form of verdicts (step V). At the same time the quality of the test system specification produced is also assessed.

Hierarchical Test System

The MiLEST specifications reflect the structure of the system requirements. The test development process and the abstractly aligned test system enable us to apply the concepts of signal feature generation and detection mechanisms while also building test specifications systematically. In this context, the test system is understood as a hierarchically structured test model – also called the ‘test design’.



Since the aim of the test system is not to provide the means for testing a single signal property but for validating complete SUT models independently of their complexity, structuring the test models in a proper way contributes to the scalability and reusability of the solution. Moreover, traceability of the test development artifacts is possible and transformation potentials can be identified.

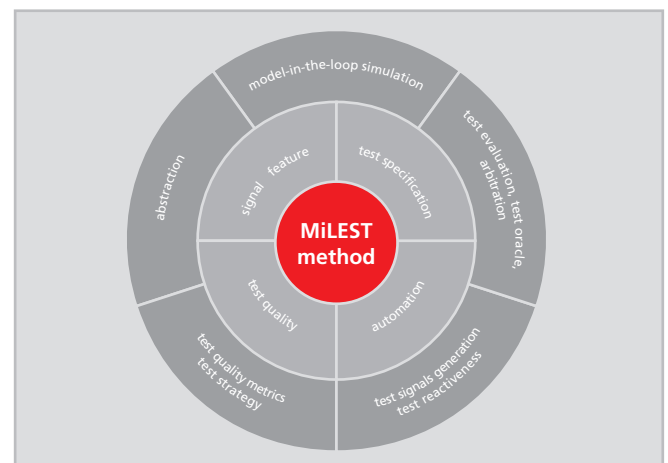
Contact

Dr. Justyna Zander-Nowicka
MOTION Competence Center
justyna.zander-nowicka@fokus.fraunhofer.de

The structure of the test system consists of four different levels that can be built systematically. This makes it less error-prone while also leaving the test engineers plenty of scope for developing the complete test specification.

Features

- Systematic, consistent functional test specification
- Signal feature-oriented paradigm
- Graphical test design
- Test process automation
 - systematic and automatic test data generation
 - online automatic test evaluation
- Model-in-the-Loop test execution
- Reusable test patterns
- Abstract and concrete views



Benefits

- Testing in early design stages
- Test of hybrid systems including temporal and logical dependencies
- Traceability of test cases to the requirements
- Traceability of verdicts to the root faults
- Increased test coverage and test completeness
- Assured test quality of the test specification

Prof. Dr. Ina Schieferdecker
Head of MOTION Competence Center
ina.schieferdecker@fokus.fraunhofer.de