



Fokus!MBT - The Test Modeling Environment

Efficient quality assurance through test models

Fokus!MBT overcomes

traditional testing challenges:

- Implicit tester's knowledge
- Unstructured test case derivation
- Lack of automatization
- Inflexible process structures
- Inadequate documentation
- Insufficient communication among stakeholders

Benefits of UML Testing Profiles

- Standardized concepts for model-based test specifications
- (Re-)Use of UML and its diagrams
- Interoperability among UML compliant tools
- Vendor independent maintenance and enhancement of the standard

Quality matters

Software-intensive systems are omnipresent: Whether in finance and healthcare, in aerospace or traffic, software plays an ever greater role. In order to remain competitive in the market, manufacturers are more and more under pressure to deliver high-quality software systems quickly and reasonably priced. Therefore reliable and reusable processes and methods are essential – both in development and testing of software-intensive systems. Yet traditional testing still faces the same challenges it has for many years: process structures are relatively inflexible, documentation is incomplete or nonexistent and most testing tasks lack automation, classification and reproducibility. The most promising technology to address these challenges is model-based testing. This approach is based on the idea of using formal models – so-called test models – to improve certain key activities, such as the design of test data and test cases. The overall goal of model-based testing is to improve the test quality, the entire system's development cycle and ultimately the quality of the system itself.

Test modeling based on the UML Testing Profile

Fokus!MBT is an integrated test modeling environment that guides the user along the methodology of Fokus!MBT and thereby simplifies the creation of the underlying test model. A test model includes test relevant structural, behavioral and methodical information. By formalization, the tester's knowledge can be machinably preserved as well as evaluated and exploited at any time – for instance to generate further test-specific artifacts, such as test cases and test scripts. Another benefit of the test model is the possibility to visualize and document the test specifications. The modeling notation used by Fokus!MBT is the UML Testing Profile (UTP) specified by the Object Management Group. It is a test-specific extension of the Unified Modeling Language (UML), which is prevalently used in industry. This enables testers to rely on the same language concepts as system architects and requirement engineers, which overcomes problems in communication and support the mutual comprehension.

Test Service infrastructure

The core of Fokus!MBT is based on a service-oriented concept for test-specific services. These include:

- Interface description for test cases and/or test data generators, test script generators, test result feedback from different test execution platforms, report engines etc.
- Adjustment respectively creation of adequate user interfaces and context-specific actions
- Individual customization of Fokus!MBT

Features

- Context-specific modeling commands
- Automated modeling rules
- Test-related diagram views
- Service-specific and semantic model validation
- Feedback of test execution results
- Test case generator: SpecExplorer
- Test script generator: TTCN-3
- Reporting engine: BIRT, GenDoc2

Customized creation of test models

Fokus!MBT is based on the flexible Eclipse RCP platform, the Eclipse Modeling Framework (EMF) and Eclipse Papyrus. As a UTP-based modeling environment it is equipped with all UML diagrams but it also includes additional test-specific diagrams. Besides these diagrams, Fokus!MBT goes for a proprietary editor framework to visualize and edit the test model. The graphical editor interface can be adapted and optimized for the user's needs and skills. Thereby, if necessary, Fokus!MBT completely abstracts from the underlying UML/UTP, which allows even non-IT experts to build model-based test specifications in reasonable time. This is supported by offering context-specific actions that guide the user along the Fokus!MBT methodology. Methodically incorrect or contextually unrewarding actions are not even provided to the user. On top of these actions, Fokus!MBT integrates automated modeling rules. Modeling rules ensure the compliance of guidelines – particularly modeling and naming conventions – after and during working with the test model. These constructive quality assurance mechanisms distinguish Fokus!MBT from other UML tools, accelerate the creation of the model and minimize cost-intensive review sessions.

Extended traceability of the requirements in the test model

The validation of the system under test regarding its requirements is the main target of all testing activities. Thereby the consequent and continuous traceability among requirements and test artifacts – especially among requirements and test cases – is indispensable, but not sufficient. Fokus!MBT takes a major step forward by integrating the test execution results into the test model's inherent traceability network. This establishes a consistent traceability network between requirement, test case, test script and test execution results, making conclusions about the coverage of the particular requirement or the test progress itself immediately calculable. Furthermore the visualization of the test execution results allows a detailed analysis of the test execution flow to preprocess and ultimately evaluate the test results. Thus the test model includes all relevant information to assess the quality of the system under test in order to support the management in its decision making with respect to the release of the system.

Kontakt

Marc-Florian Wendland
marc-florian.wendland@fokus.fraunhofer.de
Tel. + 49 30 3463-7395
www.fokusmbt.com

Fraunhofer Institute for Open
Communication Systems
Kaiserin-Augusta-Allee 31
10589 Berlin

www.fokus.fraunhofer.de

The Fokus!MBT service architecture

