

# Modellbasiertes Testen (MBT) in der Softwareentwicklung

## Model-based Testing (MBT) in Software Engineering

Bernhard Aichernig, Bernhard Peischl, Franz Wotawa



*Bernhard Aichernig ist Assistenzprofessor am Institut für Softwaretechnologie (IST) der TU Graz. Seine Forschung konzentriert sich auf die Kombination von formalen Entwicklungsmethoden und fortgeschrittenen Testtechniken. Er war (bzw. ist) leitender Forscher und Projektmanager in drei EU-Projekten zum Thema MBT: CREDO, MOGENTES und MBAT.*

► <http://aichernig.blogspot.com>

*Bernhard Aichernig is assistant professor at the Institute for Software Technology (IST) at Graz University of Technology. His research focuses on the combination of formal development methods and advanced testing techniques. He has been the key researcher and project manager in three EU projects on MBT: CREDO, MOGENTES and MBAT.*

► <http://aichernig.blogspot.com>

**Testen zur Sicherung der Softwarequalität hat einen großen Stellenwert. Dies gilt speziell für Systemtests vor Auslieferung der Produkte. Um die Testkosten bei gleicher Qualität zu senken bzw. die Qualität bei gleichen Kosten zu steigern, ist es notwendig, den Systemtest zu automatisieren. Dies kann auf zwei Arten geschehen. Einerseits durch die Automatisierung der Testausführung und andererseits durch die Automatisierung der Testfallgenerierung. Im Artikel werden wir im Speziellen auf die Automatisierung durch die Verwendung von Modellen fokussieren.**

Fehler in Programmen sind nicht nur ärgerlich, sondern verursachen der Volkswirtschaft hohe Kosten. Schätzungen gehen hier von direkten und indirekten Kosten in der Höhe von 100 bis 150 Milliarden Euro pro Jahr für Europa aus. Auf Österreich umgelegt sind das 3 bis 4 Milliarden Euro. Neben den volkswirtschaftlichen Kosten haben Programmfehler natürlich auch Auswirkungen auf die Reputation von Firmen. Eine frühzeitige Erkennung und Korrektur der Fehler hat somit einen sehr hohen praktischen Stellenwert. Frühzeitig bedeutet im Umfeld der Softwareentwicklung vor der Auslieferung von Programmen an die Kunden und Kundinnen.

Es gibt eine Reihe von Gründen, die für Fehler in Programmen verantwortlich sind. Zum einen werden Programme noch immer zu wenig getestet. Zum anderen entstehen viele Fehler erst durch das Zusammenspiel von unterschiedlichen Systemteilen beziehungsweise Teilprogrammen. Dieses Problem hat mit der Komplexität der heutigen Programme zu tun, die aufgrund immer höherer Erwartungen an die Funktionalität weiter im Steigen begriffen ist. Um Fehler auf Systemebene zu finden, sind entsprechende Systemtests erforder-

*Testing to ensure software quality is of very high importance. This holds especially for system tests performed before software deployment. In order to decrease testing costs while retaining high quality, or increasing quality while keeping testing costs constant, it is important to automate the system test. This can be done either at the test case execution or the test case generation phase. In this article we focus on the automation of test case generation using models of the system under test.*

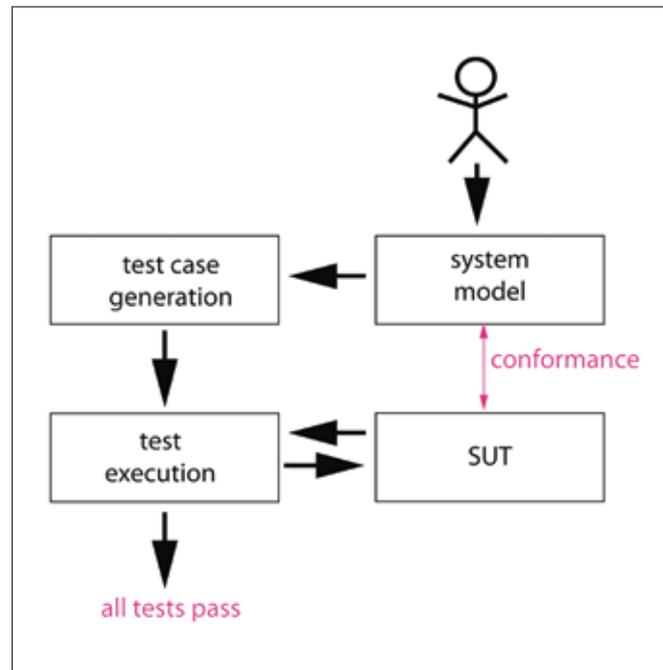
Faults in programs do not only aggravate but also cause costs. Estimates for the direct and indirect costs accrued per annum in the European Union are between 100 and 150 billion euros, which is about 3 to 4 billion euros for Austria alone. Beside the economic costs, there is also the possible loss of reputation of companies developing programs. Therefore, an early detection and correction of faults before program deployment is of great practical interest.

There are a number of reasons why today's programs still fail during use. One reason is the lack of testing during and after development. Another reason lies in the complexity of systems comprising interacting sub-systems where faults are revealed because of the interaction. Unfortunately, the complexity of programs and systems is still increasing because of the increase in demanded functionality. In order to detect faults in interacting systems, a usually rather expensive system test has to be performed.

To reduce the costs of testing while retaining the level of software quality on the one hand and to increase the quality of the software on the other hand, test automation is a must. We distinguish two types of automation – automation of test case execution and automation of test case genera-

Abb. 1: In MBT wird das vom Tester erzeugte Modell zur automatisierten Testfallerzeugung verwendet. Werden alle Testfälle korrekt ausgeführt, dann entspricht das System unter Test (SUT) dem modellierten Verhalten.

Fig. 1: In MBT the tester creates a model, then test cases are automatically generated and executed. If all tests pass, the system under test (SUT) conforms to the model.



© TU Graz/IST

lich, die jedoch üblicherweise hohe Kosten verursachen.

Um nun einerseits die Kosten für das Testen zu reduzieren und andererseits die Qualität der Programme zu erhöhen, ist Testautomatisierung erforderlich. Hier unterscheidet man die Automatisierung der Testausführung und die Automatisierung der Testfallerzeugung. Am Institut für Softwaretechnologie beschäftigen wir uns seit mehr als zehn Jahren mit der automatisierten Erzeugung von Testfällen, wobei wir hier die Testfälle aus Systemmodellen gewinnen. Ein Modell beschreibt die gewünschte Funktionalität des Systems. Hier werden unter anderem Automaten (Finite State Machines) oder auch Transitionsysteme (z. B. Labeled Transition Systems) zur Modellierung verwendet, die es ermöglichen, Systemzustandsänderungen abhängig vom aktuellen Zustand und von den Systemeingangswerten zu beschreiben.

Ausgehend von den Modellen können Testfälle erzeugt werden, die gewisse Eigenschaften aufweisen. Im Prinzip funktioniert die Testfallgenerierung durch Analyse des Modells, wobei Sequenzen von möglichen Eingangswerten und erwarteten Ausgangswerten extrahiert werden, die einen Testfall darstellen. Da diese Testfallgenerierungsverfahren auf Modellen basieren, werden sie unter dem Begriff Modellbasiertes Testen (MBT) zusammengefasst (siehe auch Abbildung 1).

In den letzten Jahren wurde am Institut eine Reihe von MBT-Projekten auf EU- (z. B. das FP7-Projekt MOGENTES) und nationaler Ebene (Pro-

tion. The Institute for Software Technology has been working in the area of automating test case generation for more than 10 years. The work is based on system models describing the required functionality of a system. In our case finite state automata and labeled transition systems are used for modeling. This allows for specifying the transition of system states based on the current state and inputs. A test case that is generated automatically from the model is a sequence of inputs and expected outputs. Because of the generation of test cases from models the underlying method is called Model-Based Testing (MBT). Figure 1 illustrates the principle of MBT.

During the past years we have been carrying out several MBT projects, e.g. MOGENTES funded by the European Union, and Softnet funded by the FFG, together with our industrial and scientific partners. The projects have a common focus that lies in transferring basic research results into daily industrial practice, thus leading to new research questions. In particular, providing a well suited modeling language and efficient test-case generation algorithms have been of interest. The efficient generation of tests from highly complex industrial models within a given time has been an especially demanding challenge. In addition, other challenges have been met because of the fruitful collaboration with industry. Generating test cases based on certain coverage criteria for test execution within a limited time, and the prioritization of test case execution are two examples that have been tackled in our projects.



Bernhard Peischl ist Koordinator des Kompetenznetzwerks Softnet Austria (Programme K-net Softnet Austria und COMET K-Projekt Softnet Austria II). Er widmet sich der anwendungs-nahen, produktbezogenen Forschung anhand von ausgewählten Referenzprojekten mit Partnern aus der Telekommunikation, dem Automotive-Bereich, der Banken- und Versicherungsbranche sowie Toolherstellern und Beratern.

Bernhard Peischl is the coordinator of the competence network Softnet Austria (Programme K-net Softnet Austria and COMET K-Project Softnet Austria II). He conducts applied and product-specific research by means of selective reference projects with partners from telecommunications, the automotive field and the banking and insurance domain as well as tool developers and consulting companies.

Abb. 2: Ein Modell eines Alarmsystems für Automobile.

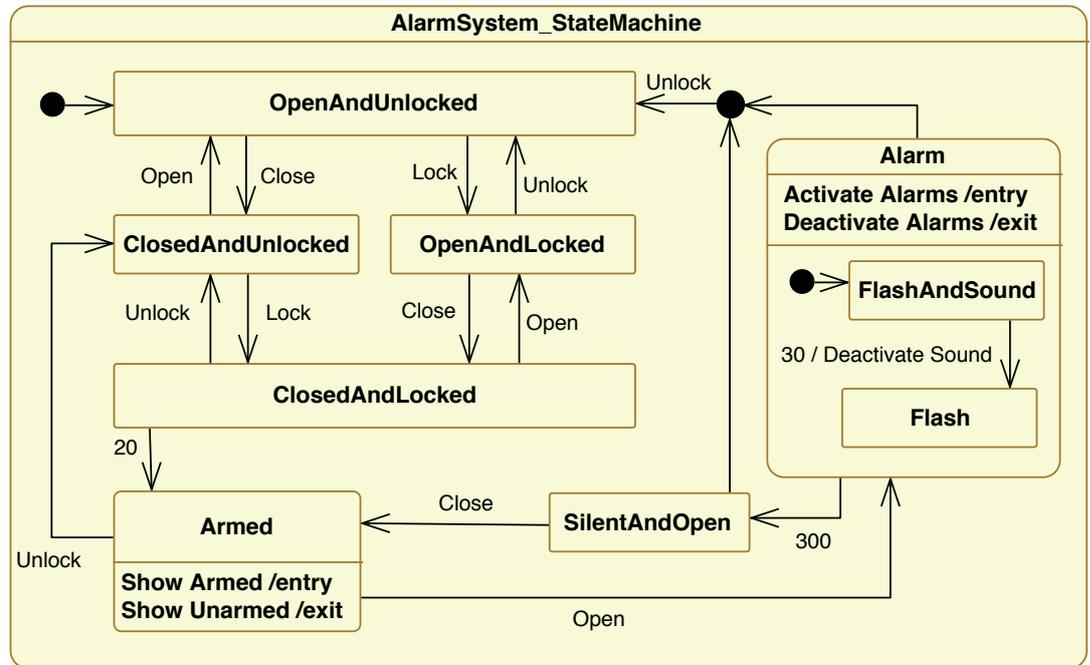
Fig. 2: A model of a car alarm system.

© EU FP7 Projekt MOGENTES



Franz Wotawa ist Professor für Softwareentwicklung am Institut für Softwaretechnologie der TU Graz und Leiter des COMET K-Projekts Softnet Austria. Neben dem Modellbasierten Testen forscht er in den Bereichen Automatisiertes Programmdebuggen, Künstliche Intelligenz und Robotik.

Franz Wotawa is full professor at the Institute for Software Technology, Graz University of Technology, and head of the COMET K-Project Softnet Austria. Beside model-based testing, his research focus is on automated program debugging, artificial intelligence, and robotics.



jekte des K-net Softnet Austria und auch Projekte des Nachfolgeprogramms COMET K-Projekt Softnet Austria II) gemeinsam mit Partnern aus der Industrie und Forschung durchgeführt. Gemeinsam ist diesen die Bestrebung, Grundlagenforschungsergebnisse in die industrielle Praxis zu transformieren, was erfahrungsgemäß zu weiteren Forschungsfragen führt. Offene Fragestellungen hierbei waren die Bereitstellung von geeigneten Modellierungssprachen und Testfallgenerierungsalgorithmen. Gerade die Erzeugung geeigneter Testfälle aus komplexen industriellen Modellen war und ist hier eine besondere Herausforderung. Zusätzliche Probleme sind durch industrielle Anforderungen an die erzeugten Testfälle bedingt. So sollen die Testfälle möglichst alle Teile eines Programms ansprechen und auch in einer vorgegebenen Zeit ausgeführt werden können. Eine Priorisierung von Testfällen hinsichtlich gegebener Kriterien wird ebenso angestrebt.

Im Rahmen der MBT-Projekte konnten Lösungen für die meisten dieser Probleme gefunden werden. Unter anderem wurden in einem Projekt Fehler in einer kommerziellen Voice-over-IP (VoIP)-Software gefunden, die durch die vorhergehenden Tests nicht entdeckt wurden. Möglich gemacht wurde dieser Erfolg durch die Modellierung von VoIP-Protokollen und der automatisierten Testfallerzeugung aus diesen Modellen. Nähere Informationen über die Projekte findet man auf folgenden Webseiten:

- ▶ [www.soft-net.at](http://www.soft-net.at)
- ▶ [www.mogentes.eu](http://www.mogentes.eu)

For most of the open challenges, specific solutions were provided in the projects, and very promising results obtained. For example, with MBT, new faults in a commercial voice-over-IP (VoIP) software have been detected. This was possible due to a model of the VoIP protocol and algorithms for efficiently generating the test cases. More information regarding the projects can be obtained from the project websites:

- ▶ [www.soft-net.at](http://www.soft-net.at)
- ▶ [www.mogentes.eu](http://www.mogentes.eu)