

# **Automatisierte Strukturtests in der modellbasierten Entwicklung**

Dipl.-Ing. Meike Lim, Dr. Sadegh Sadeghipour, Dr. Hans-Werner Wiesbrock<sup>1</sup>

## **Abstract**

To ensure the quality of automatic code generation by back-to-back tests one often needs stronger test coverage of the model than usually reached by black-box testing. In that case after exhaustive testing based on the software requirements and monitored model coverage one can use test data generators combined with an automatic test evaluation tool for performing the missing tests. In this article we present a tool coupling with a well established test environment which allows automatized structure tests for back-to-back, as well as regression tests.

## **Zusammenfassung**

Um eine automatische Codegenerierung abzusichern wird meist eine höhere Modellüberdeckung durch Back-to-Back-Tests gefordert, als durch übliche funktionale Tests. So stellt man fest, dass nach einer gründlichen und erfolgreichen Ausführung funktionaler Tests, bei denen die Modellüberdeckung aufgezeichnet und gemessen wurde, die erreichten Überdeckungsmaße noch nicht hinreichen. Hier können zwei Werkzeuge helfen, die noch fehlenden Tests zu spezifizieren und auszuwerten.

In diesem Beitrag wird vorgestellt, wie durch die Anbindung zweier Tools in eine etablierte und in der Automobilindustrie wohlerprobte Werkzeugkette automatisierte Strukturtests in Regressions- und Back-To-Back-Überprüfungen integriert und begleitend durchgeführt werden können.

---

<sup>1</sup> Unterstützt durch das BMBF- Projekt IMMOS, Proj.-Ref. 01ISC31C

## 1. Einleitung

Die zunehmende Verbreitung der softwaregesteuerten Systeme im Automobil und deren steigende Komplexität fordern neue Methoden bei Entwicklung und Test der Software. Seit Mitte der 90er Jahre ist hier ein Paradigmenwechsel im Vorgehen der Entwicklung erkennbar: Die konventionellen Entwicklungsweise wird zunehmend durch einen modellbasierten Prozess ersetzt. Dieses wird schrittweise um Design- und Implementierungsaspekte angereichert und dient schließlich als ‚Vorlage‘ für eine manuelle oder automatische Codierung der eingebetteten Software. Die Vorteile der modellbasierten Entwicklung mit Entwicklungsumgebungen wie MATLAB/Simulink/Stateflow sind lange bekannt [5]. Die modellbasierte Entwicklung eröffnet auch neue Möglichkeiten und Synergiepotenziale für den Testprozess. Insbesondere entsteht hier zu einer Funktionalität routinemäßig eine Vielzahl ausführbarer Artefakte (z.B. verschiedene Modellversionen, Software und das integrierte System), die getestet werden können. Der Test erfolgt auf mehreren Ebenen (Modelltest, Softwaretest und Test des integrierten Systems). Nutzbringend können und sollen dann die für den Modelltest aufgestellten Testszenarien auch für den Test der späteren Modellversionen, des Programmcodes oder des Systems mit integrierter Software eingesetzt werden [2,4]. In diesem Kontext und zur Sicherung der Qualität der entwickelten Software müssen Regressions- und Back-to-Back-Tests in verschiedenen Entwicklungsphasen durchgeführt werden.

Regressionstests stellen sicher, dass die Modifikationen im Modell die schon korrekt umgesetzte Funktionalität der älteren Versionen ebenfalls erfüllen. Back-to-Back-Tests dienen dem Nachweis der Äquivalenz zwischen den verschiedenen Repräsentationsformen eines Testobjekts, z.B. zwischen dem Simulink/ Stateflow-Modell und dem daraus erzeugten Programmcode.

Im systematischen Testprozess leitet man zunächst aus den Anforderungen funktionale Black-Box-Tests ab. Bei der Durchführung dieser Tests zeichnet man die verschiedenen erreichten Modellüberdeckungen auf [1]. Diese Kennzahlen geben Auskunft über die Testtiefe. Da nun Testen stets Stichprobencharakter hat, werden einige Codebereiche selten oder fast gar nicht durchlaufen. Für eine automatische Codegenerierung ist es andererseits wichtig, dass auch diese Teile korrekt übersetzt werden. Hier können Werkzeuge zur Testdatengenerierung den noch ausbleibenden Nachweis erleichtern [9]. Eine automatische Testauswertung der resultierenden Back-To-Back-Tests erlaubt dann eine vollständig automatisierte Durchführung dieser offenen Aufgabe [3].

In diesem Beitrag wird vorgestellt, wie durch die Anbindung zweier Tools in eine etablierte und in der Automobilindustrie wohlerprobte Werkzeugkette automatisierte Strukturtests in Regressions- und Back-To-Back-Überprüfungen integriert und begleitend durchgeführt werden können, siehe Abbildung 1 .

Der vorzustellende Prozess setzt auf folgende Werkzeugkette auf: *Matlab/Simulink/Stateflow* (The MathWorks, Inc.)

als Modellierungsumgebung, *TargetLink* (dSPACE GmbH) für die automatische Codegenerierung und *MTest* (ebenfalls dSPACE GmbH) als Testumgebung sowohl für das Modell als auch für den generierten Code auf dem Host- oder Target-Rechner.

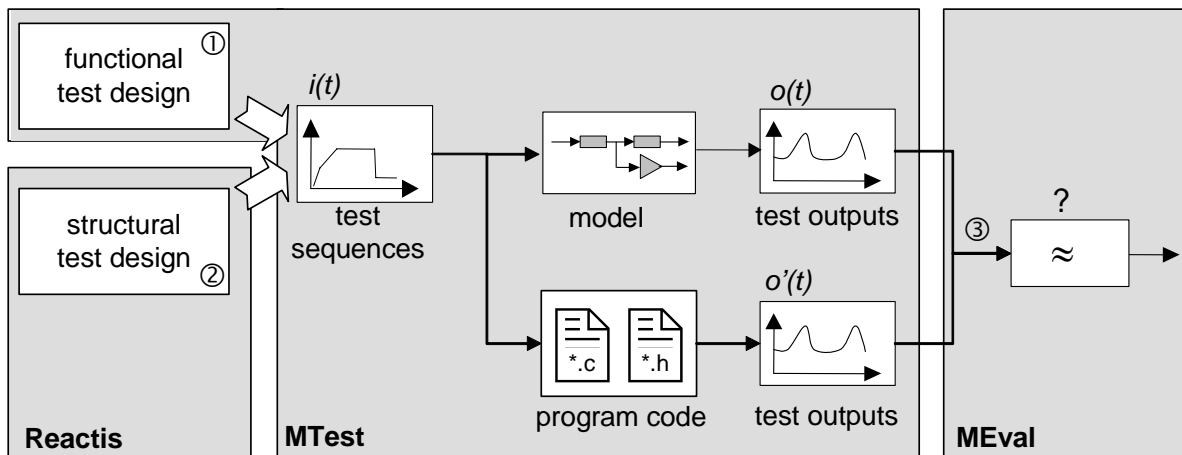


Abbildung 1 : Kopplung von *MEval* und *Reactis* an *MTest*

Um automatisiert Strukturtests durchführen zu können, werden die Werkzeuge *Reactis* (Reactive Systems, Inc.) und *MEval* (IT Power Consultants) eingebunden. Mit *Reactis* werden, ausgehend von den erstellten Modellen und geeigneten Überdeckungskriterien, automatisch Struktur-überdeckende Testdaten generiert und nach *MTest* exportiert. Dort werden sie anschließend auf dem Modell, auf dem Code oder auch auf dem Target-Prozessor ausgeführt. Die automatisierte Auswertung der Testergebnisse erfolgt mit *MEval*. Durch die Automatisierung solcher Strukturtests, die bei der vorgestellten Werkzeugkette im Batch-Betrieb durchgeführt werden können, wird ein Vielfaches an Zeit und Aufwand bei Regressions- oder Back-to-Back-Tests eingespart.

## 2. Testdatengenerierung mit *Reactis*

Ein Modell wurde ausgiebig funktional getestet und die erreichten Modellüberdeckungen aufgezeichnet. Aber für die Codegenerierung reicht die erreichte Modellüberdeckung nicht aus.

In diesem Fall lassen sich mit dem Werkzeug *Reactis* Testdaten generieren, die eine verbesserte Modellüberdeckung bezüglich der vom Benutzer gewählten Überdeckungskriterien erreichen, siehe Abbildung 2.

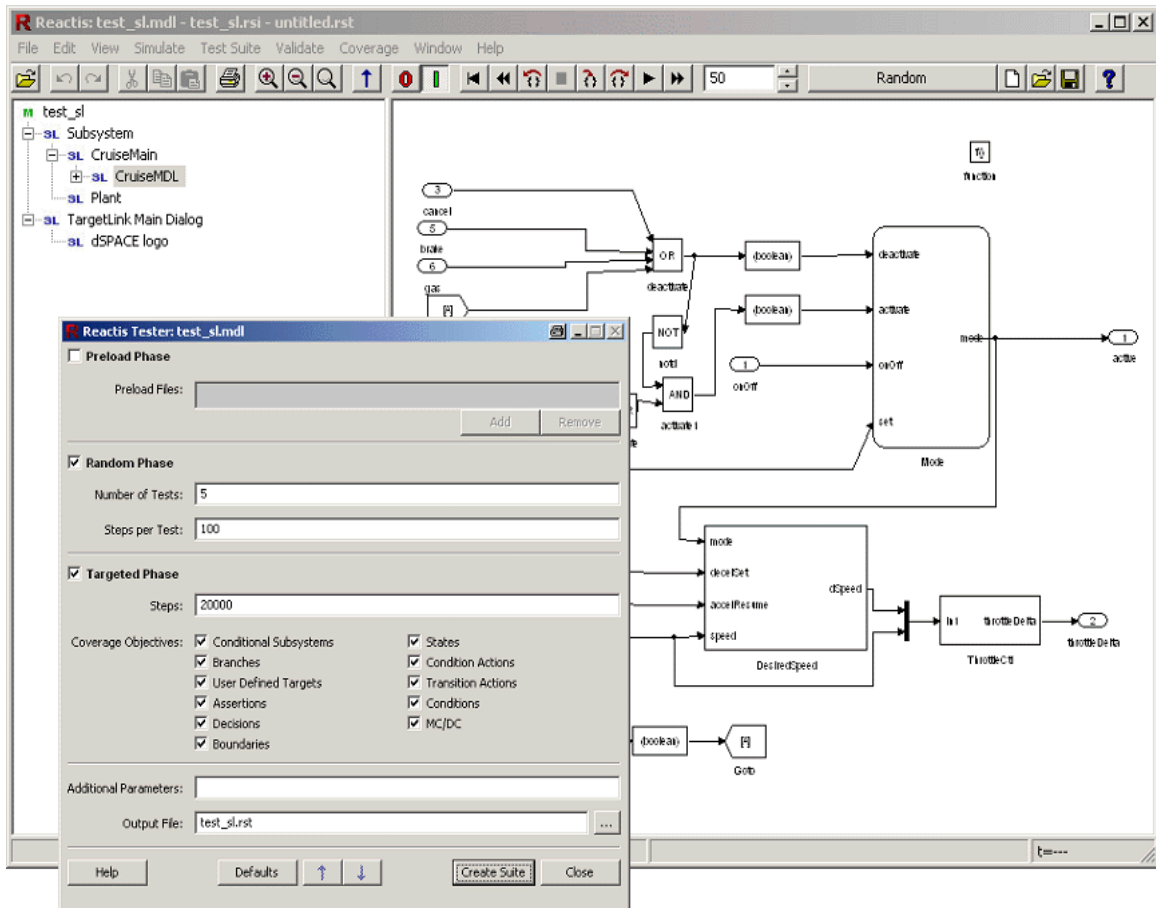


Abbildung 2 : Testdatengenerierung mit *Reactis*

Über sogenannte benutzerdefinierte Targets oder Assertions lassen sich ferner gezielt kritische Testdaten generieren, die auf diese Targets oder zur Widerlegung der Assertions hin gesucht werden, siehe Abbildung 3.

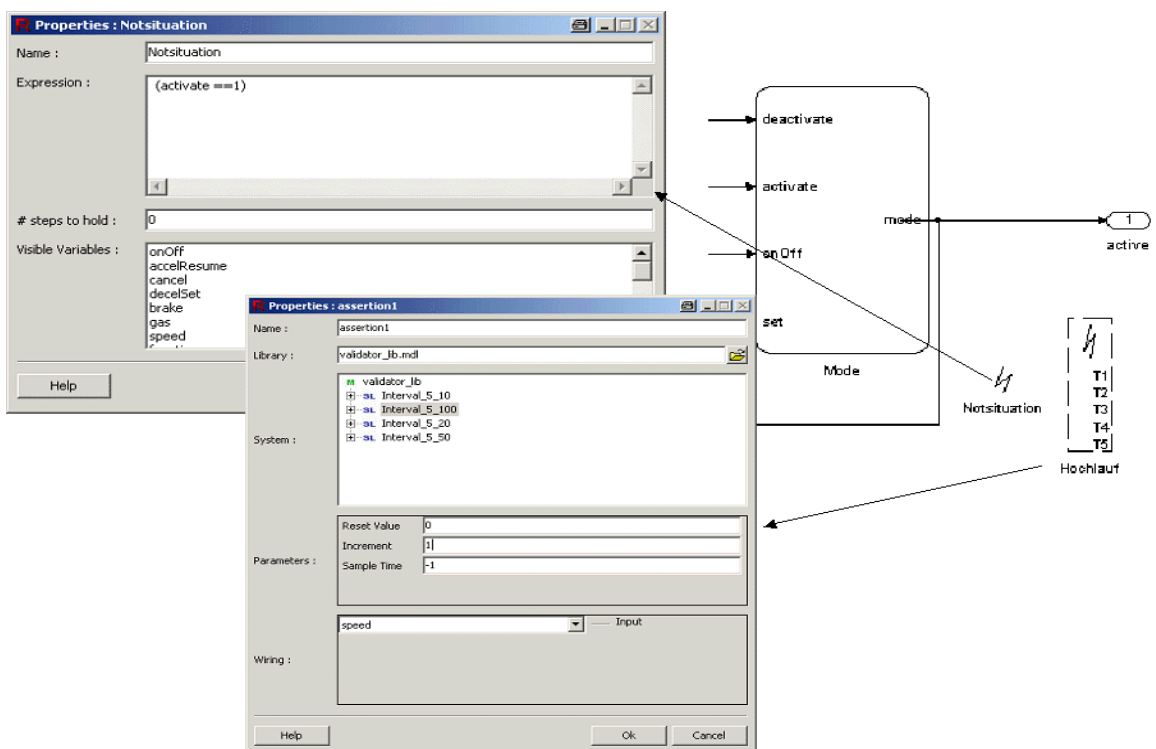


Abbildung 3 : Gezielte Suche nach Testdaten in *Reactis*

Zugleich bietet das Tool *Reactis* auch eine eigene Simulation an. Die hiermit erhaltenen Modelausgaben zu den generierten Testdaten können zusammen exportiert werden.

Die Generierung der Testdaten und der Export der Simulationsergebnisse können in einem Batchlauf durchgeführt werden. Der Benutzer ruft nur noch eine geeignete MATLAB-Funktion auf, und die Testdaten, die erreichten Überdeckungen und die Simulationsergebnisse werden ihm automatisch in einem geeigneten Format zur Verfügung gestellt.

### 3. Testdurchführung mit *MTest*

Eine etablierte Testumgebung für Modelle bietet die Firma dSpace mit *MTest* an. Dieses Werkzeug erlaubt die komfortable Verwaltung und Ausführung von Tests auf der Modellebene, (MiL, Model in the Loop), auf dem generierten Code, (SiL, Software in the Loop) und auch auf einem Evaluationsboard, (PiL, Processor in the Loop), siehe Abbildung 4 .

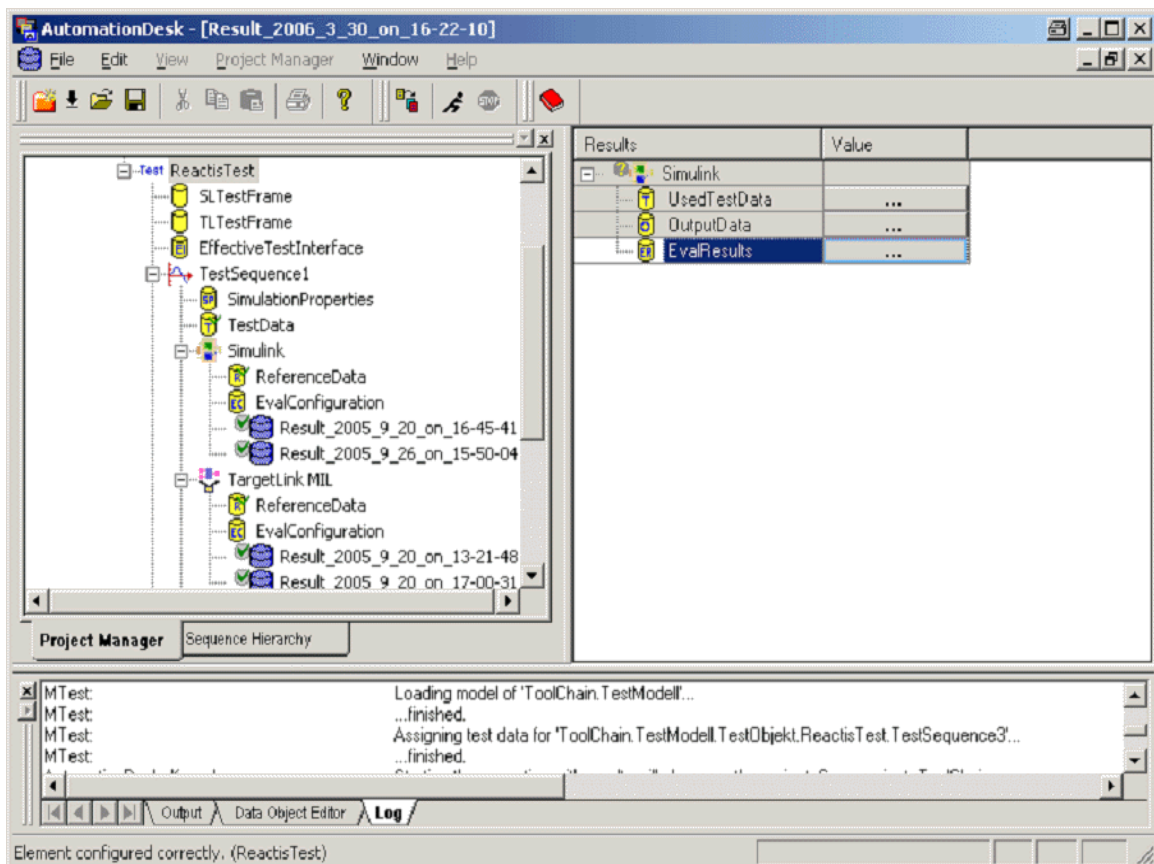
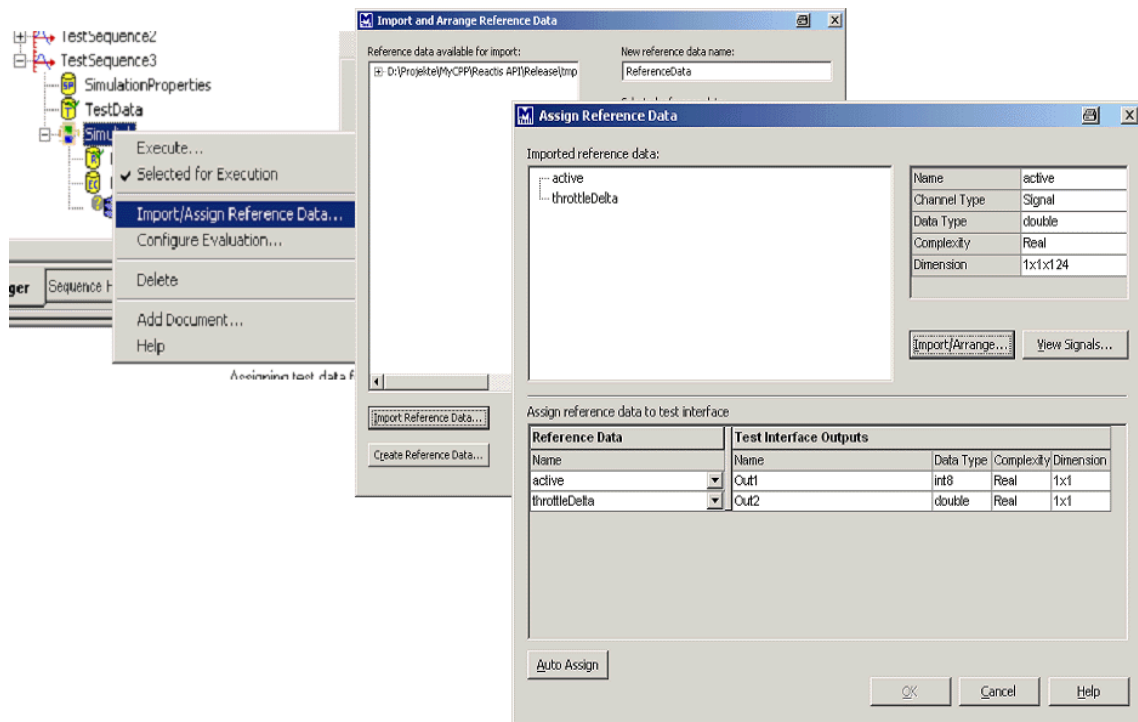


Abbildung 4 : *MTest*

Zugleich lassen sich die verschiedenen Model- und Codeüberdeckungs-Eigenschaften, z.B. MC / DC (modified condition / decision coverage) aufzeichnen. Diese Testumgebung bietet nun an, die Modelle mit den aus *Reactis* generierten Testdaten anzutreiben.

Dazu lassen sich die zuvor aus *Reactis* exportierten Testdaten in *MTest* importieren. Die Simulationsergebnisse werden dann als Referenzdaten in *MTest* gesetzt, siehe Abbildung 5 .



**Abbildung 5 : Import der Simulationsergebnisse in *MTest***

Anschließend können die Tests dann auf allen Ebenen durchfahren werden.

#### 4. Automatische Testauswertung mit *MEval*

Ziel des in den letzten Abschnitten beschriebenen Verfahrens war die Absicherung der Codegenerierung durch Testfälle mit erhöhter Modellüberdeckung. Insbesondere bedeutet dies, dass die Simulationsergebnisse der Testläufe für alle Testebenen (MiL, SiL und PiL) ausreichend ähnlich sind. Alle automatisch generierten Tests können jetzt nach dem Import in einem Batchlauf aus *MTest* heraus gefahren werden. Es bleibt noch der Vergleich auf 'Gleichheit'.

Durch Fixpunkt-Arithmetik und Rundungsfehler können Abweichungen in den Signalwerten auftreten, aber auch durch die verschiedenen Integrationen im zeitlichen Ablauf. Dazu wurden spezielle Algorithmen von IT Power Consultants gemeinsam mit der Forschung der DaimlerChrysler AG entwickelt und im Werkzeug *MEval* implementiert (siehe [6]), das einen solchen Signalvergleich durchführt (siehe Abbildung 6 ). Diese Algorithmen erlauben neben anderem eine dezidierte Beurteilung der Gleichheit von Signalen unter getrennter Berücksichtigung erlaubter Zeitverschiebungen und Werteabweichungen.

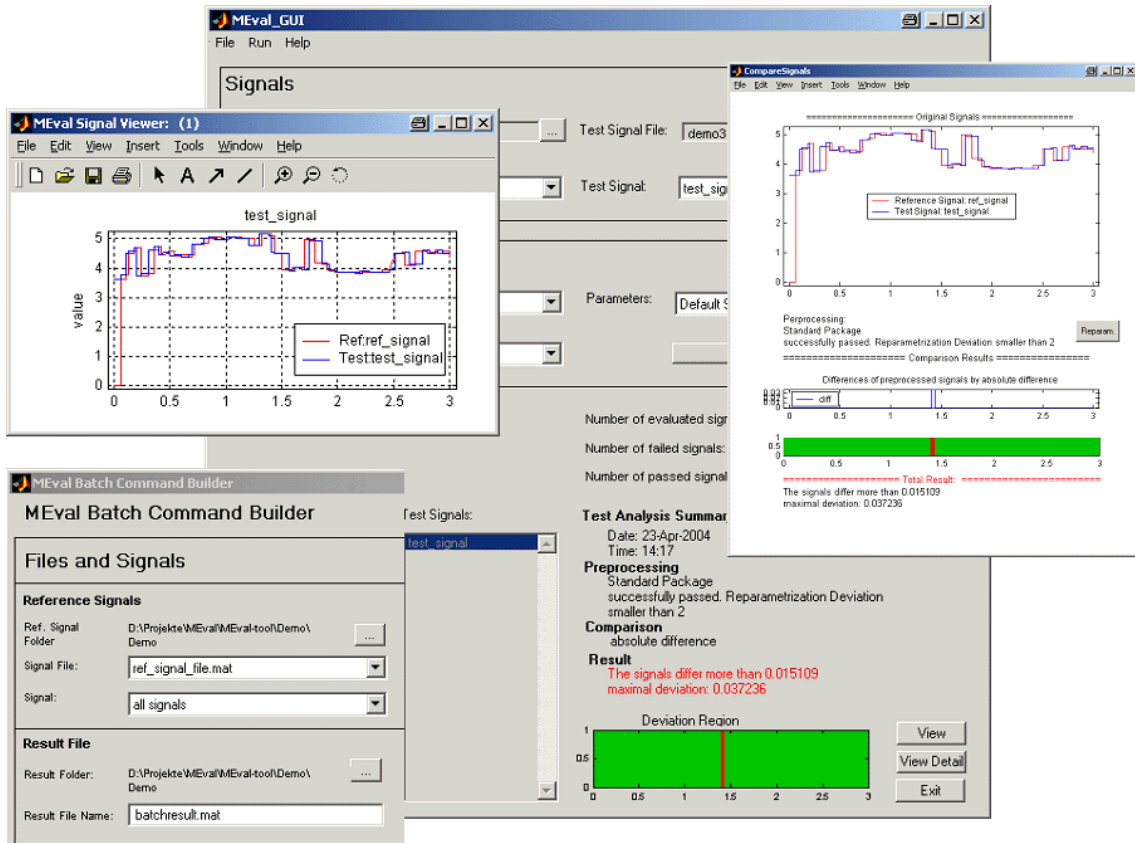


Abbildung 6 : MEval zur automatischen Testauswertung

Die Algorithmen von MEval wurden nun in die Testumgebung MTest integriert, so dass auch die Auswertung der generierten Testfälle automatisch läuft. Nach dem Testdurchlauf in MTest, sei es MiL, SiL oder auch PiL, werden die Ergebnisse automatisch mit den Referenzdaten, z.B. den Resultaten der Reactis Simulation, verglichen und deren Übereinstimmung beurteilt.

## 5. Abschlussbetrachtung

Der Wettbewerbsdruck führt Hersteller elektronischer Systeme dazu, über neue Funktionen der Software einen Vorsprung zu erzielen und sich Wettbewerbsvorteile zu sichern. Die eingebettete Software in elektronischen Systemen wird im Zuge des technischen Fortschritts immer komplexer und gleichzeitig steigen die Anforderungen an Qualität und Zuverlässigkeit. Der Test ist und bleibt in der absehbaren Zukunft das wichtigste Mittel zur Sicherung der Qualität und Zuverlässigkeit der Software.

Ein hoher Aufwand der entwicklungsbegleitenden Tests liegt in der Testauswertung; er wächst schnell mit der Anzahl der durchzuführenden Testszenarien. Diese Arbeit kann durch die Integration spezieller Auswertungstools in die Testumgebung, wie das Werkzeug MEval in MTest, deutlich optimiert werden. Regressions- und Back-To-Back-Tests können so automatisiert im Batch-Betrieb durchgeführt und evaluiert werden. Zugleich erlaubt MTest die Messung verschiedener erreichter Überdeckungskriterien der Testdurchläufe. Nach ausgiebigen funktionalen Tests kann so leicht überprüft werden, ob die zur Absicherung der Codegenerierung benötigten Kriterien erfüllt wurden. Ist dies noch nicht der Fall, bietet es sich an, Testdatengeneratoren einzusetzen. Eine Kopplung des Werkzeugs Reactis in die Testumgebung MTest rundet dann den Pro-

zess ab. Durch *Reactis* können gezielt Testdaten gesucht und generiert werden, die die noch offenen Lücken füllen. Die Test- und Simulationsdaten können dann in *MTest* importiert werden, dort ausgeführt und anschließend durch *MEval* ausgewertet werden, so dass der Aufwand dieser verbliebenen Testaktivitäten drastisch reduziert werden kann.

## Literatur

- [1] Baresel, A., Conrad, M., Sadeghipour, S., Wegener, J.: The Interplay between Model Coverage and Code Coverage, 11. Europ. Int. Conf. on Software Testing, Analysis and Review (EuroSTAR 03), Amsterdam, NL, 2003.
- [2] Conrad, M., Fey, I., Sadeghipour S.: Systematic Model-Based Testing of Embedded Automotive Software - The MB<sup>3</sup>T Approach. ICSE 2004 workshop on Software Engineering for Automotive Systems (SEAS '04), Edinburgh, UK, 2004.
- [3] Conrad, M., Sadeghipour S., H.-W. Wiesbrock: Automatic Evaluation of ECU Software Tests, SAE World Congress, Detroit, USA, 2004.
- [4] Lamberg, K., Beine, M., Eschmann, M., Otterbach, R., Conrad, M., Fey, I.: Model-based testing of embedded automotive software using MTest, SAE World Congress, Detroit, USA, 2004.
- [5] MATLAB/Simulink/Stateflow (product information). The MathWorks Inc., [www.mathworks.com/products](http://www.mathworks.com/products).
- [6] MEval (product information). IT Power Consultants, [www.itpower.de/meval\\_e.html](http://www.itpower.de/meval_e.html)
- [7] MTest (product information). dSPACE GmbH, [www.dspace.de/ww/en/pub/products/sw/expsoft/mtest.htm](http://www.dspace.de/ww/en/pub/products/sw/expsoft/mtest.htm)
- [8] Reactis Tester (product information), Reactive Systems Inc., [www.reactive-systems.com](http://www.reactive-systems.com).
- [9] Sadeghipour, M. Lim, Einsatz automatischer Testvektorgenerierung im modellbasierten Test, Workshop Modellbasierte Qualitätssicherung im Rahmen der 35. Jahrestagung der Gesellschaft für Informatik "Informatik 2005 - Informatik Live!", Bonn.

## Kontaktadressen

Meike Lim, Sadegh Sadeghipour, Hans-Werner Wiesbrock  
IT Power Consultants  
Gustav-Meyer-Allee 25 /Geb. 12  
13355 Berlin

{meike.lim, sadegh.sadeghipour, hans-werner.wiesbrock}@itpower.de

Tel.: +49 - (0)30 - 46 79 98 43

Fax: +49 - (0)30 - 46 30 76 49



**IT Power**  
Consultants

## **Autoren**

Dipl. Ing. Meike Lim  
IT Power Consultants  
Gustav-Meyer-Allee 25 /Geb. 12  
13355 Berlin

Dr. Sadegh Sadeghipour  
IT Power Consultants  
Gustav-Meyer-Allee 25 /Geb. 12  
13355 Berlin

Dr. habil. Hans-Werner Wiesbrock  
IT Power Consultants  
Gustav-Meyer-Allee 25 /Geb. 12  
13355 Berlin