

Benutzungsorientiertes und modellzentriertes Testen im HiL-Testing

Das Finden der richtigen Testfälle ist für Tester, Testmanager und Projektverantwortliche eine entscheidende Aufgabe. Um das Lösen dieser Aufgabe zu systematisieren, hat die Audi AG in Projekten verschiedene Möglichkeiten geprüft. Am Ende hat die modellgetriebene Testfallgenerierung überzeugt: Das Generieren von Testfällen auf Basis von Benutzungsmodellen wird im Rahmen einer Promotion mit der Universität Erlangen-Nürnberg weiterentwickelt sowie das modellzentrierte Testen (.mzT) der Firma sepp.med GmbH.

1 Einleitung

Die Audi AG hat frühzeitig die Relevanz der Softwareentwicklung in der Automobilentwicklung erkannt. Das Testen als wesentlicher Bestandteil der Absicherung von zunehmend komplexen Funktionalitäten hat höchste Bedeutung. Die dabei verwendete Testmethodik ist durch Exam (Extended Automation Method) beschrieben. Die automatisierte Ausführung der Testfälle findet auf HiL (Hardware-in-the-loop)-Simulatoren statt. Exam sieht die manuelle Erstellung von einzelnen Testfällen vor. Damit ist die Ermittlung von Überdeckungskriterien und nicht geprüften Szenarien eine schwierige Aufgabe. Dennoch konnte die Aufgabe gelöst werden, durch die Einführung von Modellen und systematischer Ableitung von Testfällen aus diesen. Dennoch konnten diese Aufgaben durch die Einführung von Modellen und systematischer Testfallgenerierung gelöst werden.

2 Extended Automation Method (Exam) Version Exam 2.0

Die bei der Audi AG eingesetzte Testmethodik Exam (Extended Automation Method) definiert den Prozess, die Rollen und die eingesetzten Werkzeuge zur Durchführung der Testaktivitäten. Testfälle werden formal und plattformunab-

hängig in UML-Sequenzdiagrammen spezifiziert. Ausgehend von diesen wird automatisch plattformspezifischer Code unter Verwendung von Referenzbibliotheken in „Python“ zur Ausführung am Zielsystem generiert. Die Testfallspezifikationen, die Funktionsbibliotheken sowie die Testreports werden zentral in einer Datenbank abgespeichert, so dass die gemeinsame und geteilte Entwicklung von Testfällen und Testautomatisierungsfunktionalitäten unterstützt wird. Die Ausführung des Codes findet automatisiert auf HiL-Prüfständen statt. Reports zur Testergebnisbewertung werden automatisch erzeugt.

Allerdings muss jeder Testfall einzeln und manuell anhand der Spezifikationen erstellt werden, **Bild 1**. Damit ist es im Nachhinein schwierig zu ermitteln, warum bestimmte Testfälle definiert oder andere Abläufe nicht getestet wurden. Darüber hinaus ist die Ermittlung und Einbeziehung von Überdeckungskriterien sowie Testmanagementindikatoren zur Testfallbestimmung im Vorfeld kaum möglich. Hinzu kommt die aufwändige manuelle Wartung der existierenden Testfälle bei Änderungen in den Anforderungen. Um diese Schwächen zu beheben, wurden bei der Audi AG modellgetriebene Ansätze zur Testmodellbeschreibung und Testfallgenerierung verfolgt. Zwei Ansätze werden im Folgenden beschrieben:

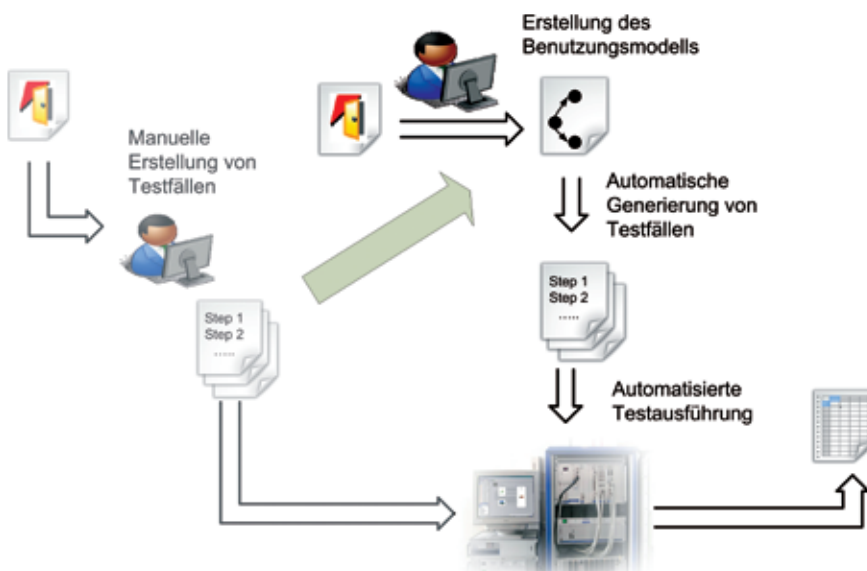


Bild 1: Ehemaliger Prozess mit manueller Testfallerstellung und Prozess mit modellgetriebener Testfallgenerierung

Die Autoren



Gerhard Kiffe ist verantwortlich für das Testen, die Testmethodik und Testautomatisierung auf HiL-Simulatoren bei der Audi AG in Ingolstadt.



Sebastian Siegl entwickelt als Doktorand modellbasierte Verfahren zur funktionalen Absicherung bei der Audi AG in Ingolstadt.



Florian Prester arbeitet als Geschäftsführer mit Fokus auf die Bereiche Automotive, Automation & Systems bei der sepp.med GmbH in Röttenbach.



Dr. Martin Beisser ist Abteilungsleiter bei der sepp.med GmbH in Röttenbach.



Martin Seel arbeitet als Senior Consultant und Key-account Manager Audi bei der sepp.med GmbH in Röttenbach.

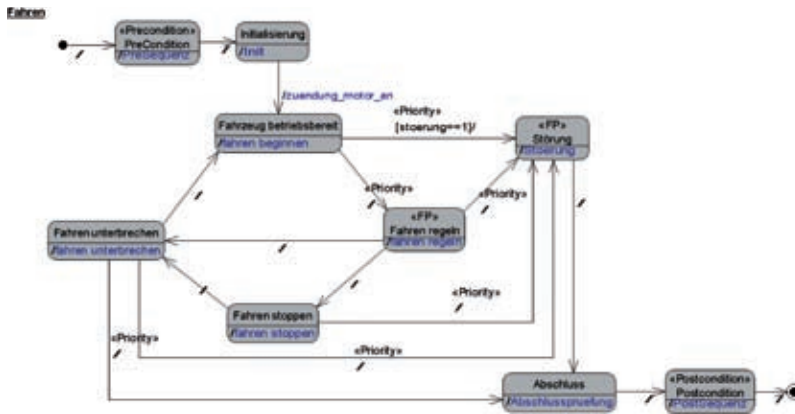


Bild 2: .mzT-Modell – inklusive „Conditions“, Strategien, Prioritäten und benannten Pfaden

- Der erste Ansatz wurde im Rahmen einer Diplomarbeit erfolgreich eingeführt. Er basiert auf Benutzungsmodellen und sieht die systematische Unterscheidung zwischen Benutzungs- und Systemverhaltensbeschreibung vor. Im Rahmen einer Promotion wird dieser Ansatz weiterentwickelt.
- Der zweite Ansatz ist das modellzentrierte Testen (.mzT) der Firma sepp.med GmbH, das im Rahmen erster Projekte bei Audi erfolgreich eingesetzt wurde. Dieser Ansatz verwendet .mzT-Testmodelle auf Basis der UML.

3 Benutzungorientiertes Testen

Ein Benutzungsmodell beschreibt alle Benutzungsszenarien, die am „System-under-Test“ (SuT) ausgeführt werden können. Die Verknüpfung von Anforderungen und Referenzverhalten lassen sich auf unterschiedliche Arten realisieren, wobei der Test-Designer die Möglichkeit hat, die für seine Testaufgabe beste Lösung zu wählen. Testmanagementinformationen, wie mittlere Testfalllängen oder Überdeckungskriterien, können bereits vor der Testfallgenerierung bestimmt werden. Ergebnisse von bereits stattgefundenen Testläufen können bei der Generierung von neuen Testfällen mit berücksichtigt werden, so dass der Test-Designer neben dem SuT auch die bereits erzielten Testergebnisse bei der Auswahl neuer Testfälle im Blick hat. Damit bildet das Benutzungsmodell die Basis zur Entscheidung, welche und wann Testfälle ausgeführt werden sollen.

Eine besondere Eigenschaft von Benutzungsmodellen ist die übersichtliche und klare Darstellung des Testthemas, da es sich genau auf diese Aufgabe konzentriert. Zur Testfallgenerierung können unterschiedliche Strategien eingesetzt werden. Standard-Algorithmen zur Testfallgenerierung wie solche zur zufälligen Testfallgenerierung, minimalen Kantenüberdeckung oder zur Generierung aller möglichen Pfade, sind auf Benutzungsmodelle anwendbar. Allerdings sind diese im Bereich des HiL-Testen nicht optimal, da nicht sichergestellt ist, dass der HiL optimal ausgenutzt wird. In einer Diplomarbeit wurden daher Algorithmen entworfen, die die Testfallgenerierung aus Benutzungsmodellen unter Verwendung von Kriterien und Testzielen ermöglichen. So unterstützen die Strategien den Test-Designer dabei, sein

Know-how optimal einzusetzen. Dazu können zum Beispiel Kriterien vorgegeben werden, beispielsweise Überdeckung von verknüpften Anforderungen unter Berücksichtigung von Testzielen. Zu den Zielen zählen beispielsweise Kriterien wie kürzeste Ausführungsdauer am HiL oder Pfade, welchen ein höheres Risiko zugeordnet wird.

Der Test-Designer kann sich damit auf die Beschreibung der zu testenden Funktionalität konzentrieren. Die automatisierte Generierung von Testfällen für bestimmte Testziele wie Stress-tests wird durch Strategien unterstützt. Hierzu können Informationen wie Anforderungen, Ausführungszeiten, Risikoeinstufungen zum Beispiel in Tagged Values hinterlegt werden. Dadurch kann der Test-Designer sein gesammeltes Know-how einbringen und die Generierung entscheidend und bewusst beeinflussen.

Die Testmethodik basierend auf Benutzungsmodellen und die entwickelten Strategien werden nun am Lehrstuhl für Informatik 7 der Universität Erlangen-Nürnberg im Rahmen einer INI.FAU-Promotion, einer Kooperation zwischen der Universität Erlangen-Nürnberg und der Audi AG, weiter entwickelt.

4 Modellzentriertes Testen „mzT“

Der zweite Ansatz im Rahmen der Optimierung der Testmethodik, der erfolgreich bei der Audi AG erprobt wurde,

Strategien:

- Full Path – alle Pfade durch das Modell
- Random – n zufällige Pfade durch das Modell
- Named Path – ein/mehrere benannte Pfade durch das Modell
- Full Transition – alle Kanten

Managementinformationen:

- Hierarchien
 - o unterschiedliche Strategien auf Diagrammebene
 - o unterschiedliche Schleifenbehandlung
- Bezeichner
- Loop-Begrenzung
- Testfalllänge
- Prioritäten
- Bedingungen (Pythoninterpreter zur Laufzeit (während der Generierung))

Bild 3: Strategien und Steuerungselemente zur Testfallerzeugung

ist das modellzentrierte Testen der Firma sepp.med GmbH. Auch hier beschreiben die Modelle alle möglichen Anwenderaktionen, um daraus die Testfälle abzuleiten. Die Modelle nehmen auch Informationen auf, die zur Steuerung der Generierung dienen. Beispielsweise können Übergänge und Zustände mit Prioritäten oder Bedingungen belegt werden, um sie je nach Testphase zu berücksichtigen. .mzT ist unabhängig von Modellierungs- beziehungsweise Testmanagement- und Testausführungswerkzeugen. Somit lässt sich das Testdesign systematisieren, ohne bestehende Testsysteme oder Prozesse verändern zu müssen.

4.1 Das .mzT-Testmodell

Zur Beschreibung der .mzT-Testmodelle werden Aktivitäts- oder UML-Zustandsdiagramme herangezogen. Dabei kommt es nicht auf die Semantik der UML an, sondern auf die Möglichkeit, Systeme aus Testersicht zu beschreiben und auf hinterlegte Informationen wieder zugreifen zu können. **Bild 2** zeigt ein .mzT-Testmodell. Dabei beschreibt jeder gültige Weg vom Startknoten bis zum Endknoten einen Testfall. Wichtig ist hierbei der logische Aufbau der Testmodelle. Ähnlich zu den Benutzungsmodellen wird das System aus Benutzersicht dargestellt und zusätzlich mit „Testers Mindset“ versehen. Das bedeutet, ein Software- oder System-Tester versteht die Modelle mit allen Informationen und Stimuli, so dass das SuT vollständig – inklusive Stresstests und „Bad Cases“ – geprüft werden kann.

Dazu werden in einem .mzT-Testmodell die Komponenten in zwei Klassen eingeteilt: Aktionskomponenten (Testschritte, Aktionskanten und Übergangskanten) und Verifikationspunkte. Um die Modelle übersichtlich und intuitiv zu halten, werden hierarchische Modelle verwendet. Die Übersichtlichkeit der erzeugten Modelle ist dabei entscheidend, denn das Modell soll nicht nur zur Generierung von Testfällen herangezogen werden, es dient viel mehr als zentrales Kommunikationsmedium. Der Test visualisiert das SuT aus seiner Sicht und kann damit aktiv die Anforderungsdefinition unterstützen und der Entwicklung seine Sicht des neuen Systems anschaulich darlegen.

Embedded Tester^{BASE}

DAS Starter-Kit für den automatisierten Test mit TargetLink®

- /// Testmanagement
- /// Import/Export von Testvektoren
- /// Autom. Testumgebungsgenerierung
- /// Autom. Testausführung & Regression
 - /// MiL, SiL, PiL
- /// Datenstromvergleich
 - /// Benutzerdefinierte Toleranzen
 - /// FLP vs. FXP Testen
- /// Autom. Debugging-Unterstützung
- /// Test-/Coverage-Reportgenerierung
- /// Bestes Preis-/Leistungsverhältnis!

Besuchen Sie uns auf der:
Elektronik im Kraftfahrzeug 2009
7. - 8. Okt. 2009 in Baden-Baden
Stand 82

Just in Case!



www.btc-es.de



Piezo-Aktoren

Diese und weitere Produkte und Dienstleistungen von Spezial-Anbietern finden Sie direkt unter

www.BranchenIndex.de

Die B2B-Suchmaschine für Industrie und Wirtschaft

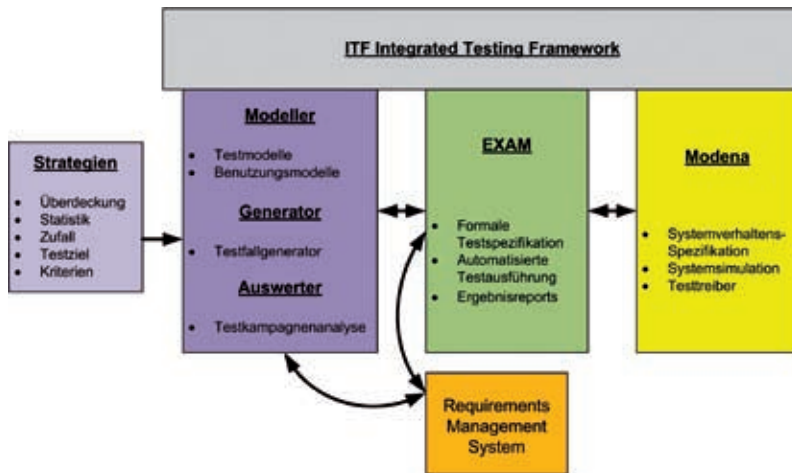


Bild 4: Integrated Testing Framework (ITF)

4.2 Generierung und Testmanagementinformationen

Für die Generierung werden verschiedenste Strategien unterstützt. Es besteht die Möglichkeit, zahlreiche Strategien miteinander zu kombinieren und mit Hilfe von Managementinformationen die Generierung entscheidend und bewusst zu beeinflussen, Bild 3.

5 Erzielte Verbesserungen

Bei der Audi AG konnten durch die Verwendung von Benutzungsmodellen und .mzT-Testmodellen bereits bei der Erstellung Vorteile erzielt werden. So findet bei der Erstellung des Benutzungsmodells oder .mzT-Testmodells eine Analyse der Anforderungen statt. Dadurch können bereits deutlich vor der Ausführung des ersten Testfalls Fehler in den Anforderungen identifiziert werden.

Ein weiterer Vorteil ergab sich vor der Testfall-Generierung. Anhand des Benutzungsmodells oder .mzT-Testmodells ist es nun möglich, Überdeckungsmaße zu definieren und Metriken zu der Testaufgabe zu erzeugen. Der nächste Vorteil ergibt sich durch die Möglichkeit Testfälle durch Wahl passender Strategien zu generieren – wodurch jetzt Testfälle erzeugt werden, an die vorher nicht gedacht wurde. Der zeitliche Aufwand der modellgetriebenen Testfallgenerierung war mit der manuellen Erstellung vergleichbar, allerdings konnten durch die modellgetriebene Vorgehensweise zusätzliche Vorteile erzielt werden.

6 Integrierte Testumgebung – Integrated Testing Framework (ITF)

Im Rahmen des Integrated Testing Framework (ITF) soll die Zusammenarbeit von Exam 2.0, welches als Freeware zur Verfügung steht, mit weiteren Komponenten realisiert werden. Der Kerngedanke von ITF ist die optimale Zusammenarbeit aller Komponenten in einem gemeinsamen Framework. Daher hat man sich für eine Lösung auf Basis einer „Rich Client Platform“ entschieden. Alle Komponenten müssen diesem Ansatz folgen. Verschiedene Einrichtungen und Firmen arbeiten gemeinsam an diesen Komponenten, welche auf den gesamten VW-Konzern ausgerollt werden sollen.

Die formale Testfallspezifikation, automatisierte Testausführung und Auswertung wird durch Exam 2.0 realisiert. Im Moment sind neben Exam 2.0 zwei weitere Komponenten des ITF in der Planung. Eine Möglichkeit zur Beschreibung und Simulation von Systemverhalten wird von Modena, einer Simulationsumgebung basierend auf Systemverhaltensmodellen der Firma Berner und Mattner, bereitgestellt. Ein Modellierungswerkzeug zur Erstellung von Benutzungsmodellen bildet die dritte Komponente. Aus diesen wird eine systematische Ableitung von Testfällen unter Verwendung verschiedener Strategien ermöglicht, Bild 4. Im Rahmen einer Promotion bei der Audi AG in Zusammenarbeit mit dem Lehrstuhl für Informatik 7 der Universität Erlangen-Nürnberg werden Testfall-Generierungsstrategien auf

Basis von Benutzungsmodellen entwickelt, die die automatische Generierung von optimierten Testfällen unter Berücksichtigung von Testzielen ermöglichen.

Dazu werden Benutzungsmodelle oder .mzT-Testmodelle erstellt. Aus diesen können anschließend zum Beispiel mit Hilfe des Testfallgenerators .getmore der Firma sepp.med GmbH Testfälle generiert werden. Die Testfälle können als Testrequirements in DOORS importiert und mit den Funktionsrequirements verknüpft werden. Dadurch kann eine durchgängige Anforderungsverfolgbarkeit vom Requirement über das Modell bis zum Testfall gewährleistet werden. Geplant ist ein weiteres Plugin welches eine Analyse der Ergebnisse von Testdurchläufen ermöglicht, um diese bei der Generierung von neuen Testfällen mit zu berücksichtigen.

7 Zusammenfassung

Durch das Integrated Testing Framework (ITF) wird die Zusammenarbeit von den am Testen beteiligten Komponenten in einem gemeinsamen Framework realisiert. Offenheit und Erweiterbarkeit für weitere Komponenten ist ein Grundkonzept des ITF, so dass die Erweiterbarkeit jederzeit möglich ist. Die Erstellung von Benutzungsmodellen und .mzT-Testmodellen kann durch eine Modellierungskomponente ermöglicht werden. Das automatisierte und systematische Generieren von Testfällen ist zum Beispiel mittels .getmore möglich. Diese können dann durch Exam, entweder am HiL auf realen ECUs oder unter Modena auf Systemmodellen zur Ausführung gebracht werden. Strategien zur Testfallgenerierung werden im Rahmen einer Promotion in Zusammenarbeit mit der Universität Erlangen-Nürnberg entwickelt.

Mit der modellgetriebenen Testfallgenerierung aus Benutzungs- und .mzT-Testmodellen wurde bei der Audi AG eine Testmethodik zur Testfallfindung und funktionalen Absicherung für zunehmend komplexe Systeme gefunden. ■

Download des Beitrags unter
www.ATZonline.de

ATZ
online

ATZ
elektronik

Read the English e-magazine.
Order your test issue now:
SpringerAutomotive@abo-service.info