Automatisiertes Testen asynchroner nichtdeterministischer Systeme mit Daten

vorgelegt von
Diplom-Informatiker
Dirk Seifert

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Sahin Albayrak
Berichtende:	Prof. Dr. Stefan Jähnichen
	Prof. Dr. Ina Schieferdecker
	PD Dr. Thomas Santen

Tag der wissenschaftlichen Aussprache: 20. Juli 2007

Berlin 2007
D 83

Berichte aus der Informatik

Dirk Seifert

Automatisiertes Testen asynchroner nichtdeterministischer Systeme mit Daten

D 83 (Diss. TU Berlin)

Shaker Verlag
Aachen 2007

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Zugl.: Berlin, Techn. Univ., Diss., 2007

Printed in Germany.

ISBN 978-3-8322-6579-3
ISSN 0945-0807

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Zusammenfassung

Eingebettete Systeme setzen sich aus Hardware- und Softwarekomponenten zusammen, die asynchron miteinander kommunizieren und nichtdeterministisches Verhalten aufweisen können. Dies macht eine umfassende und systematische Prüfung mit manuellen Techniken nahezu unmöglich. Auch automatisierte Prüftechniken können mit Asynchronität und Nichtdeterminismus bislang nur unzureichend umgehen. Der breite Einsatz eingebetteter Systeme im täglichen Leben und das blinde Vertrauen in deren Funktionsfähigkeit erfordern jedoch eine zuverlässige und umfassende Qualitätssicherung.

Die vorliegende Arbeit beschäftigt sich mit dem automatisierten Testen asynchroner nichtdeterministischer Systeme mit Daten. Zur Beschreibung des reaktiven Verhaltens von Komponenten technischer Systeme werden Zustandsmaschinen der *Unified Modeling Language* in reduzierter Form verwendet. Ausgehend von einer solchen Zustandsmaschine werden automatisiert Testfälle für einen Konformitätstest abgeleitet und gegen das zu testende System ausgeführt. Das korrekte, mögliche Verhalten wird für ausgewählte Eingaben a priori berechnet und in einem Testfall gespeichert. Die Speicherung ermöglicht eine wiederholte Ausführung der Testfälle nach Änderungen in der Entwicklung. Der Hauptbeitrag der Arbeit liegt einerseits in der vollständigen Formalisierung der reduzierten Zustandsmaschinen, die die wesentlichen syntaktischen Ausdrucksmittel und komplex strukturierte Daten umfassen, und andererseits in der Entwicklung eines praktikablen Ansatzes zum automatisierten Testen reaktiver technischer Systeme, deren Verhalten durch solche Zustandsmaschinen beschrieben werden kann. Das Testen solcher Systeme ist aus zwei Gründen eine besondere Herausforderung. Einerseits ist die Anzahl möglicher Eingabesequenzen und damit die Anzahl der möglichen Testfälle unendlich groß. Andererseits führen die asynchrone Kommunikation und der inhärente Nichtdeterminismus in Zustandsmaschinen für jede einzelne Eingabesequenz zu einem sehr komplexen Verhalten. Die präzise und widerspruchsfreie Interpretation dieses komplexen Verhaltens ist notwendige Voraussetzung für jegliche Art von Automatisierung, erfordert jedoch einen enorm hohen Berechnungsaufwand und ist damit nicht immer praktikabel. Um die Anzahl der Testfälle sinnvoll zu beschränken werden Testspezifikationen verwendet. In Testspezifikationen werden Art und Umfang der betrachteten Eingaben festgelegt und relevante Eingabesequenzen durch Nutzungsprofile beschrieben. Einfache Nutzungsprofile erlauben die probabilistische Auswahl von Eingaben oder die Vorgabe fester Eingabesequenzen. Komplexere Nutzungsprofile beschreiben das Verhalten der Umgebung explizit in Form von Zustandsmaschinen mit probabilistischen Zustandsübergängen und ermöglichen damit eine systematische und automatisierte Auswahl relevanter Eingaben. Durch ein Nutzungsprofil kann zum Beispiel die technische Umgebung, in die das System später eingebettet werden soll, explizit modelliert werden.

Die Bestimmung des möglichen korrekten Verhaltens für eine gegebene Eingabesequenz erfolgt im vorgestellten Testansatz in einer schrittweisen Ausführung der Zustandsmaschine. Da der Berechnungsaufwand mit zunehmender Länge der betrachteten Eingaben exponentiell ansteigt und deshalb eine Berechnung für typische Sequenzlängen unmöglich ist, können kürzere Eingabesequenzen mehrerer Testfälle kombiniert werden und für diese jeweils das korrekte

mögliche Verhalten bestimmt werden. Die Einfügung von Beobachtungspunkten nach jeder Eingabesequenz führt dabei zu einer erheblichen Reduzierung des Berechnungsaufwands für die folgenden Eingaben, wobei das mögliche korrekte Verhalten approximiert wird. Der Berechnungsaufwand steigt dabei linear mit der Anzahl der kombinierten Testfälle. Durch eine geeignete Wahl der Anzahl an Teilsequenzen, aus denen eine Eingabesequenz zusammengesetzt wird, kann hinsichtlich des Aufwands für die Testfallerzeugung und der Erkennungsrate der Testfälle ein guter Kompromiss erzielt werden.

Praktisches Ergebnis der Arbeit ist die prototypische Werkzeugumgebung TEAGER. Diese setzt sich auf der einen Seite aus einer Umgebung zur Testfallerzeugung und -ausführung und auf der anderen Seite aus einer Umgebung zur Simulation von Zustandsmaschinen zusammen. Letztere ermöglicht eine realitätsnahe Ausführung von Zustandsmaschinen, d. h. insbesondere, dass nichtdeterministisches und zeitlich variables (probabilistisches) Ausführungsverhalten simuliert werden kann. Zum einen kann damit eine Zustandsmaschine, z. B. die Spezifikation, schon in einer frühen Entwicklungsphase analysiert werden und zum anderen konnte der von mir entwickelte Ansatz zum Testen eingebetteter Systeme selbst validiert werden. Mit der Werkzeugumgebung TEAGER habe ich die Anwendbarkeit des Ansatzes anhand kleinerer Beispiele und zweier Fallstudien demonstriert. Die durchgeführten Untersuchungen zeigen, dass durch die Verhaltensapproximation mit der Komplexität und der Zustandsexplosion innerhalb von Zustandsmaschinen umgegangen, eine akzeptable Erkennungsrate erzielt und die Ablehnung korrekter Systeme vermieden werden kann.

Inhaltsverzeichnis

Abbildungsverzeichnis

Definitionsverzeichnis