BestMasters

Malte Schmitz

Verteilte Laufzeitverifikation auf eingebetteten Systemen

Logiken und Monitorkonstruktionen für asynchrone Prozesse

Springer Vieweg

BestMasters

Weitere Informationen zu dieser Reihe finden Sie unter
http://www.springer.com/series/13198

Mit „BestMasters“ zeichnet Springer die besten Masterarbeiten aus, die an renommierten Hochschulen in Deutschland, Österreich und der Schweiz entstanden sind. Die mit Höchstnote ausgezeichneten Arbeiten wurden durch Gutachter zur Veröffentlichung empfohlen und behandeln aktuelle Themen aus unterschiedlichen Fachgebieten der Naturwissenschaften, Psychologie, Technik und Wirtschaftswissenschaften.
Die Reihe wendet sich an Praktiker und Wissenschaftler gleichermaßen und soll insbesondere auch Nachwuchswissenschaftlern Orientierung geben.

Malte Schmitz

Verteilte Laufzeitverifikation auf eingebetteten Systemen

Logiken und Monitorkonstruktionen für asynchrone Prozesse

Mit einem Geleitwort von
Prof. Dr. Martin Leucker

Springer Vieweg

Malte Schmitz
Lübeck, Deutschland

BestMasters
ISBN 978-3-658-12851-7 ISBN 978-3-658-12852-4 (eBook)
DOI 10.1007/978-3-658-12852-4

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

Springer Vieweg

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Springer Vieweg ist Teil von Springer Nature
Die eingetragene Gesellschaft ist Springer Fachmedien Wiesbaden GmbH

Geleitwort

Industrie 4.0 bedeutet die automatische integrierte und kosteneffiziente Erstellung von Gütern. Essentiell für die Industrie 4.0 ist die sichere und korrekte Steuerung von Industrieanlagen. Im Rahmen seiner Masterarbeit hat Herr Schmitz die Grundlagen für eine derartige Steuerung mit Hilfe von innovativen Überwachungskonzepten zur Laufzeit von Industrieanlagen gelegt. Er hat seine Konzepte modellhaft mit Hilfe von LEGO Mindstorms visualisiert, um neben den theoretischen Grundlagen insbesondere auch die praktische Anwendung transparent machen zu können.

Prof. Dr. Martin Leucker

Direktor des Instituts für Softwaretechnik und Programmiersprachen
an der Universität zu Lübeck

Institut für Softwaretechnik und Programmiersprachen

Das Institut für Softwaretechnik und Programmiersprachen (ISP) ist ein international ausgewiesenes Institut der Universität zu Lübeck und ist spezialisiert auf die systematische Entwicklung nachweisbar sicherer und zuverlässiger Systeme.

Daher wird neben allgemeinen programmier- und softwaretechnischen Aspekten ein Schwerpunkt auf Formale Methoden gelegt. Konkret arbeitet das ISP an Verifikationsverfahren für Softwaresysteme, wie Model-Checking, Testen und Runtime-Verification, aber auch Techniken zur Spezifikation von funktionalen und nicht-funktionalen Eigenschaften von Systemen, ihren Komponenten und deren Kommunikation.

Das ISP wird seit August 2010 von Prof. Dr. Martin Leucker geleitet und ist in zahlreichen Verifikations- und Testprojekten in den Domänen Medical, Automotive und Industry Automation involviert, in denen ein Schwerpunkt auf den zertifizierbaren Nachweis der Sicherheit von Systemen gelegt wurde.

Darüberhinaus bietet das ISP Vorlesungen, Seminare und Praktika in den Bereichen Softwaretechnik und Verifikation an. Auf diese Weise fließen aktuelle Forschungsergebnisse direkt in die Ausbildung der Studenten ein.

Kurzfassung

In dieser Arbeit werden Varianten der linearen Temporallogik (LTL) für den Einsatz zur Laufzeitverifikation verteilter, asynchroner, eingebetteter Systeme und die zugehörigen Monitorkonstruktionen untersucht und implementiert. Dazu wird eine verteilte Temporallogik mit Vergangenheitsoperatoren (ptDTL) und eine auf dreiwertiger Temporallogik (LTL_3) basierende, verteilte Temporallogik (fDTL) sowie eine synchronisierte Variante dieser Logik (fSDTL) verwendet. Für die dreiwertigen Logiken werden neue Automatenmodelle entwickelt und mit bekannten Monitorkonstruktionen verglichen. Als eingebettete Systeme kommen dabei Roboter von LEGO Mindstorms zum Einsatz, die in dem C-Dialekt NXC programmiert werden. Im Rahmen dieser Arbeit wurde ein Scala-Programm zur Monitorinjektion über Programmtransformation entwickelt. Das Programm liest den C-Quelltext mehrerer Roboter ein, parst die in Kommentaren enthaltenen Annotationen und ergänzt die für die verteilte Laufzeitverifikation notwendigen Routinen im Quelltext. Anhand dieser Software werden die verschiedenen Monitorkonstruktionen und Techniken der Zustandsgenerierung im praktischen Einsatz verglichen.

Inhaltsverzeichnis

Abkürzungsverzeichnis

ABA	alternierender Büchi-Automat, engl. *a*lternating *B*üchi *a*utomaton
AFA	alternierender endlicher Automat, engl. *a*lternating *f*inite *a*utomaton
BA	Büchi-Automat, engl. *B*üchi *a*utomaton
BNF	Normalform kontextfreier Grammatiken, engl. *B*ackus–*N*aur *f*orm
DFA	endlicher Automat, engl. *d*eterministic *f*inite *a*utomaton
DTL	verteilte Linearzeit-Temporallogik, engl. *d*istributed *t*emporal *l*ogic
fDTL	verteilte Linearzeit-Temporallogik mit Zukunftsoperatoren, engl. *f*uture *d*istributed *l*inear *t*emporal *l*ogic
fDTL_ω	verteilte Linearzeit-Temporallogik mit Zukunftsoperatoren auf unendlichen Worten, engl. *f*uture *d*istributed *l*inear *t*emporal *l*ogic on *w*ords
FLTL	lineare Temporallogik auf endlichen Worten, engl. *f*inite *l*inear *t*emporal *l*ogic
FLTL_3	dreiwertige lineare Temporallogik auf endlichen Worten, engl. *3*-valued *f*inite *l*inear *t*emporal *l*ogic
FLTL_4	vierwertige lineare Temporallogik auf endlichen Worten, engl. *4*-valued *f*inite *l*inear *t*emporal *l*ogic
fSDTL	synchronisierte verteilte Linearzeit-Temporallogik mit Zukunftsoperatoren, engl. *f*uture *s*ynchronized *d*istributed *l*inear *t*emporal *l*ogic
fSDTL_ω	synchronisierte verteilte Linearzeit-Temporallogik mit Zukunftsoperatoren auf unendlichen Worten, engl. *f*uture *s*ynchronized *d*istributed *l*inear *t*emporal *l*ogic on *w*ords

LTL	lineare Temporallogik, engl. *l*inear *t*emporal *l*ogic
LTL_3	dreiwertige lineare Temporallogik, engl. *3*-valued *l*inear *t*emporal *l*ogic
NFA	nichtdeterministischer endlicher Automat, engl. *n*ondeterministic *f*inite *a*utomaton
NNF	Negationsnormalform, engl. *n*egation *n*ormal *f*orm
NXC	C-artige Programmiersprache, engl. *n*ot *e*xactly *C*
ptDTL	verteilte Linearzeit-Temporallogik mit Vergangenheitsoperatoren, engl. *p*ast-*t*ime *d*istributed *l*inear *t*emporal *l*ogic
ptLTL	lineare Temporallogik mit Vergangenheitsoperatoren, engl. *p*ast-*t*ime *l*inear *t*emporal *l*ogic
RLTL	reguläre Linearzeit-Temporallogik, engl. *r*egular *l*inear *t*emporal *l*ogic
TL	beliebige temporale Logik, engl. *t*emporal *l*ogic

Tabellenverzeichnis