it 5/2008

Beschreibung und Verifikation räumlicher und zeitlicher Eigenschaften mobiler Systeme

Specification and Verification of Spatio-Temporal Properties of Mobile Systems

Andreas Schäfer, European Patent Office, Rijswijk (Niederlande)

Zusammenfassung Dieser Aufsatz gibt einen Überblick über eine formale Methode zur Analyse mobiler Realzeitsysteme. Beispiele solcher Systeme sind Fahrzeugsteuerungen und mobile Roboter. Es wird eine spatio-temporale Logik entwickelt, mit der sowohl die Systeme modelliert als auch die gewünschten Eigenschaften beschrieben werden können. Für diese Logik werden die theoretischen Grundlagen wie Entscheidbarkeit und Axiomatisierbarkeit untersucht und daraus der Prototyp eines automatischen Verifikationswerkzeuges entwickelt. Die Anwendung wird anhand einer aus der Industrie stammenden Fallstudie demonstriert. ►►►

Summary This paper provides an overview over a formal method for the analysis of mobile real-time systems. Control systems for cars and trains as well as mobile robots are examples of such systems. We develop a spatio-temporal logic that is used to model both the systems and safety requirements. We investigate the theoretical foundations like decidability and axiomatisability and develop a prototype tool for the automatic verification based on these results. The application of this logic is exemplified with an industrial case study.

KEYWORDS D.2.4 [Software: Software Engineering: Software/Program Verification]; F.4.1 [Theory of Computation: Mathematical Logic and Formal Languages: Mathematical Logic]; Realzeitsysteme, mobile Systeme, spatio-temporale Logik, Model-Checking / real-time systems, mobile systems, spatio-temporal logics, model checking

1 Einleitung

Eingebettete Computersysteme findet man in allen Bereichen des täglichen Lebens, angefangen bei Waschmaschinen bis hin zu Assistenzsystemen für die Steuerung von Fahrzeugen. Gerade für die letztere Klasse von sicherheitskritischen eingebetteten Systemen muss sichergestellt sein, dass sie sich in jedem Fall korrekt verhalten. Da intensives Testen immer nur einzelne Abläufe eines Systems untersucht, kann man damit nur die Anwesenheit von Fehlern aber nicht deren Abwesenheit sicher zeigen. Eine Lösung bieten formale Methoden wie z. B. das Model-Checking [1]. Bei diesen Verfahren wird ein Modell des Systems erstellt und ein mathematischer Beweis erbracht, dass das System die geforderten Eigenschaften besitzt. Zur Beschreibung von Realzeitsystemen, bei denen Antwortzeiten große Bedeutung besitzen, wurden die formalen Methoden z. B. mit „Uhren“ erweitert, um die zeitlichen Aspekte beschreiben zu können. Bei mobilen Realzeitsystemen wie Steuerungssystemen für Eisenbahnen und Flugzeuge oder mobilen Robotern sind aber neben den zeitlichen Aspekten auch räumliche Aspekte für die Beschreibung wichtig, da sich sichere und unsichere Zustände wie in der hier untersuchten Fallstudie u. U. nur in der räumlichen Konfiguration der Agenten unterscheiden.

Die Dissertation von Dr. Andreas Schäfer wurde von der Carl von Ossietzky Universität Oldenburg für den GI-Dissertationspreis 2006 vorgeschlagen. Gutachter der an der Fakultät Informatik, Wirtschafts- und Rechtswissenschaften durchgeführten Promotion waren Prof. Dr. Ernst-Rüdiger Olderog, Universität Oldenburg, und Prof. Dr. Michael Reichhardt Hansen, Technical University of Denmark.

 © Oldenbourg Wissenschaftsverlag

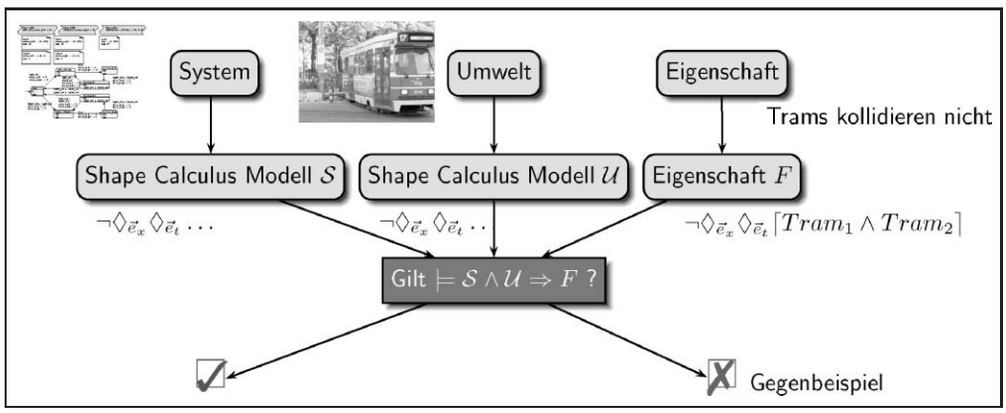

Bild 1 Verifikationsmethode.

Wir haben deshalb den Shape Calculus entwickelt, eine Erweiterung der Realzeitlogik Duration Calculus [3]. Mit dem Shape Calculus können sowohl die räumlichen Eigenschaften von Systemen – wie zum Beispiel minimale Abstände zweier Agenten – als auch die Realzeiteigenschaften – wie zum Beispiel garantierte Antwortzeiten – formalisiert werden. Wie in Bild 1 dargestellt, werden in unserem Verifikationsansatz zuerst sowohl das System, z. B. die Signalsteuerung, als auch Annahmen über die Umwelt, z. B. die Streckentopologie, und die Sicherheitseigenschaft im Shape Calculus formalisiert. Danach wird geprüft, ob die Sicherheitseigenschaft eine logische Folgerung daraus ist. Falls dies nicht der Fall ist, kann im Idealfall anhand eines Gegenbeispiels nachvollzogen werden, wieso die Eigenschaft nicht erfüllt ist.

2 Modellierung

Die Anwendung des Shape Calculus wird an folgender Fallstudie illustriert. Sie stammt aus dem UniForm Projekt [4] und wurde von dem Instustriepartner ElPro bereitgestellt.

Fallstudie

Aufgrund von Wartungsarbeiten ist bei einer normalerweise zweigleisigen Straßenbahnstrecke ein Abschnitt auf einer Seite gesperrt, so dass der gesamte Straßenbahnverkehr für beide Fahrtrichtungen über ein gemeinsam zu nutzendes Streckenstück geleitet werden muss. Dies ist in Bild 2 skizziert. Durch Signalanlagen muss sichergestellt werden, dass es innerhalb des Streckenstücks zu keinen Kollisionen kommt. Die Besonderheiten liegen in folgenden zusätzlichen Anforderungen:

(1) Es dürfen mehrere Straßenbahnen hintereinander in gleicher Richtung das Steckenstück durchfahren, ohne dass das entsprechende Signal zwischendurch rot ist.
(2) Eine Straßenbahn darf die Richtung innerhalb des Streckenstücks ändern, sofern sich keine Straßenbahn dahinter befindet. Dies wird durch den Fahrer durch Fahren auf Sicht festgestellt.

Für die Formalisierung der Sicherheitseigenschaft muss die räumliche Konfiguration der Straßenbahnen betrachtet werden. Aufgrund der ersten Anforderung reicht es nicht aus, zu verlangen, dass sich jeweils nur eine Straßenbahn im kritischen, gemeinsam genutzten Abschnitt befindet. Wegen der zweiten Anforderung reicht es ebenfalls nicht aus, zu verlangen, dass falls sich zwei Straßenbahnen in diesem Abschnitt befinden, diese auch gleiche Fahrtrichtungen haben.

Im Rahmen des UniForm-Projektes war es nicht möglich, die allgemeine Sicherheitseigenschaft der Kollisionsfreiheit zu formulieren. Ebenfalls nicht formalisiert werden konnte das sinnvolle Verhalten des Fahrers, wie in der zweiten Anforderungen angegeben. Von diesen Eigenschaften musste bei der Analyse abstrahiert werden [2]. Mit Hilfe des Shape Calculus kann sowohl das Verhalten der Tramfahrer als auch des Signalcontrollers formal beschrieben werden. Die gewünschte Sicherheitseigenschaft kann wie folgt formalisiert werden:

$$\neg\Diamond_{\vec{e}_x}\Diamond_{\vec{e}_t}\lceil Tram1 \wedge Tram2\rceil.$$

Diese Formel beschreibt formal, dass es nicht möglich ist, dass irgendwo $\Diamond_{\vec{e}_x}$ irgendwann $\Diamond_{\vec{e}_t}$ die erste und die zweite Tram den gleichen Raum belegen.

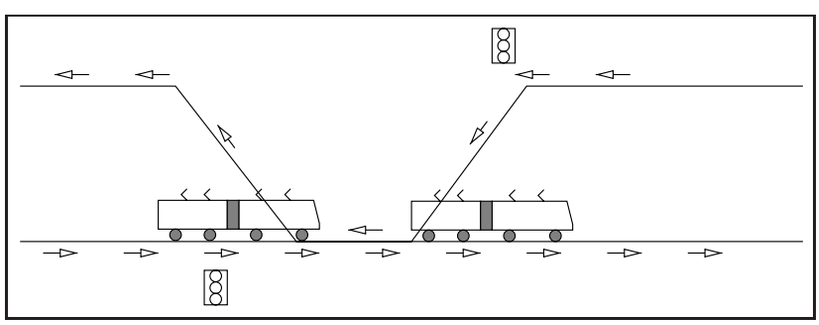

Bild 2 Sicherung eingleisiger Streckenabschnitte.

3 Analyse

Nachdem das System und die Umwelt mathematisch modelliert und zu erreichende Eigenschaften formalisiert sind, soll ein mathematischer Beweis erbracht werden, dass das System unter den Umweltannahmen die gewünschte Eigenschaft besitzt.

3.1 Automatische Verifikation

Wir untersuchen zuerst vollautomatische Verfahren. Durch einen Reduktionsbeweis auf das unentscheidbare Dominoproblem können wir zeigen, dass dieses Problem im Allgemeinen sowohl für kontinuierlichen Raum und kontinuierliche Zeit als auch für diskreten Raum und diskrete Zeit unentscheidbar ist und es somit ein allgemeines Verfahren nicht geben kann. Wir haben jedoch zwei relevante entscheidbare Teilklassen identifiziert, für die dieses Problem entscheidbar ist. Die eine Teilklasse betrachtet endlichen diskreten Raum und unendliche diskrete Zeit und die andere enthält eine syntaktische Einschränkung der Klasse der logischen Formeln. Die Einschränkung auf endlichen Raum ist dabei nicht gravierend, da reale Systeme meist ohnehin räumlich beschränkten Aktionsradius besitzen. Für diese Teilklasse haben wir ein automatisches Verifikationswerkzeug entwickelt und damit die oben beschriebene UniForm Fallstudie erfolgreich bearbeitet [5].

3.2 Beweissysteme

Theorembeweiser sind Werkzeuge, die den Benutzer bei der Erstellung von Beweisen unterstützen, aber meist nicht ganz vollautomatisch arbeiten. Für Theorembeweiser und für die Erstellung von Beweisen per Hand ist die Frage wichtig, ob es ein vollständiges und korrektes Beweissystem gibt. Wir haben gezeigt [6; 7], dass dies zwar im Allgemeinen nicht existiert, aber eine vollständige Axiomatisierung relativ zu einer mehrdimensionalen Intervalllogik möglich ist. Um die praktische Anwendung zu vereinfachen, haben wir außerdem Formelpattern für die Beschreibung häufiger Eigenschaften – wie z. B. Bewegungen – angegeben.

4 Fazit

In der Dissertation wird eine spatiotemporale Logik zur Beschreibung mobiler Realzeitsysteme, der Shape Calculus, entwickelt und untersucht. Zu den grundlegenden Eigenschaften gehören Unentscheidbarkeit und Nicht-Axiomatisierbarkeit der vollen Logik, eine relative Axiomatisierung und die Angabe zweier entscheidbarer Teilklassen. Für eine dieser Teilklassen existiert das automatische Verifikationswerkzeug MoDiShCa. Mit diesem Werkzeug wird die Benutzung der Logik für eine aus der Industrie stammende Fallstudie demonstriert. Offene Forschungsfragen bestehen in der Verbesserung der Anwendbarkeit für Nicht-Logiker und in der Verbesserung der automatischen und halbautomatischen Verifikationsverfahren, um größere Fallstudien handhaben zu können. Neuere Arbeiten untersuchen, wie sich das Beweisverfahren des natürlichen Schließens auf den Shape Calculus übertragen lässt.

Literatur

[1] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 2000.

[2] H. Dierks. *Specification and Verification of Polling Real-Time Systems.* PhD thesis, University of Oldenburg, July 1999.

[3] M. R. Hansen and Zhou Chaochen. *Duration Calculus: A Formal Approach to Real-Time Systems.* EATCS: Monographs in Theoretical Computer Science. Springer, 2004.

[4] B. Krieg-Brückner, J. Peleska, E.-R. Olderog, and A. Baer. The UniForM Workbench, a Universal Development Environment for Formal Methods. In: J. M. Wing, J. Woodcock, and J. Davies, editors, *FM'99 – Formal Methods*, vol. 1709 of *LNCS*, pp. 1186–1205. Springer, 1999.

[5] J.-D. Quesel and A. Schäfer. Spatio-Temporal Model Checking for Mobile Real-Time Systems. In: K. Barkaoui, A. Cavalcanti, and A. Cerone, editors, *Theoretical Aspects of Computing, ICTAC 2006*, vol. 4281 of *LNCS*, pp. 347–361. Springer, 2006.

[6] A. Schäfer. A Calculus for Shapes in Time and Space. In: Z. Liu and K. Araki, editors, *Theoretical Aspects of Computing, ICTAC 2004*, vol. 3407 of *LNCS*, pp. 463–478. Springer, 2005.

[7] A. Schäfer. Axiomatisation and Decidability of Multi-Dimensional Duration Calculus. In: *Information and Computation*, 205(1):25–64, 2007.

Dr. rer. nat. Andreas Schäfer studierte Informatik an der Carl von Ossietzky Universität in Oldenburg. Nach Abschluss des Diploms im Jahr 2002 war Andreas Schäfer als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe von Prof. Dr. Ernst-Rüdiger Olderog in Forschung und Lehre tätig. In dieser Arbeitsgruppe promovierte er 2006 mit Auszeichnung mit der Arbeit „Specification and Verification of Mobile Real-Time Systems". Seit April 2007 ist Andreas Schäfer beim Europäischen Patentamt als Patentprüfer in den Bereichen Betriebssysteme, Verteilte Systeme und Sicherheit von IT-Systemen tätig.

Adresse: European Patent Office, Patentlaan 3–9, 2288 EE Rijswijk, Niederlande, Tel.: +31 70 340 2841, E-Mail: schaefer@informatik.uni-oldenburg.de