Verifikation zeitlicher Anforderungen in automotiven komponentenbasierten Software Systemen

Matthias Gehrke, Martin Hirsch,
Wilhelm Schäfer
Software Quality Lab (s-lab)
Universität Paderborn
D-33095 Paderborn
[mgehrke|mahirsch|wilhelm]@upb.de

Oliver Niggemann, Dirk Stichling

dSPACE GmbH
Technologiepark 25
D-33100 Paderborn
[ONiggemann|DStichling]@dspace.de

Ulrich Nickel
Hella KGaA
Rixbecker Str. 75
D-59552 Lippstadt
Ulrich.Nickel@hella.com

Die in Kraftfahrzeugen eingesetzten Steuergeräte werden insbesondere durch den verstärkten Einsatz von Software immer leistungsfähiger. Darüber hinaus wird durch einen massiven Ausbau ihrer Vernetzung untereinander wesentlich weitergehende Funktionalität und damit zusätzlicher Komfort und mehr Sicherheit möglich und realisiert. Ein Beispiel für diesen Trend sind moderne Fahrerassistenzsysteme. Für einen Automobil-Zulieferer wie die Hella KGaA Hueck & Co führt dieser Trend zu immer größeren Projekten und vor allem immer umfangreicher werdenden Komponentenspezifikationen (insb. Softwarekomponenten). Bei Herstellern wird der Aufwand für die Integration der einzelnen Steuergeräte dementsprechend immer aufwendiger und auch schwieriger. Ein wesentliches Problem dieser Integration besteht darin, dass die einzelnen Steuergeräte in ihrer Funktionalität zwar spezifiziert werden, dass aber oft eine genaue Spezifikation der zeitlichen Abläufe, Restriktionen und insbesondere ihrer Vernetzung fehlt oder nur in Form von natürlich sprachlichem Text ausgedrückt wird. Selbst wenn diese Spezifikation in formaler Notation vorliegt, sind keine komponentenübergreifenden Prüfungen auf Widerspruchsfreiheit oder Vollständigkeit möglich. Bei der Integration der Steuergeräte, d.h. beim Test, der typischerweise hohen Aufwand und entsprechende Kosten bedeutet, stellt sich dann erst heraus, ob die in den einzelnen Steuergeräten realisierten Funktionen und ihre zeitlichen Randbedingungen miteinander kompatibel sind und nicht sogar zu Fehlfunktionen führen (können).

Unser Ansatz: Wir schlagen ein modellbasiertes Vorgehen vor, das die Spezifikation der Funktionalität und der zeitlichen Randbedingungen für Steuergerätesoftware unterstützt. Diese Funktionalitäten und Randbedingungen basieren auf einer AUTOSAR-kompatiblen Komponentenarchitektur und werden mit Hilfe zeitbehafteter Automaten beschrieben. Darüber hinaus soll eine komponentenübergreifende automatische Überprüfung von zeitlichen Anforderungen an das Gesamtsystem auf der Basis der Modelle der einzelnen Steuergeräte ermöglicht werden. Da diese automatische Überprüfung bereits weit vor der Testphase durchgeführt wird, kann der Aufwand in der Testphase signifikant reduziert werden.

Die Architekturmodellierung eines automotiven Softwaresystems in [GH+06] basiert auf einem Komponentenansatz gemäß der AUTOSAR-Spezifikation[1]. Die damit modellierten Software-Komponenten kapseln Teilfunktionalitäten oder sogar die vollständige Funktionalität eines Steuergeräts oder Steuergerätenetzwerkes. Das Verhalten einer einzelnen Komponente wird mit Hilfe von „Automotive Specific Automata“ spezifiziert. Diese sind einfache Automaten die um Zeitannotationen erweitert wurden, wobei die Semantik über eine Abbildung auf Timed Automata definiert ist. Die formale Abbildung auf Timed Automata wird bei der Verifikation ausgenutzt, die mittels des Modelcheckers UPPAAL[2] durchgeführt wird.

Würde man zur Formulierung komplexer, komponentenübergreifender Eigenschaften, wie sie im Bereich Karosserieelektronik in der Automobilindustrie vorkommen, nur die in UPPAAL verwendete Timed-Computation-Tree-Logic verwenden, müsste man das Verhalten aller Automaten global in einem Automaten zusammenzufassen und nur eine globale Uhr verwenden. Dieses Vorgehen ist nicht sinnvoll, da dieser Produktautomat ein Verhalten beschreibt, welches durch entstehende Seiteneffekte möglicherweise nicht mehr mit dem spezifizierten Verhalten einzelner Automaten übereinstimmt. Um dieses Problem zu lösen wurde das Prinzip der szenariobasierten Verifikation angewandt. Hierbei wird abhängig von den zu prüfenden Eigenschaften ein so genannter „Master“-Automat automatisch generiert, welcher parallel zu den eigentlichen Automaten ausgeführt wird. Da dieser „Master“-Automat genau das minimal benötige Verhalten aller vernetzten Komponenten für die zu überprüfende Eigenschaft in einem Automaten zusammenfasst, ist hierdurch eine korrekte, effiziente Verifikation möglich.

Darüber hinaus ist die manuelle Konstruktion von temporallogischen Formeln zur präzisen Formulierung der zu prüfenden Eigenschaften sehr aufwendig und fehleranfällig. Deshalb ermöglicht unser Ansatz eine automatische Generierung, deren grundlegende Idee darin besteht, die verschiedenen Zeit-Anforderungen zu typisieren. Für jeden Typ ist es dann möglich, entsprechende Generierungsvorschriften zu definieren. Die Typen „Maximale Verarbeitungszeit“, „Exakte Ausführungszeit“, „Synchronizität“ und „Konditionale Anforderung“ konnten bei der Untersuchungen verschiedener Lasten-/Pflichtenhefte für den Bereich der Karosserieelektronik im Bereich der Automobilindustrie identifiziert werden. Für jeden Anforderungstyp existiert eine einfache Eingabemaske, in die der Anwender lediglich den Typ der Zeit-Anforderung und Eingabedaten, wie zum Beispiel die Start- und Endereignisse, eintragen muss.

[GH+06] M. Gehrke, M. Hirsch, P. Nawratil, O. Niggemann, und W. Schäfer: Scenario-Based Verification of Automotive Software Systems: In *Proc. 2nd Workshop on Modellbasierte Entwicklung eingebetteter Systeme (MBEES), Schloss Dagstuhl, Germany* (Holger Giese, Bernhard Rumpe, und Bernhard Schätz, eds.), Band Informatik-Bericht 2006-01, Technische Universität Braunschweig, Seiten 35-42, Januar 2006.

---

[1] http://www.autosar.org

[2] UPPAAL ist frei verfügbar (http://www.uppaal.com) und wird im s-lab bereits seit einiger Zeit erfolgreich eingesetzt.