

Modellierung und Analyse von Zeitanforderungen basierend auf der UML

Matthias Hagner und Michaela Huhn

Institut für Programmierung und Reaktive Systeme, TU Braunschweig
Mühlenpfordtstr. 23, 38106 Braunschweig
{hagner, huhn}@ips.cs.tu-bs.de

Abstract: Bei der Entwicklung von automotive Anwendungen gewinnt die frühzeitige Spezifikation und Überprüfung von Performanz- und Echtzeitanforderungen an Bedeutung, da aufgrund der Komplexität der Systeme und des verteilten Entwicklungsprozesses frühzeitige Festlegungen zwischen OEM und Zulieferer notwendig sind. Wir stellen existierende Modellierungs- und Analysetechniken für Zeitanforderungen vor und zeigen, wie diese Techniken in einen modellbasierten Entwicklungsprozess integriert werden können.

1 Einleitung

Komplexität und Umfang der durch Software realisierten Funktionen im Auto sind in den letzten Jahren kontinuierlich gestiegen. Die Systeme müssen hohe Qualitätsanforderungen insbesondere bei sicherheitsrelevanten Anwendungen wie Fahrerassistenzsystemen erfüllen, gleichzeitig aber kosteneffizient für eine Massenproduktion mit COTS-Hardware entwickelt werden. In diesem Spannungsfeld werden zunehmend in frühen Phasen modellbasierte Softwareentwicklungsmethoden eingesetzt. System- und Software-Architektur werden oft mit Modellen der UML [OMG03], der AADL [SAE04] oder mit Blockdiagrammen etwa mit MATLAB/Simulink beschrieben. Moderne Entwicklungsumgebungen unterstützen Simulation und prototypische Codegenerierung aus Modellen zur Konzept-Validierung, aber Modelle können auch für eine frühzeitige formale Analyse von Systemeigenschaften genutzt werden. Eine zentrale, zu analysierende Qualitätanforderung ist das Zeitverhalten, einerseits unter dem Gesichtspunkt der Performanz, die trotz kostenoptimierter Hardware erreicht werden soll, andererseits unter dem Aspekt des Echtzeitverhaltens, das als Teil sicherheitsrelevanter Funktionalität gewährleistet werden muss.

Allerdings werden Informationen für die Zeitanalyse wie z.B. Ausführungszeiten, Deadlines oder Zuordnung von Ressourcen bei der Architekturmodellierung nur partiell erfasst, weil Architektur- und Verhaltensmodelle oft in erster Linie die Struktur und funktionalen Anforderungen beschreiben. Hinzu kommt, dass die Modellierungsnotationen kaum Möglichkeiten bieten, Informationen über das Zeitverhalten und die Zeitanforderungen in die Modelle einfließen zu lassen. So unterstützt erst die Erweiterung des UML Standards um Profile, speziell das „UML-Profil für Schedulability, Performance and Time“ (SPT-Profil) [OMG02] Modellannotationen zur Beschreibung von Echtzeit-Systemen.

Für die Zeitanalyse gibt es eine Reihe von speziellen Werkzeugen (z.B. TIMES [FY04], SymTA/S [HHJ⁺05]), die Echtzeitverhalten und Performanz von Systemmodellen formal analysieren und Fehler, Deadlocks und Zeitüberschreitungen in frühen Phasen der Entwicklung aufdecken und damit beheben lassen. Allerdings hat jedes Analysewerkzeug seine eigene Notation zur Modellierung von Systemen, Zeitverhalten und Anforderungen, die sich nicht nur in Bezug auf das Format, sondern auch hinsichtlich der Modellierungskonzepte unterscheidet. Die Analysewerkzeuge konzentrieren sich auf die Untersuchung des Zeitverhaltens, für die vollständige Modellierung aller Gesichtspunkte eines Systems sind sie nicht vorgesehen. Es besteht also eine Kluft zwischen den gängigen Modellen zur Architektur- und Verhaltensmodellen im Entwurf und den Zeitanalyse-Modellen, die es im Sinne eines durchgängigen, effizienten Entwicklungsprozesses zu schließen gilt.

Einen technischen Baustein zur Überbrückung dieser Kluft liefern [HMPY04], [HV06], [Hag05] mit einer Transformation von UML-Modellen erweitert um Zeitannotationen gemäß des SPT-Profiles in die Eingabeformate verschiedener Zeitanalysewerkzeuge. Diese Arbeiten geben rigide Modellierungsrichtlinien vor, um eine vollautomatische Transformation in ein Zeitanalysewerkzeug zu ermöglichen. Selbst wenn diese Richtlinien den Entwickler nicht zu stark einschränken, führen die notwendigen Modellerweiterungen, z.B. gemäß des SPT-Profiles, dazu, dass die Übersichtlichkeit abnimmt. Entwurfsmodelle der System- und Softwarearchitektur sind die Basis für die Entwicklung. Zeitanalyse ist nicht ihr Hauptzweck, stellt aber eine von mehreren Qualitätsanforderungen dar. Außerdem erfordert die Analyse des Zeitverhaltens eine Reihe von Festlegungen zu Implementierungsdetails, die man zu einem Modell mit dem Ziel eines Proof of Concept hinzufügen kann, z.B. zu geschätzten Übertragungszeiten unter einer bestimmten Buskonfiguration, die man üblicherweise aber nicht als Designentscheidungen verstanden wissen möchte. Aus diesen Gründen schlagen wir vor, aus Architektur- und Entwurfsmodellen (halb-)automatisch ein Zeitanalyse-Modell in der gleichen Modellierungsnotation (also etwa basierend auf der UML, der AADL oder Simulink) zu konstruieren, das sich auf die für das Zeitverhalten relevanten Modellelemente konzentriert und für diese dann um die notwendigen Informationen zu Deployment, Scheduling und Zeitverhalten erweitert wird. Dieses Zeitanalyse-Modell wird sowohl von Anwendungsexperten als auch von Analyseexperten verstanden und kann zur Exploration von Architekturvarianten und zur Entwurfsvalidierung dienen. Das Zeitanalyse-Modell kann dann vollautomatisch in die Eingabeformate der verschiedenen Zeitanalysewerkzeuge transformiert werden. Durch die klare Trennung von Entwurfs- und Analysemodellen kann dann nach einem erfolgreichen Validierungsschritt entschieden werden, ob und welche Informationen aus der Zeitanalyse in das Entwurfsmodell übernommen werden sollen.

2 Das Zeitanalyse-Modell

Das Zeitanalyse-Modell ist ein UML-Modell, das genau die Modellelemente mit Annotationen enthält, die für eine Zeitanalyse relevant sind. Um diese Informationen darzustellen, werden Modellelemente in Standard UML-Diagrammen mit ausgewählten Elementen des SPT-Profiles annotiert. Die Verwendung der Diagramme und der Profilelemente unterliegt Regeln, um eine eindeutige, automatisch transformierbare Darstellung zu ermöglichen.

So werden mit Zustands- oder Sequenzdiagrammen Abhängigkeiten bzw. Reihenfolgen der Ausführung von Tasks visualisiert (siehe Abb. 1). Eine Verteilung von Funktionsmodulen auf Ressourcen wird mit Objektdiagrammen modelliert. Die Elemente werden als Objekte dargestellt und mit Stereotypen genauer spezifiziert (eine Prozessor-Objekt wird z.B. durch den Stereotyp „SAengine“, ein Task durch den Stereotyp „SAaction“ beschrieben). Um diese Elemente genauer beschreiben zu können, werden Tagged Values verwendet. So wird bei einem Task u.a. die Ausführungsdauer und die Priorität angegeben.

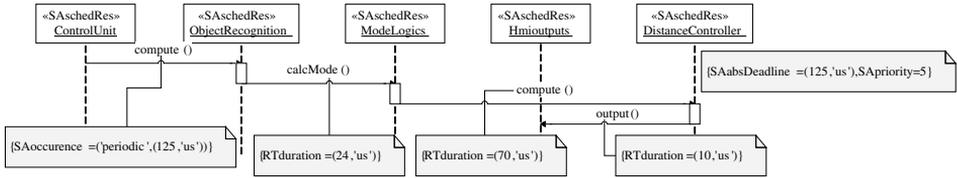


Abbildung 1: Ein Sequenzdiagramm mit Profile-Elementen

Eine vollständige Modellierung des Zeit- und Schedulingverhaltens ist für die Zeitanalyse erforderlich, da die Analysewerkzeuge nur ein vollständiges System überprüfen können. Zu den notwendigen Informationen gehören u.a. die Beschreibung der Tasks, funktionale Abhängigkeiten zwischen Tasks, die Verteilung, Ressourcenanforderungen, die Beschreibung der verwendeten Hardware und der verwendete Scheduling-Algorithmus.

Das Zeitanalyse-Modell dient als Basis und kann in die Formate von verschiedenen Zeitanalysewerkzeugen transformiert werden. Da die Regeln für das Zeitanalyse-Modell eindeutig beschrieben wurden, kann die Transformationen automatisiert werden. Die Analyse ist z.B. mit den Werkzeugen TIMES [FY04] oder SymTA/S [HHJ⁺05] möglich.

3 Einsatz des Zeitanalyse-Modells im Entwicklungsprozess

Architektur- und Entwurfsmodelle in einem modellbasierten Entwicklungsprozess für automotive Anwendungen bilden einerseits die Kommunikationsgrundlage zwischen OEM und Zulieferern und andererseits die Brücke zwischen informellen Anforderungen und konstruktiven codeerzeugenden Verfahren. Unserer Überzeugung nach werden diese Modelle überfrachtet und zweckentfremdet, wenn sie zusätzlich noch unmittelbar als Eingabe für eine Zeitanalyse dienen sollen.

Für die Zeitanalyse muss ein Modell um spezifische Spezifikationsdetails (s. Absatz 2) ergänzt werden, die zu einem frühen Zeitpunkt noch nicht festgelegt werden können oder sollen (z.B. Prioritäten einzelner Tasks, Verteilung auf Ressourcen . . .). Durch die Annotationen nimmt die Übersicht des Modells ab. Ein Sequenzdiagramm etwa, das mit Stereotypen und Tagged Values erweitert wurde, bietet eine gute Übersicht bezüglich des zeitlichen Verhaltens, wogegen das Nachvollziehen des dargestellten Ablaufes durch die Erweiterungen behindert wird. Um eine automatische Transformation in vorhandene Analysewerkzeuge zu gewährleisten, ist die Einhaltung rigider Modellierungsrichtlinien notwendig, die aber auf der Entwurfsebene als zu einschränkend einzustufen sind.

In unserem Ansatz wird das Zeitanalyse-Modell als Zwischenmodell zwischen Entwurfsmodellen und Zeitanalyse-Werkzeugen eingesetzt (siehe Abb. 2). Entwurfsmodelle werden durch eine Abstraktion umgesetzt und ergänzt, so dass sie den Regeln des Zeitanalyse-Modells genügen. Durch die Abstraktion wird das Entwurfsmodell auf die Elemente reduziert, die für die Zeitanalyse relevant sind. Abhängig vom Entwurfsmodell kann die Abstraktion halbautomatisch stattfinden. Informationen über Verteilung oder Abhängigkeiten können z.B. Komponenten-, Sequenz- und Zustandsdiagrammen entnommen und in das Zeitanalyse-Modell transformiert werden. Fehlende Informationen müssen dem Zeitanalyse-Modell manuell hinzugefügt werden. Durch die Abstraktion bleibt die Übersicht im Entwurfsmodell erhalten. Der Entwickler hat die Möglichkeit, verschiedene Varianten innerhalb des Zeitanalyse-Modells zu überprüfen, ohne dass das Entwurfsmodell verändert werden muss. Nachdem die Zeitanalyse abgeschlossen ist, können für den weiteren Entwurf notwendige Festlegungen in das Entwurfsmodell übernommen werden. Ein weiterer Vorteil des Zeitanalyse-Modells ist die Unabhängigkeit der Werkzeuge und Notationen für Entwurfsmodelle von den Einschränkungen der Modellierungsrichtlinien für Zeitanalyse-Annotationen. Insbesondere muss das Entwurfsmodell nicht auf UML basieren, sondern es kann auch MATLAB/Simulink verwendet werden.

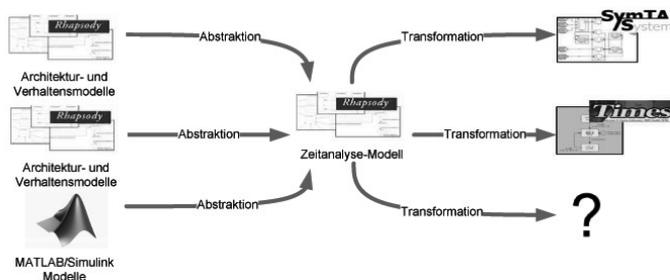


Abbildung 2: Abstraktion und Transformation vom Entwurfsmodell zu den Analysewerkzeugen

4 Fallstudie

Als Fallstudie wurde ein Adaptive Cruise Control System (ACC) betrachtet (angelehnt an [HACT04]), das die Geschwindigkeit des Fahrzeuges der Wunschgeschwindigkeit des Fahrers unter Berücksichtigung des Abstands und der Geschwindigkeit eines vorausfahrenden Autos anpasst. Die Funktionen des ACC-Systems sind verteilt: die Benutzersteuerung und die Verarbeitung der Eingaben (Sensoren, Geschwindigkeit . . .) werden auf verschiedenen Steuerungseinheiten ausgeführt, die über FlexRay verbunden sind.

Das ACC-System wurde im Entwurf mit UML-Klassen-, Objekt- und Zustandsdiagrammen in Rhapsody modelliert. Durch eine halbautomatische Abstraktion wurde ein Zeitanalyse-Modell generiert. Informationen über Abhängigkeiten von Tasks wurden automatisch aus dem Entwurfsmodell (u.a. aus Zustandsdiagrammen) entnommen und transformiert. Vorhandene Objektdiagramme aus dem Entwurf wurden aufbereitet und Elemente des SPT-Profiles wurden hinzugefügt. Für die Analyse fehlende Informationen wurden manuell ergänzt. Das entstandene Zeitanalyse-Modell des ACC-Systems wurde vollautomatisch

in die Formate der Analysewerkzeuge TIMES und SymTA/S transformiert und analysiert. Durch die Analyse wurde die Auslastung der Ressourcen, maximale Antwortzeiten von Tasks und die Latenzzeit zwischen Benutzereingaben und deren Verarbeitung ermittelt.

5 Zusammenfassung und Ausblick

Wir haben ein Zeitanalyse-Modell vorgestellt, das als Zwischenschritt zwischen Entwurfsmodellen und verschiedenen Zeitanalysewerkzeugen eingesetzt wird, um das Echtzeit- und Performanzverhalten verschiedener Architekturvarianten zu spezifizieren, zu analysieren und zu bewerten. Das Zeitanalyse-Modell basiert auf Elementen der UML und des SPT-Profiles. Entwurfsmodelle können - unabhängig auf welcher Modellierungsnotation sie basieren - halbautomatisch abstrahiert und dann manuell mit Zeitaspekten annotiert werden. Eine Zeitanalyse dieses Modells wird durch eine automatische Transformation zu Analysewerkzeugen ermöglicht. Das Zeitanalyse-Modell trennt einerseits klar zwischen den Belangen von Entwurf und werkzeuggestützter Validierung, andererseits wird ein Weg zur effizienten Anbindung der Validierung an den Entwurf aufgezeigt.

In zukünftigen Arbeiten kann das SPT-Profil durch das MARTE-Profil [OMG07] ersetzt werden, in dem genauere Spezifikationen des Echtzeit- und Performanzverhaltens möglich werden. Außerdem kann das Zeitanalyse-Modell in Form von Sichten dem Entwurfsmodell hinzugefügt werden, um die Trennung der verschiedenen Aspekte zu gewährleisten.

Literatur

- [FY04] Elena Fersman und Wang Yi. A generic approach to schedulability analysis of real-time tasks. *Nordic Journal of Computing*, 11(2):129–147, 2004.
- [HACT04] Hans Hansson, Mikael Akerholm, Ivica Crnkovic und Martin Torngren. SaveCCM - A Component Model for Safety-Critical Real-Time Systems. In *Proceedings of the 30th EUROMICRO Conference*, Washington, DC, USA, 2004. IEEE Computer Society.
- [Hag05] Matthias Hagner. *Realisierung einer Modelltransformation zur Schedulability-Analyse von UML-Modellen mit dem TIMES-Tool*. Studienarbeit TU Braunschweig, 2005.
- [HHJ⁺05] Rafik Henia, Arne Hamann, Marek Jersak, Razvan Racu, Kai Richter und Rolf Ernst. System Level Performance Analysis - the SymTA/S Approach, 2005.
- [HMPY04] John Haakansson, Leonid Mokrushin, Paul Pettersson und Wang Yi. An Analysis Tool for UML Models with SPT Annotations. In *SVERTS workshop of UML*, 2004.
- [HV06] M. Hendriks und M. Verhoef. Timed Automata Based Analysis of Embedded System Architectures. In *Workshop on Parallel and Distributed Real-Time Systems 2006*, 2006.
- [OMG02] OMG. UML Profile for Schedulability, Performance and Time, 2002.
- [OMG03] OMG. Unified Modeling Language Specification, 2003.
- [OMG07] OMG. A UML Profile for MARTE, 2007.
- [SAE04] SAE. *Architecture Analysis & Design Language (Aadl)*. 2004.