Antrag auf Forschungsstipendium der DFG

Thema: „Methoden zur Modellbasierten Programmierung Autonomer Systeme“

Dipl.-Inform. Dr. Martin Sachenbacher
Institut für Informatik der Technischen Universität München
Orleansstraße 34, D-81667 München
E-Mail: sachenba@in.tum.de

Zusammenfassung

Modellbasierte Programmierung unterstützt die Entwicklung von Software für eingebettete und reaktive Systeme durch eine zusätzliche, wissenbasierte Schicht, welche auf Basis eines Systemmodells und der verfügbaren Beobachtungen den aktuellen Systemzustand einschätzt (Diagnosekomponente) bzw. einen gewünschten Zielzustand herstellt (Planungskomponente). Dies erlaubt einen Schritt hin zu höhersprachlichem Programmcode, was die Entwicklung von Software und deren Anpassung an Systemvarianten - z.B. in der Anlagen- und Automobiltechnik, in der Luft- und Raumfahrt - wesentlich vereinfacht. Für zeit- und speicherplatzkritische Anwendungen, wie sie vor allem im Bereich autonomer Systeme auftreten, ist dabei die Adäquatheit des verwendeten Modells sowie die Effizienz der eingesetzten Inferenzalgorithmen von entscheidender Bedeutung. Hierzu sollen an der TU München entwickelte Methoden der automatischen qualitativen Modellabstraktion und der strukturellen Dekomposition von constraintbasierten Verhaltensmodellen im Rahmen von modellbasierter Programmierung eingesetzt und weiter verfeinert werden. Das Forschungsvorhaben soll in Kooperation mit dem Artificial Intelligence Laboratory/Space Systems Laboratory am Massachusetts Institute of Technology (Cambridge, USA) durchgeführt werden.

Schlagwörter: Wissenbasierte Systeme, Softwareentwurfsmethodik, Modellbasiertes und Qualitatives Schließen.

Motivation und Darstellung des Erkenntnisstands

In zahlreichen Anwendungsgebieten, z.B. in der Maschinen- und Anlagentechnik, im Automobilbereich sowie in der Luft- und Raumfahrt, treten sogenannte eingebettete bzw. reaktive Softwaresysteme auf. Aufgabe dieser Programme ist es, anhand der Beobachtungen, die für ein physikalisches System verfügbar sind (Sensorsignale), geeignete Steuerbefehle (Aktuatorsignale) zu bestimmen, um so ein vorgegebenes Ziel zu erreichen bzw. dessen Einhaltung zu überwachen (Abbildung 1, linke Seite).

Modellbasierte Programmierung erweitert eingebettete und reaktive Softwaresysteme dahingehend, daß nicht mehr nur die unmittelbar beobachtbaren Eingabegrößen bzw. die unmittelbar steuerbaren Ausgabegrößen, sondern allgemeiner tatsächliche bzw. gewünschte Restriktionen des Systemzustands betrachtet werden. Die „Übersetzung“ eines gewünschten Systemzustands in die Steuersignale, die zu seiner Erreichung nötig sind bzw. die Bestimmung eines nicht unmittelbar beobachtbaren Systemzustands aus dem Zusammenhang der verfügbaren Sensorsignale ist die Aufgabe einer zusätzlichen, zwischengeschalteten Schicht (Abbildung 1, rechte Seite). Diese besteht aus einem wissenbasierten System, welches auf einem Verhaltensmodell des physikalischen Systems arbeitet. Es versucht laufend, aufgrund der Beobachtungen den aktuellen Systemzustand zu erschließen bzw. einzuschätzen (Diagnose) bzw. das System in einen gewünschten Zielzustand zu überführen (Planung).

Modellbasierte Programmierung erlaubt höhersprachlichen Programmcode als traditionelle eingebettete Software, die Wissen über das Verhalten des spezifischen physikalischen Systems nur in impliziter Form enthält und daher meist relativ schwer lesbar und wartbar ist. Durch die Auftrennung in ein gerätespezifisches Verhaltensmodell, aufgabenspezifischen Programmcode und einen Satz generischer Inferenzalgorithmen wird die Wiederverwendung

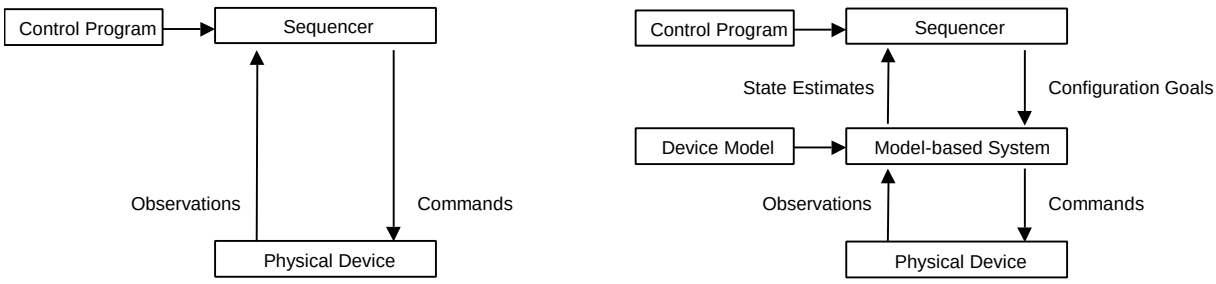

Abbildung 1: Das Grundschema der Programmsteuerung von Geräten (linke Seite) wird bei modellbasierter Programmierung (rechte Seite) durch eine wissensbasierte Schicht ergänzt („Sequencer“ bezeichnet eine Einheit, die den Programmcode schrittweise abarbeitet).

für Systemvarianten erleichtert und so der Entwicklungsprozess eingebetteter Software vereinfacht. Schließlich sind auf modellbasierter Programmierung beruhende Systeme robuster, da die wissensbasierte Schicht mit Hilfe des Systemmodells „intelligent“ auf unvorhergesehene (d.h. zur Entwicklungszeit nicht antizipierte) Situationen, wie z.B. Komponentenfehler, reagieren kann. Letzteres ist besonders wichtig für Anwendungen, bei denen das System für einen gewissen Zeitraum unabhängig von äußeren Eingriffen arbeiten soll. Dies ist u.a. der Fall bei Raumfahrtmissionen, wo aufgrund der mit zunehmender Entfernung von der Erde auftretenden Verzögerungen beim Senden und Empfangen von Signalen bzw. Funkschatten größtmögliche Autonomie gefordert ist (z.B. Mars-Rover). Ein solches Anwendungsfeld stellt außerdem hohe Anforderungen an die Diagnose- und (Re-)Konfigurierungsfähigkeiten des Systems, weil die Wartung einer einmal gestarteten Mission schwierig bis unmöglich ist, und aufgrund der extremen Umgebung (UV-Strahlung, thermische Belastung, etc.) die Ausfallwahrscheinlichkeit einzelner Komponenten relativ hoch ist. Modellbasierte Programmierung erleichtert bzw. ermöglicht die Entwicklung solch komplexer autonomer Systeme.
Wesentliche Grundlagen für modellbasierte Programmiertechniken sind am Xerox Palo Alto Research Center (vgl. [Fromherz et al. 99]) und NASA Ames Research Center ([Williams and Nayak 96], [Muscettola et al. 98]) gelegt worden. In der Forschungsgruppe um Prof. Brian Williams am Massachusetts Institute of Technology wurden in den letzten Jahren Komponenten für ein prototypisches System zur modellbasierten Programmierung entwickelt ([Williams et al. 01], [Kim et al. 01], [Ingham et al. 01]). Inzwischen ist eine Phase erreicht, in der die einzelnen Komponenten (Diagnose- und Planungsalgorithmen, Modellierungssystem und modellbasierte Programmiersprache) zusammengeführt und anhand durchgängiger Beispiele erprobt werden können.

Aufgabenstellung und eigene Vorarbeiten

Für die Machbarkeit modellbasierter Programmierung sind - besonders im Kontext von zeit- und speicherplatzkritischen Anwendungen - zwei Aspekte von entscheidender Bedeutung. Zum einen werden geeignete Verhaltensmodelle benötigt, welche die „richtigen“ (d.h. alle notwendigen, aber möglichst keine unnötigen) Aspekte des Systemverhaltens abbilden. Zum anderen müssen geeignete Inferenzalgorithmen zur Verfügung stehen, die hinreichend effizient sind, um die erforderlichen Reaktionszeiten zu gewährleisten. Ein kritischer Punkt ist insbesondere die Prüfung eines Systemzustands auf Konsistenz mit gegebenen Beobachtungen oder Zielen, da dies eine zentrale Rolle innerhalb der wissenbasierten Schicht spielt.

Im Rahmen meiner Promotion an der Technischen Universität München ([Sachenbacher 01]) habe ich modellbasierte Techniken entwickelt, die in diesem Zusammenhang relevant sind und an denen die Forschungsgruppe am Massachusetts Institute of Technology Interesse zeigt. Diese werden im folgenden kurz erläutert.

Automatische Qualitative Abstraktion

Ein Ansatz, um modellbasiertes Schließen effizient und handhabbar zu machen, liegt darin, nur die für ein Problem wesentlichen, „qualitativen" Verhaltensaspekte zu berücksichtigen. Das Ziel ist dabei, ein Verhaltensmodell (zusammengesetzt aus einer Bibliothek von Modellfragmenten) zu finden, das passend für den Zweck ist, d.h. so grob wie möglich, aber immer noch fein genug für eine gegebene Diagnose- oder Planungsaufgabe. In meiner Dissertation habe ich mich genauer mit einer Problemstellung befaßt, die ich als aufgabenabhängige qualitative Wertebereichsabstraktion (task-dependent qualitative domain abstraction) bezeichne. Aufgabenabhängigkeit wird hierbei charakterisiert durch die Granularität möglicher Eingaben (z.B. die Genauigkeit der verfügbaren Beobachtungen) bzw. die Granularität der gewünschten Ergebnisse (z.B. die Unterscheidung zwischen Normal- und Fehlverhalten für die Diagnose). Das Problem kann dann formalisiert werden als Suche nach Unterscheidungen in den Wertebereichen von Modellvariablen (sogenannte qualitative Werte), die notwendig und hinreichend zur Lösung einer gegebenen Aufgabe sind. In meiner Dissertation werden Algorithmen vorgestellt, mit denen automatisch qualitative Werte für Modellvariablen bestimmt werden können ([Sachenbacher and Struss 01]).

Der Wert dieser Forschungsrichtung für die modellbasierte Programmierung autonomer Systeme liegt darin, „maßgeschneiderte" Modelle zu erzeugen, welche die strikten Laufzeit- und Speicherplatzanforderungen im Bereich von reaktiven und eingebetteten Anwendungen erfüllen können. Qualitative Abstraktion von Wertebereichen ist in diesem Zusammenhang gleich auf zweierlei Art und Weise nützlich: sie reduziert die Größe des Modells, aber auch die Anzahl der Beobachtungen, da in der wissensbasierten Schicht nur die qualitativ unterschiedlichen Bebachtungen berücksichtigt werden müssen. Die Methode erleichtert zudem die Wiederverwendung von Verhaltensmodellen für unterschiedliche Anforderungen und unterstützt dadurch den Aufbau von Modellbibliotheken.

Strukturelle Dekomposition von Verhaltensmodellen

Ein zweiter Ansatzpunkt betrifft die Entwicklung effizienter algorithmischer Verfahren, um realistische Diagnose- und Planungsprobleme hinreichend schnell lösen zu können. Im Kon-

text von modellbasierten Systemen können beide Problemstellungen auf sogenannte Constraint Satisfaction Probleme (CSPs) zurückgeführt werden. Im Rahmen meiner Promotion habe ich mich mit einer Technik befaßt, die es erlaubt, systematisch strukturelle Besonderheiten von CSPs ausnutzen, welche Verhaltensmodellen technischer Geräte entsprechen. Strukturelle Dekomposition (vgl. [Gottlob et al. 00]) basiert darauf, daß Probleme mit azyklischer (d.h. Baum-)Struktur effizient lösbar sind, während schwierige Probleme meist durch eine hohe Zahl auftretender Schleifen charakterisiert sind. Das grundlegende Prinzip ist daher, CSPs in eine Baumstruktur zu überführen, wobei der hierzu nötige Aufwand von der sogenannten Breite (width) des CSP als Maß für dessen Zyklizität abhängt. Das resulierende, zyklenfreie CSP kann dann effizient und vollständig mittels bekannter Constraintfilter-Techniken gelöst werden. In meiner Dissertation habe ich eine heuristische Dekompositionsstrategie in Kombination mit binären Entscheidungsbäumen (sogenannten OBDDs) entwikkelt, um Verhaltensmodelle zu verarbeiten, die aus mehreren hundert Variablen bestehen.
Der Nutzen dieser Forschungsrichtung für modellbasierte Programmierung liegt in der Möglichkeit, realistische Diagnose- und Planungsprobleme mit sehr großen Zustandsräumen anzugehen. Strukturelle Dekomposition kann als Vorverarbeitungsschritt verstanden werden, da es in einer Phase durchgeführt werden kann, in der noch keine Beobachtungen verfügbar sind (d.h. „offline“). Die Möglichkeit, Modelle „vorzukompilieren“, ist besonders für Systeme mit Echtzeitanforderungen hilfreich. Basierend auf der Dualität von Constraint Satisfaction und logischem Schließen legt strukturelle Dekomposition außerdem eine interessante Grundlage für Inferenzverfahren, die garantierte Vollständigkeits- und Korrektheitseigenschaften aufweisen, was für sicherheitskritische Anwendungen sehr nützlich sein könnte.

Die beschriebenen Verfahren wurden bisher überwiegend im Kontext von Automobilanwendungen und modellbasierter Diagnose bzw. Überwachung entwickelt und eingesetzt. Während meiner bisherigen Tätigkeit an der TU München bzw. bei der Robert Bosch GmbH (Stuttgart) geschah dies hauptsächlich im Rahmen des Brite-Euram-Projekts VMBD (Vehicle Model-based Diagnosis) und des BMBF-Verbundprojekts INDIA (Intelligente Diagnose in der Anwendung). Im Projekt VMBD wurden die Methoden in einem Prototyp für die modellbasierte Onboard-Diagnose auf einem Volvo-Versuchsfahrzeug erfolgreich angewendet ([Sachenbacher et al. 00a], [Sachenbacher et al. 00b]) und so deren prinzipielle Eignung für Onboard-Anwendungen demonstriert.

Arbeitsprogramm und vorgesehene Untersuchungsmethoden

In dem beantragten Vorhaben sollen die beiden im vorigen Abschnitt beschriebenen Ansätze (automatische qualitative Abstraktion, strukturelle Dekomposition von Verhaltensmodellen) im Kontext modellbasierter Programmierung eingesetzt und weiterentwickelt werden. Dazu sollen die Verfahren in das am MIT existierende Rahmensystem integriert und auf Szenarien aus dem Bereich der Steuerung und Überwachung autonomer Systeme angewendet werden. Im einzelnen sind dabei folgende Aufgaben zu lösen:

- Die Technik der automatischen qualitativen Abstraktion betrifft vor allem den Kasten „Device Model“ in Abbildung 1. Die Aufgabe ist dabei, den Detaillierungsgrad von Verhaltensmodellen für die spezifische Anwendung in eingebetteten Systemen zu optimieren. Dabei soll die Möglichkeit untersucht werden, Aufgabenbeschreibungen direkt aus modellbasiertem Programmcode zu extrahieren. Ferner ist auch die Weiterentwicklung des Verfahrens für die Abstraktion reellwertiger (d.h. kontinuierlicher) Modelle vorgesehen; in [Sachenbacher 01] wurde dies bereits prinzipiell anhand kleiner Beispiele demonstriert. Automatische Diskretisierung von Modellen würde es ermöglichen, eine Brücke zwischen zwei getrennten Lösungsverfahren für kontinuierliche und diskrete Modelle zu schlagen, auf die das MIT-System bislang zurückgreift.
- Die Technik der strukturellen Dekomposition von Verhaltensmodellen betrifft vor allem den Kasten „Model-based System“ in Abbildung 1. Die Aufgabe besteht hier in der Integration mit einem an der Forschungsgruppe um Prof. Williams entwickelten Algorithmus zur effizienten Lösung von Constraint-Optimierungsproblemen (siehe [Williams and Ragno 02]), der die Grundlage der Diagnose- und der Planungskomponente des modellbasierten Programmsystems bildet. Das Ziel ist dabei, die Erkennung von Inkonsistenzen des Modells mit Beobachtungen oder Zielvorgaben und die Generierung sogenannter Konfliktmengen aus diesen Inkonsistenzen zu verbessern. Bisher geschieht dies mit relativ simplen Verfahren, die für größere Beispiele jedoch rasch ineffizient werden. Hier soll das Verbesserungspotential durch strukturelle Dekompositionsmethoden untersucht werden.

Als Anwendungsbeispiele stehen voraussichtlich (vereinfachte) Szenarien der NASA und des JPL (Jet Propulsion Laboratory) zur Steuerung und Überwachung autonomer Systeme (kooperierende Mars-Rover, Deep Space One-Mission, Regelung der Biosphäre eines Mars-Habitats) zur Verfügung.

Bedeutung des Forschungsvorhabens für die weiteren wissenschaftlichen und beruflichen Pläne

Das beantragte Forschungsstipendium dient vor allem dem Ziel, die im Rahmen der Promotion entwickelten Methoden und Kenntnisse in Form von praktischer wissenschaftlicher Tätigkeit zu vertiefen. Gegenüber der Promotion wird dabei eine Verbreiterung und Verallgemeinerung sowohl im Hinblick auf die Anwendungsdomäne als auch hinsichtlich des zu unterstützenden Aufgabenspektrums angestrebt. Die Eingliederung der Methoden in den Kontext modellbasierter Programmierung in der Arbeitsgruppe um Prof. Williams eröffnet in dieser Hinsicht besondere Möglichkeiten.

Erstens bietet die angestrebte Tätigkeit durch die Nähe zum Space Systems Laboratory (SSL) Gelegenheit, den Horizont der Anwendungen in Richtung Luft- und Raumfahrt zu erweitern. Im Vergleich zu den bisher von mir behandelten Automobilsystemen sind diese gekennzeichnet durch eine höhere Anzahl an verfügbaren Sensorsignalen, größere Redundanz, aber auch höhere Ausfallwahrscheinlichkeit der Komponenten. Gegenüber der Automobildomäne erscheint die Unterstützung durch wissenbasierte Systeme deshalb dringlicher, zugleich aber auch anspruchsvoller. Ein Ziel des geplanten Aufenthalts ist, die entwickelten Methoden auf diesen neuen Anwendungskontext zu übertragen und entsprechend fortzuentwickeln.

Zweitens spielen gegenüber meinem bisherigen Arbeitskreis, der sich hauptsächlich auf die modellbasierte Unterstützung von Diagnose und Verhaltensvorhersage beschränkte, bei modellbasierter Programmierung zusätzlich Planungs- und (Re-)konfigurationsaufgaben eine wichtige Rolle. Dies gilt besonders für Anwendungen im Kontext autonomer Systeme. Die avisierte Forschungsgruppe von Prof. Williams hat entsprechende Stärken und Fachkenntnisse im Bereich der (temporalen) Planung, Steuerung und modellbasierten Rekonfiguration aufzuweisen. Ich möchte den Aufenthalt auch nutzen, um meine Kenntnisse und Fähigkeiten in diesem Bereich zu erweitern.

Im Anschluß an den geplanten Postdoc-Aufenthalt strebe ich die Rückkehr an das Institut für Informatik der Technischen Universität München an. Möglichkeiten, die gesammelten Erfahrungen in den Forschungs- und Lehrbetrieb einzubringen, bieten sich insbesondere an den Arbeitsgruppen von Prof. Broy (Softwareentwurfsmethodik), Prof. Radig (autonome, reaktive Systeme) und Prof. Struss (modellbasierte Systeme).

Dauer und Zeitraum des Forschungsvorhabens

Das Forschungsstipendium wird für die Dauer von einem Jahr erbeten. Für den beabsichtigten Auslandsaufenthalt wird der Zeitraum von Oktober 2002 bis September 2003 angestrebt.

Überlegungen zur Wahl des Arbeitskreises

Prof. Brian Williams ist ein herausragender und weltweit renommierter Forscher im Bereich des modellbasierten und qualitativen Schließens. In der achtziger Jahren leistete er am Xerox Palo Alto Research Center mit der „General Diagnostic Engine“ ([de Kleer and Williams 87]) Pionierarbeit auf dem Gebiet der modellbasierten Diagnose und der automatischen Generierung von Messvorschlägen. In den neunziger Jahren gehörte er dem NASA Ames Research Center an, wo er mit der Entwicklung des modellbasierten, autonomen Systems „Remote Agent“ ([Muscettola et al. 98]) befaßt war. Dieses wurde im Rahmen der NASA-Mission „Deep Space One“ erfolgreich engesetzt und gewann im Jahr 1999 den NASA Space Act Award. Für seine Publikationen erhielt Prof. Williams mehrere „Best Paper“-Auszeichnungen. Prof. Williams lehrt und forscht heute am Massachusetts Institute of Technology und gehört dort sowohl dem Artificial Intelligence Laboratory (AIL) als auch dem Space Systems Laboratory (SSL) an.

Der Prototyp einer modellbasierten Programmierumgebung, der dort in den letzten Jahren unter seiner Leitung entstanden ist, stellt im wesentlichen eine Weiterentwicklung des „Remote Agent“-Systems der NASA dar. Die Existenz dieses Software-Rahmens und der entsprechenden Infrastruktur bringt für das geplante Forschungsvorhaben den Vorteil mit sich, daß direkt der Kern der beschriebenen Forschungsaspekte angegangen werden kannn. Die Nähe zur Anwendungsdomäne (Space Systems Laboratory) hat weiterhin den Vorteil, daß realistische Anwendungsszenarien (voraussichtlich einige Beispiele aus NASA- bzw. JPL-Missionen) zur Verfügung stehen. Innerhalb Deutschlands ist mir zur Zeit keine Forschungsgruppe bekannt, bei der eine vergleichbar fortgeschrittene Ausgangssituation vorhanden wäre. Zu der Forschungsgruppe von Prof. Williams bestehen seit längerer Zeit gute Kontakte. Ich selbst war dort Ende letzten Jahres für einige Tage zu Besuch, wobei die dargelegten Perspektiven für eine mögliche Zusammenarbeit besprochen wurden.

Zuwendungen von anderer Seite

Keine.

Literatur

[de Kleer and Williams 87] Johan de Kleer, Brian C. Williams: Diagnosing multiple faults. Artificial Intelligence, 32(1):97-130, 1987.

[Fromherz et al. 99] Markus P.J. Fromherz, Vijay A. Saraswat, and Daniel G. Bobrow: Model-based Computing: Developing Flexible Machine Control Software. Artificial Intelligence, 114(1-2): 157-202, 1999.

[Gottlob et al. 00] Georg Gottlob, Nicola Leone, Francesco Scarcello: A comparison of structural CSP decomposition methods. Artificial Intelligence, 124(2):243-282, 2000.

[Ingham et al. 01] Michel Ingham, Roberto Ragno, Brian C. Williams, A reactive model-based programming language for robotic space explorers. Proceedings of ISAIRAS-01, Montreal, 2001.

[Kim et al. 01] Phil Kim, Brian C. Williams, Mark Abramson: Executing reactive, model-based programs through graph-based temporal planning. Proceedings of IJCAI-01, Seattle, 2001.

[Muscettola et al. 98] Nicola Muscettola, P. Pandurang Nayak, Barney Pell and Brian C. Williams: Remote Agent: To Boldly Go Where No AI System Has Gone Before. Artificial Intelligence 103(1-2):5-48, 1998.

[Sachenbacher 01] Martin Sachenbacher: Automated Qualitative Abstraction and its Application to Automotive Systems. Dissertation, Institut für Informatik, Technische Universität München, 2001.

[Sachenbacher and Struss 01] Martin Sachenbacher and Peter Struss: AQUA: A Framework for Automated Qualitative Abstraction. Working Papers of the 15th International Workshop on Qualitative Reasoning (QR-01), San Antonio, USA, 2001.

[Sachenbacher et al. 00a] Martin Sachenbacher, Peter Struss, Reinhard Weber: Advances in the design and implementation of on-board diagnosis functions for diesel injection systems based on a qualitative approach to diagnosis. Society of Automotive Engineers World Congress and Exhibition, Detroit, 2000.

[Sachenbacher et al. 00b] Martin Sachenbacher, Peter Struss, and Claes M. Carlén: A prototype for model-based on-board diagnosis of automotive systems. AI Communications, 13(2):83-97, 2000.

[Williams and Nayak 96] Brian C. Williams and P. Pandurang Nayak: A Model-based Approach to Reactive Self-Configuring Systems. Proceedings of AAAI-96, 1996.

[Williams and Ragno 02] Brian C. Williams, Robert J. Ragno: Conflict-directed A* and its role in model-based embedded systems. Journal of Discrete Applied Mathematics, to appear.

[Williams et al. 01] Brian C. Williams, Seung Chung, Vineet Gupta: Mode estimation of model-based programs: Monitoring systems with complex behavior. Proceedings of IJCAI-01, Seattle, 2001.