

Abschlussbericht des Projektes PE 748/1-3
**HYBRIS – Effiziente Analyse von hybriden
Systemen**
im Rahmen des Schwerpunktprogramms SPP 1064 – Integration
von Techniken der Softwarespezifikation für
ingenieurwissenschaftliche Anwendungen
Bremen, 31. März 2006

Inhaltsverzeichnis

1	Allgemeine Angaben	2
2	Zusammenfassung	3
3	Arbeits- und Ergebnisbericht	6
3.1	Ausgangslage	6
3.2	Beschreibung der durchgeführten Arbeiten	8
3.3	Darstellung der erzielten Ergebnisse	11
3.4	Ausblick und zukünftige Arbeiten	16
3.5	Interdisziplinäre Weiterentwicklung	18
3.6	Anwendung	19
3.7	Beteiligte Wissenschaftler	21
3.8	Weiterqualifikation	23
4	Publikationen	25
4.1	Publikationen in Fachzeitschriften	25
4.2	Kongressbeiträge	25
4.3	Buchbeiträge	28
4.4	Studien- und Diplomarbeiten, Dissertation, Habilitationen, Be- richte, sonstige Publikationen	28

1 Allgemeine Angaben

DFG-Geschäftszeichen:	PE 748/1-3
Antragsteller:	Prof. Dr. habil. Jan Peleska
Institut/Lehrstuhl	Universität Bremen Fachbereich 3 – Mathematik und Informatik Arbeitsgruppe Betriebssysteme/verteilte Systeme
	Postfach 330 440 D-28334 Bremen
Telefon:	0421 / 218-7092
Fax:	0421 / 218-3054
e-mail:	jp@informatik.uni-bremen.de
Aus DFG-Mitteln bezahlte wiss. Mitarbeiter	Stefan Bisanz (Bat Ila) – gesamter Projektzeitraum
Thema des Projektes	Effiziente Analyse von hybriden Systemen
Förderzeitraum insgesamt	01.04.1999 — 31.03.2005
Fachgebiet, Arbeitsrichtung	Informatik — theoretische Informatik, Spezifikation, Verifikation, Simulation und Test
Anwendungsfelder	Eingebettete sicherheitsrelevante Steuerungssysteme für Bahn und Luftfahrt, allgemeiner für Transportsysteme
Am Projekt beteiligte Kooperationspartner	Kooperationspartner im Rahmen des SPP 1064: Prof. Dr. Schnieder, TU Braunschweig Kooperationspartner an der Universität Bremen: Prof. Dr. Rolf Drechsler internationale Kooperationspartner: Prof. Dr. Anne Haxthausen, Denmark Technical University DTU, Lyngby (Kopenhagen)
Am Projekt beteiligte weitere wissenschaftliche Mitarbeiter (ohne DFG-Förderung) aus der Arbeitsgruppe Betriebssysteme/verteilte Systeme	Peter Amthor, Kirsten Berkenkötter, Jan Brederke, Bettina Buth, Markus Dahlweid, Ulrich Hannemann, Detlef Kendelbacher, Oliver Meyer, Holger Schlingloff, Uwe Schulze, Elena Svetogorova

2 Zusammenfassung

Im Projekt **HYBRIS – Effiziente Analyse von hybriden Systemen** wurden im Rahmen der Referenzfallstudie zur Verkehrsleittechnik des Schwerpunktprogramms SPP 1064¹ Beschreibungsmittel aus der Informatik und aus ingenieurtechnischen Disziplinen für **Hybride Systeme**² integriert und in Bezug auf ihre effiziente Verwendbarkeit bei Verifikation, Simulation und Test analysiert. Dabei standen folgende wissenschaftliche Fragestellungen im Vordergrund: (1) Welche **Beschreibungsformalismen** für Hybride Systeme sind für den Einsatz im ingenieurtechnischen Umfeld praxisgeeignet und gleichzeitig ausreichend formal, um mathematisch fundierte Analysen in Bezug auf Korrektheit, Sicherheitseigenschaften etc. zu ermöglichen? (2) Mit welchen Methoden lässt sich eine hybride Spezifikation **automatisiert** – d. h. ohne manuelle Verfeinerungs- und Codierschritte – in Code transformieren, der sich auf eingebetteten Systemplattformen effizient ausführen lässt? (3) Welche **Verifikations- und Testverfahren** sind geeignet, Spezifikationen, sowie die danach implementierten eingebetteten HW/SW-Systeme in Bezug auf ihre Korrektheit zu untersuchen? (4) Welche spezialisierten Ergebnisse lassen sich mit den erarbeiteten Methoden für die **Anwendungsdomäne Bahnsteuerungssysteme** erzielen? (5) Mit welchen **Werkzeugen** lassen sich die entwickelten Methoden und Verfahren effizient im ingenieurtechnischen Umfeld anwenden?

Für die Spezifikation Hybrider Systeme wurde der Beschreibungsformalismus **HybridUML** entwickelt. Dieser unterscheidet sich von bisherigen vergleichbaren Spezifikationsansätzen dadurch, dass HybridUML keine eigenständige formale Sprache ist, sondern als **Profil** in die **Unified Modeling Language UML 2.0** integriert wurde. Dieser Ansatz ermöglicht es, auf die bestehenden Sprachelemente der UML zurückzugreifen und vorhandene UML-CASE Tools weiter zu verwenden.

Die Semantik einer HybridUML-Spezifikation wird durch Übersetzung in eine formale Zielsprache niedrigerer Abstraktionsstufe, die **Hybrid Low-Level Language HL3**, definiert³. Das im Zuge dieser Transformation entste-

¹SPP 1064: Integration von Techniken der Softwarespezifikation für ingenieurwissenschaftliche Anwendungen.

²Auch **diskret-kontinuierliche Systeme** genannt: Es handelt sich hierbei um Systeme, in denen sowohl diskrete als auch zeitlich kontinuierliche Abläufe beobachtet, verarbeitet und gesteuert werden.

³Wir sprechen im folgenden von **High-Level** und **Low-Level Formalismen**. Erstere sind durch höhere Anstraktion und geeignete – teilweise grafische – Syntax für den interaktiven Systementwurf geeignet. Letztere sind ebenfalls formal, verwenden aber eine “maschinen-nähere” Syntax, wodurch sie für die Erzeugung ausführbaren Codes geeignet sind, aber in

hende verteilte HL3-Programm ist nicht nur in seinem Verhalten über eine operationelle Semantik mathematisch wohldefiniert, sondern auch in harter Echtzeit ausführbar. Damit haben wir das Problem der modellbasierten Entwicklung für eine große Klasse Hybrider Systeme vollständig gelöst.

Auf dem Gebiet der **Testmethoden für Hybride Systeme** wurden auf zwei Gebieten neue Ergebnisse erzielt: (1) Im Bereich der automatisierten Testfallgenerierung wurde gezeigt, wie durch Zuhilfenahme einer internen Modellrepräsentation und einer geschichteten Architektur für die zur Testfallgenerierung beitragenden Softwarekomponenten mächtige Automatisierungssysteme entstehen, die Teillösungen für die systematische Modelltraversion, die symbolische Spezifikationsausführung, SAT-Solver und Verfahren des Constraint Programming auf effiziente Weise kombinieren. (2) Im Bereich automatisierte Auswertung⁴ wurde gezeigt, wie statistische Verfahren eingesetzt werden können, um die Vertrauenswürdigkeit beobachteter Analogabläufe zu bewerten.

Die Spezialisierung der erzielten Ergebnisse auf **Bahnsteuerungssysteme** führte zu einem domänenspezifischen **Railway Control System Domain (RCSD)** Profil für die UML 2.0. Es wurde gezeigt, wie solche Modelle mit Hilfe einer Low-Level Repräsentation in SystemC durch induktive Beweismethoden in Kombination mit **Bounded Model Checking (BMC)** verifiziert werden können. Konventionelle Modellprüfungsverfahren galten bisher auf Grund der nicht beherrschbaren Modellgrößen als für den Bahnbereich ungeeignet. Weiterhin lässt sich aus dem Spezifikationsmodell automatisch Objektcode generieren, der durch Äquivalenzchecks auf SystemC-Ebene auf Korrektheit überprüfbar wird. Damit ist die für sicherheitsrelevante Bahnsteuerungen besonders relevante Frage der Einsatzmöglichkeit für nicht validierte Compiler positiv beantwortet.

Entwicklungs- und Testverfahren wurden zusammen mit Visualisierungskomponenten in einem UML CASE Tool und ein daran gekoppeltes Testwerkzeug vollständig integriert.

Detaillierte Ergebnisbeschreibungen, die Anwendungen der hier skizzierten Ergebnisse, sowie der Ausblick auf weitergehende Forschungsaktivitäten sind in den Abschnitten [3.3](#), [3.6](#) und [3.4](#) dargestellt.

Generelle Verweise: Das Projekt HYBRIS wurde im Rahmen des SPP 1064 in drei Phasen ((a) 04/1999 — 03/2001, (b) 04/2001 — 03/2003, (c) 04/2003 — 03/2005) abgewickelt. Der Beitrag [[BBHPO4a](#)] gibt eine Zu-

der Regel nicht für manuelle Bearbeitung und Analyse verwendet werden sollten.

⁴Sog. **Testorakel**.

sammenfassung der in der letzten Projektphase erzielten Ergebnisse. Eine umfassende Darstellung der Ergebnisse der Phase (c) ist unter [\[PBB+06\]](#) online verfügbar. Für eine zusammenfassende Darstellung der in den beiden früheren Projektphasen erzielten Ergebnisse verweisen wir auf die zugehörigen der DFG zugegangenen Zwischenberichte. Detaillierte Verweise auf die im Rahmen des Vorhabens entstandene Publikationen enthält Abschnitt [4](#) und das Literaturverzeichnis.

3 Arbeits- und Ergebnisbericht

3.1 Ausgangslage

Zielsetzung des Projektes **HYBRIS – Effiziente Analyse von hybriden Systemen** war die Integration unterschiedlicher Beschreibungsmittel aus Informatik und ingenieurtechnischen Disziplinen, um **Hybride Systeme** zu spezifizieren, zu verifizieren und zu testen. Typische Anwendungsfelder für die zu bearbeitenden Fragestellungen sind **eingebettete Systeme**, mit einem besonderen – durch die Referenzfallstudie des SPP 1064, sowie durch die Forschungsschwerpunkte des Antragstellers motivierten – Fokus auf sicherheitsrelevanten Bahnsteuerungsanwendungen.

Hierzu wurden zunächst verschiedene zu Projektbeginn bzw. im Verlauf des Projektes verfügbar gewordene Beschreibungsformalismen anderer Forschungsgruppen untersucht und bzgl. ihrer Eignung für die Anwendung im ingenieurtechnischen Umfeld geprüft. Dabei stellte sich bald heraus, dass diese in den Forschungsgruppen für Formale Methoden entstandenen Spezifikationsprachen – trotz ihres erheblichen Nutzens für die Erforschung Hybrider Systeme – gerade diesem Eignungskriterium nicht genügten:

Der bekannte *Duration Calculus* [ZRH93] hatte sich bereits vor Projektbeginn in Experimenten mit Kooperationspartnern aus der Industrie auf Grund seiner impliziten Spezifikationstechnik und der von konventionellen CASE-Methoden stark abweichenden Syntax als ungeeignet erwiesen.

Hybrid CSP [Jif94] wurde in der ersten Projektphase im Detail untersucht [Amt99] und schied aus, weil der Formalismus bereits auf höchstem Abstraktionslevel eine Diskretisierung kontinuierlicher Abläufe erzwingt: Kontinuierliche Abläufe können in Hybrid CSP nur auf lokalen Variablen sequenzieller Prozesse modelliert werden, der Datenaustausch im verteilten System vollzieht sich zu diskreten Zeitpunkten über synchrone Kanäle. Damit lassen sich physikalische Modelle, in denen globale reellwertige Variablen als natürliche Beobachtungsgrößen vorkommen, mit Hybrid CSP nicht modellieren.

Henzingers *Hybrid Automata (HA)* [Hen96], die in der 2. Projektphase untersucht wurden, dienten vor allem dem Ziel, die Möglichkeiten der Modellprüfung für Hybride Systeme zu untersuchen. Ein Einsatz der HA im industriellen Umfeld war vom Autor sicher nie beabsichtigt worden und ist auch – vor allem wegen fehlender Möglichkeiten zur Hierarchisierung komplexer Automaten – nicht zu empfehlen.

Die *Hierarchical Hybrid Automata (HHA)* von Alur und anderen [ADE+03] (untersucht in den Phasen 2 und 3) beseitigen den Nachteil der “flachen”

HA. Die von den Autoren vorgeschlagene Syntax, sowie einige semantische Feinheiten unterscheiden den Formalismus nach unserer Einschätzung allerdings so weit von den gängigen CASE-Methoden, dass ein weit verbreiteter Einsatz im ingenieurtechnischen Umfeld für die nächste Zeit als sehr unwahrscheinlich erscheint.

Die im Rahmen des SPP mit anderen Forschern durchgeführten intensiven Diskussionen über die Stärken und Schwächen der *Unified Modeling Language UML*, sowie die in UML 2.0 eingebrachten methodischen Verbesserungen, führten auf Grund dieser Untersuchungsergebnisse dazu, einen dem Eignungskriterium genügenden Beschreibungsformalismus für Hybride Systeme nicht etwa vollständig neu zu entwerfen, sondern als systematische Spracherweiterung in die UML 2.0 zu integrieren. Diese Vorgehensweise wurde – trotz unserer ursprünglichen Kritik an der UML – weiter durch die Erkenntnis bestärkt, dass sich auch domänenspezifische Formalismen als UML 2.0-Erweiterung codieren lassen. Im Gegensatz zu anderen (semi-)formalen Sprachen ist syntaktische und semantische Erweiterbarkeit gerade ein Merkmal der UML, so dass wir auch jetzt – nach Abschluss des Projektes HYBRIS – keine für ingenieurtechnische Projekte geeignete Alternative zu unserem Vorgehen sehen.

3.2 Beschreibung der durchgeführten Arbeiten

Für die Projektphasen 1 und 2 verweisen wir auf den Zwischenbericht, der zum Ende der Phase 2 erstellt wurde. Wir gehen hier daher nur auf die für Phase 3 geplanten und durchgeführten Arbeitspakete ein. In Bezug auf die dabei erzielten Ergebnisse verweisen wir auf Abschnitt 3.3.

AP 1 – Verallgemeinerte Semantik für Henzingers Hybride Automaten (HA): Die geplanten Arbeitsschritte wurden im Kontext unseres o. g. HybridUML Profils durchgeführt. Es wurde eine differentialgeometrische Interpretation der Invarianten und Flussbedingungen der HA entwickelt, welche die Bildung von Heuristiken zur Testdatenauswahl unterstützt. Auf Grund von Erwägungen bzgl. der praktischen Nutzbarkeit für die Spezifikation großer Systeme erweiterten wir die ursprünglichen Zielsetzungen dieses Arbeitspakets auf die Einführung von Hierarchisierungsmöglichkeiten für Hybride Automaten. Dieser Schritt nahm die Arbeiten von Alur *et. al.* [ADE⁺03] zum Ausgangspunkt. Im Gegensatz zu [ADE⁺03] wurde dies jedoch nicht durch eine neue Syntax, sondern durch Annotation hierarchischer Statecharts mit Invarianten und Flussbedingungen erreicht. Weiterhin im Gegensatz zu [ADE⁺03] wurde zunächst kein direkt auf Hybriden Statecharts aufsetzendes semantisches Modell entwickelt, sondern – mit der Zielsetzung der effizienten modellbasierten Entwicklung – eine Transformationssemantik in eine semantisch wohldefinierte und gleichzeitig in Echtzeit ausführbare Zielsprache – das Hybrid Low-Level Language Framework HL3 – konstruiert.

AP 2 – Testtheorie – Testdatenerzeugung für Hybride Systeme: Auf der Grundlage von Timed CSP wurden zunächst einige spezielle Algorithmen für Echtzeitsysteme ohne Analogabläufe entwickelt: Diese können nach geeigneter struktureller Dekomposition des Zielsystems in zeitlich diskrete und kontinuierliche Abläufe verwendet werden, um die diskreten Steuerungsübergänge zwischen den Modi (= Kontrollzustände) des Systems zu prüfen. Die Zeitpunkte, an denen solche diskreten Übergänge erfolgen, sind allerdings im Allgemeinen an Bedingungen geknüpft, für deren Eintreten kontinuierliche Evolutionen bestimmter Beobachtungsgrößen verantwortlich sind. Die Komplexität dieser Abhängigkeiten zwischen diskreten und kontinuierlichen Verläufen wurde beherrschbar gemacht, indem eine von der konkreten syntaktischen Repräsentation des High-Level Formalismus unabhängige Internrepräsentation für Hybride Systemmodelle entwickelt wurde. Algorithmen zur Testfallerzeugung operieren auf dem internen Systemmodell und können damit für unterschiedliche Formalismen wieder verwen-

det werden. Weiterhin wurde eine geschichtete Softwarearchitektur für Testfallerzeugungsssoftware entwickelt, welche eine saubere Trennung zwischen den Softwarekomponenten ermöglicht, welche die verschiedenen Teilprobleme der gesamten Aufgabenstellung bearbeiten. Über die ursprüngliche Aufgabenstellung hinaus gehend wurden probabilistische Fragestellungen für die Bewertung von Analogverläufen im Testlingsverhalten untersucht, um die Vertrauenswürdigkeit der Aussagen automatisierter Testorakel statistisch bewerten zu können.

AP 3 – Domänenspezifische Beschreibungssprache für Bahnsteuerungen: Eine domänenspezifische Beschreibungssprache für Bahnsteuerungen wurde entwickelt. Entgegen unserer ursprünglichen Annahme, dass solche Formalismen nicht im Rahmen der UML repräsentierbar sind, konnte – motiviert durch die Zielsetzung, auf einfache Weise Werkzeugunterstützung für solche Formalismen bieten zu können und Standardspezifikationskonstrukte nicht für jede Domäne wieder neu konstruieren zu müssen – gezeigt werden, dass sich diese Beschreibungssprache vollständig als **Railway Control Systems Domain RCSD**-Profil der UML 2.0 ausdrücken lässt.

AP 4 – Grafische Beschreibungstechniken für zeitlich kontinuierliche Abläufe: In der Zusammenarbeit mit den Industriepartnern Airbus und Siemens wurde festgestellt, dass eine universell verwendbare grafische Beschreibungstechnik zur Zeit noch nicht in Sicht ist. Wir haben stattdessen einen “Baukasten” grafischer Unterstützungswerkzeuge integriert, mit Hilfe dessen sich Testspezifikationen in unterschiedlichen *Views* entwickeln und Testabläufe entsprechend visualisieren lassen. Die Views umfassen HybridUML-Diagramme und domänenspezifische Darstellungen, die alle über ein gemeinsames Datenabstraktions- und Kommunikationskonzept vom Testwerkzeug und CASE-Tool aus angesteuert werden können.

AP 5 – Referenzfallstudie Verkehrsleittechnik: Die Fallstudie wurde im HybridUML-Formalismus ausgearbeitet.

AP 6 – Transformation Spezifikation → ausführbares Programm: Zur Lösung dieser Aufgabe wurde der Low-Level Formalismus HL3 so konstruiert, dass seine Spezifikationen direkt in ein ausführbares verteiltes Programm übersetzbar sind. Zuordnung der transformationellen Semantik und Erzeugung eines ausführbaren Programms konnte damit für HybridUML-Spezifikationen in einem Schritt geschehen.

AP 7 – Transformation Spezifikation → **Testsuite:** Die Erzeugung automatisch ablaufender Testsuiten wurde durch die Kombination der Ergebnisse aus AP 2 – hieraus ergaben sich die konkreten Testdaten und Ablauffolgen – und AP 6 – hierdurch wurde die Testprozedurausführung in harter Echtzeit ermöglicht – bewerkstelligt.

AP 8 – Integration in Workbench: Die Arbeitsabläufe zwischen UML-CASE Werkzeug, Codegenerierungskomponenten und Simulations- und Testumgebung wurden durch Kopplung der Werkzeuge *Together* (UML Tool) und *RT-Tester* (Test- und Simulationswerkzeug) integriert. Die Codegenerierungskomponenten wurden im CASE-Tool als *Plug-Ins* realisiert; die HL3 Laufzeitumgebung wurde zur Ausführung von Tests und Simulationen in das Testwerkzeug integriert.

3.3 Darstellung der erzielten Ergebnisse

Beschreibungsformalismen: Nach einer kritischen Analyse bestehender Beschreibungsformalismen für Hybride Systeme wurde **HybridUML** als **Profil der UML 2.0** entworfen: Diese nach dem Framework der Object Management Group ausgearbeitete Spracherweiterung ermöglicht die Spezifikation zeitlich kontinuierlicher Abläufe im Kontext der UML 2.0: Systeme bestehen aus kooperierenden nebenläufigen **Agenten**; das Verhalten der letzteren ist durch hierarchisch strukturierte Kontrollmodi spezifiziert, die in der UML 2.0 durch Statecharts repräsentiert werden. Die Profilerweiterung ermöglicht, die Kontrollmodi durch Invarianten über diskrete und reellwertige Zustandsvariablen, Flussbedingungen über die Evolution der kontinuierlichen Abläufe über der Zeit in Form von Differentialgleichungen, algebraische Bedingungen über den Zusammenhang verschiedener reellwertiger Variablen zu annotieren. Architektur wird durch eine Erweiterung der **UML 2.0 Strukturdiagramme (Composite Structure Diagrams)** innerhalb des Profils beschrieben. Das HybridUML-Profil und seine Interpretation wurde in [BBHP03, BBHP06, BBHP04a, BBHP04b] veröffentlicht. Ein zusammenhängende Darstellung findet man ausserdem in [PBB⁺06]. Gegenüber konkurrierenden Ansätzen zeichnet es sich vor allem durch seine enge Integration mit den unten genannten Codegenerierungs- und Testmethoden aus.

Modell-basierte Codegenerierung aus dem HybridUML-Profil: Durch die Spezifikation der HybridUML Konstrukte als Profil der UML 2.0 erhält der Formalismus eine formale Syntax und statische Semantik, sowie – dem von der OMG vorgeschlagenen Beschreibungsstil folgend – eine informelle Verhaltenssemantik. Für die automatische Umsetzung von Spezifikationen in Code, ihre “mechanische” Verifikation sowie die automatisierte Ableitung von Tests ist allerdings eine vollständige Formalisierung der Semantik unumgänglich. Bei deren Definition entschieden wir uns für eine **Transformationssemantik**: Das Verhalten einer HybridUML-Spezifikation wird hierbei durch Übersetzung in eine Zielsprache niedrigerer Abstraktionsstufe mit bereits bekannter formaler operationeller Semantik festgelegt. Die von uns neu entworfene Zielsprache **Hybrid Low-Level Language HL3** ist direkt in ein verteiltes Programm übersetzbar, welches in harter Echtzeit auf Mehr-CPU Clustersystemen ausgeführt werden kann. Der Transformationsansatz garantiert Konsistenz zwischen Modellsemantik und generiertem Programm; ausserdem lassen sich Spracherweiterungen einfach durch Erweiterung der Transformationsvorschrift einführen. Zur Illustration

des HybridUML-Formalismus wurde die im SPP 1064 verwendete Fallstudie über die Steuerung von Zügen und Bahnübergängen im funkbasierten Fahrbetrieb spezifiziert [BBHP04a, Bis05]. Dort ist auch das HL3-Framework und seine operationelle Semantik im Detail beschrieben.

HL3 konkurriert vor allem mit Henzingers GIOTTO [HHK03], einem der wenigen Low-Level Formalismen, die tatsächlich eine formale Semantik besitzen. Im Gegensatz zu GIOTTO ermöglicht HL3 die kombinierte Verwendung von *Time-Triggered Modes* und *Event-Triggered Modes*; weiterhin wurde das HL3 Framework von vornherein dazu konzipiert, als Zielsprache für unterschiedliche High-Level Formalismen zu dienen.

Testmethoden für Hybride Systeme wurden nach zwei Gesichtspunkten untersucht, die beiden zum Themenfeld **spezifikationsbasiertes Testen** gehören: (1) Verfahren zur automatisierten Testfall- und Testdatengenerierung auf Grundlage hybrider Spezifikationen, und (2) Verfahren zur automatisierten Auswertung der Reaktionen des Testlings gegen die Spezifikation. Aufbauend auf den Ergebnissen zur automatisierten Codegenerierung mittels HL3 wurde eine interne Modellrepräsentation entwickelt, in die unterschiedliche High-level Formalismen abgebildet werden können. Algorithmen zur Testfallgenerierung können auf dieser Modellrepräsentation ablaufen und sind damit von der spezifischen Ausprägung der Eingangsspezifikationssprache unabhängig. Auf Grundlage dieser internen Modellrepräsentation kann das allgemeine Testfallgenerierungsproblem in 4 Abstraktionsschichten gegliedert werden:

1. **Synchronisationsalgorithmen** koordinieren die Auswahl von Testdaten, die gleichzeitig an verschiedene sequenzielle Agenten eines zu testenden nebenläufigen Gesamtsystems zu übergeben sind,
2. **Traversionsalgorithmen auf hierarchischen Graphen** dienen zur Auswahl von Pfaden durch die Kontrollstrukturen sequenzieller Agenten, ohne dabei schon die Beschriftungen der Kanten⁵ und Zustände⁶ zu interpretieren. Damit wird eine Obermenge der im System tatsächlich möglichen Abläufe generiert.
3. Mittels **symbolischer Ausführung** werden aus Kanten- und Knotenbeschriftungen abgeleiteten Bedingungen ermittelt, unter denen ein Übergang entlang einer Kante tatsächlich möglich ist.

⁵Bei diese Beschriftungen handelt es sich um Bedingungen, Ereignisse und Aktionen.

⁶Zustände sind mit Eingangsaktionen, Invarianten, Flussbedingungen und algebraischen Bedingungen beschriftet.

4. Mittels unterschiedlicher Lösungsmethoden aus dem Bereich **SAT-Solver** und **Constraint Programming** werden konkrete Daten ermittelt, welche die vorher abgeleiteten Bedingungen erfüllen, so dass durch diese zeitgesteuerten Folgen von Testvektoren die gewünschten Zustandsübergänge und zugehörige Ausgaben des Testlings stimuliert werden.

Zur Optimierung des Testfall- und Testdatengenerierungsverfahrens interagieren die Ebenen kontinuierlich miteinander: Damit können Folgen, die auf Grund einschränkender Synchronisationsbedingungen zwischen nebeneinanderlaufenden Agenten oder wegen restriktiver Invarianten und sonstiger Bedingungen nicht auftreten können, von vornherein aussortiert werden.

Die Pfade durch das Systemverhaltensmodell lassen sich in der Regel durch unterschiedliche Testvektoren stimulieren. Für kontinuierliche Abläufe hilft hier die **Intervallanalyse** bei der Bestimmung von Auswahlintervallen für Testvektorkomponenten, deren zugehörige Daten alle dieselbe Wirkung auf den Kontrollfluss im Testling haben. Diese Information kann zur Konstruktion von Äquivalenzklassentests verwendet werden.

Alle genannten Methoden werden für sich in Modellprüfungsverfahren eingesetzt; die kombinierte Verwendung zur automatisierten Testfallgenerierung ist nach unserer Kenntnis neu.

Im Bereich der automatisierten Testauswertung wurden neue Erkenntnisse gewonnen, auf welche Weise statistische Auswertungsverfahren anzuwenden sind, wenn die Korrektheit der vom Testling gesteuerten Analogabläufe zu beurteilen ist: Vom Testling kontrollierte zeitlich kontinuierliche Abläufe unterliegen mehreren Verfälschungen, die teilweise statistischen Charakter besitzen. Zu dieser Problemstellung wurde gezeigt, wie mit Hilfe statistischer Tests und unterstützender Methoden der Zeitreihenanalyse sowie der Signalfilterung die Entscheidung des Testsystems über Annahme oder Ablehnung des beobachteten Testlingsverhaltens mit einem statistischen **Vertrauensbereich** bewertet werden kann.

Die in den ersten Projektphasen entwickelten Verfahren zum Test (hybrider) Echtzeitsysteme wurden in [Amt99, Mey01, DS04, Pel02, PT02] veröffentlicht. Die Ergebnisse der letzten Phase sind bisher in [KS05, Pel05, PBJH06, PSF06, PBB⁺06] beschrieben, weitere Publikationen sind in Vorbereitung.

Ergebnisspezialisierung auf die Bahndomäne: Im ersten Teil des Projektes HYBRIS wurde zunächst eine neue domänenspezifische Beschreibungssprache für Bahnsteuerungsanwendungen entworfen [PBH00, HP02]. In der

letzten Projektphase wurde herausgearbeitet, dass solche Beschreibungsformalismen wiederum als Profile der UML 2.0 ausgeprägt werden können. Für Bahnsteuerungsanwendungen wurde daher das **Railway Control System Domain Profile RCSD** entwickelt und in [PBD⁺05] vorgestellt. Die vollständige Profildefinition ist in [PBB⁺06] verfügbar und wurde in [BH06] zur Veröffentlichung eingereicht. Nach unserem Kenntnisstand ist dies das erste systematisch definierte UML 2.0 Profil, welches für Bahnsteuerungssysteme angegeben wurde.

Auch beim RCSD-Profil wurde die Verhaltenssemantik wieder transformationell zugeordnet: Für die diskreten Steuerungsaspekte wurde als “leichtgewichtige” Alternative zu HL3 die formale Spezifikationsprache **SystemC** verwendet. Dabei wurden folgende Hauptergebnisse erzielt:

1. Eine große Klasse von Bahnsteuerungen lässt sich aus einem einzigen Framework durch Instanziierung mit den Konfigurationsdaten des konkreten Streckennetzes erzeugen.
2. Die Instanz wird in der Low-Level Repräsentation aus SystemC-Templates mit Hilfe einer geeigneten Codierung der Konfigurationsdaten erzeugt.
3. Das durch 2. entstandene formale Modell kann durch **Bounded Model Checking (BMC)** in Kombination mit einer induktiven Beweisstrategie automatisch in Bezug auf seine Sicherheitseigenschaften verifiziert werden. Hierdurch ist es gelungen, die Verifikation für konkrete Instanzen des Steuerungssystems automatisch und in kurzer Zeit durchzuführen. Dieser Verifikationsschritt galt im Bahnbereich bisher immer als sehr aufwändig, da automatisierte Verfahren noch nicht zur Verfügung standen. Konventionelle Modellprüfungsansätze scheiterten bisher auf diesem Gebiet immer an der Modellgröße. Im Gegensatz dazu kommt BMC mit deutlich kleineren Modellen aus, da diese nur eine beschränkte Anzahl von Verhaltensschritten, beginnend bei einer Teilmenge der insgesamt möglichen Ausgangszustände, codieren müssen.
4. Das SystemC-Modell M_1 kann durch konventionelle Compilation in ausführbaren Code umgesetzt werden. Die besondere Codestruktur, sowie die von uns eingeführte einfache Codierung der Objektzustände ermöglicht es, den erzeugten Objektcode automatisiert zurück in eine SystemC-Repräsentation M_2 zu überführen. Dann lässt sich – wiederum automatisiert – die Modelläquivalenz zwischen M_1 und M_2 zei-

gen, was die Sicherheitseigenschaften des automatisch erzeugten Objektcodes beweist. Mit diesem Verfahren ist die bisher üblicherweise bei sicherheitsrelevanten Bahnanwendungen erforderliche Compiler-Validierung überflüssig geworden.

Die Ergebnisse zu diesen Themen wurden im Detail in [PGHD04, PBD⁺05, PHK⁺06, KPD06] beschrieben. Weitere Resultate über Bahnsteuerungssysteme, ihre Modellierung und Verifikation wurden in [HP99, PBH00, HP00a, HP00b, HP02, HP03b, HP03a] veröffentlicht.

Auf weitere erzielte Ergebnisse wird bei der Vorstellung der wichtigsten aus Hybris entstandenen Qualifikationsarbeiten in Abschnitt 3.8 verwiesen.

3.4 Ausblick und zukünftige Arbeiten

Aus den HYBRIS-Forschungen resultierende weitere Arbeiten lassen sich in Grundlagenforschung und Transferaktivitäten gliedern: Erstere konzentrieren sich auf

1. die Erweiterung des HybridUML-Profiles um probabilistische Systemeigenschaften, mit Hilfe derer sich nicht-funktionale Merkmale wie Verfügbarkeit und Zuverlässigkeit formulieren und verifizieren lassen,
2. die Entwicklung einer zur HL3-Semantik konsistenten High-Level Semantik für HybridUML,
3. die Entwicklung einer für HybridUML maßgeschneiderten *Property Language*, mit der sich hybride temporallogische und probabilistische Zusicherungen zwecks Validierung des Spezifikationsmodells definieren lassen, und
4. die weitere Erforschung der Zusammenhänge zwischen (Bounded) Model Checking und automatisierter Testfallgenerierung.

Die transferorientierten Aktivitäten haben

- (a) Anwendungsreife der prototypisch entwickelten Plug-Ins für die beschriebenen UML 2.0 Profile – optimiert zum industriellen Einsatz im Bahnbereich,
- (b) die automatische Generierung der SystemC-Modellrepräsentation und der zugehörigen Beweisverpflichtungen für den Sicherheitsnachweis aus dem domänenspezifischen Modell, sowie
- (c) die Effizienzsteigerung des Testautomatisierungsprozesses durch situationsabhängige, optimierte Auswahl der verfügbaren Algorithmen,

zum Ziel.

Eine High-Level Semantik laut 2. ermöglicht zusammen mit den Ergebnissen aus 3. die Verifikation von Spezifikationsmodellen, ohne jedesmal in die ausführbare HL3-Zielumgebung übersetzen zu müssen. Da HL3 für die Erzeugung ausführbarer Systeme optimiert ist, lassen sich darin unterbestimmte, nichtdeterministische Modelle, sowie statistische Modelleigenschaften schwer formulieren. Diese sind jedoch für die Untersuchung von Systemvarianten und für die Modellierung zufälliger Ereignisse in der Systemumgebung relevant.

Bei (c) handelt es sich vor allem um die automatische Erkennung unterschiedlicher Problemstellungen der Constraint-Programmierung zur Konstruktion von Testfällen, welche die gezielte Überdeckung bestimmter Verhaltensweisen des zu testenden Systems bewirken. Die genaue Erkennung der Problemstellung ermöglicht den Einsatz optimierter Algorithmen und beschleunigt damit den Testdatenauswahlprozess erheblich.

3.5 Interdisziplinäre Weiterentwicklung

Die im Projekt HYBRIS erarbeiteten Fragestellungen sind für alle Bereiche von Interesse, in denen eingebettete Systeme unternehmens- oder sicherheitskritische Aufgaben übernehmen. Im weitesten Sinne sind dies Fragestellungen der **Automatisierungstechnik**, so dass eine entsprechender Beitrag in einem einschlägigen Journal platziert wurde [BBHP04b]. Weiter spezialisiert wurden die HYBRIS-Ergebnisse vor allem für den Bereich Verkehrssysteme, und dort für die Domänen Bahnsteuerungssysteme und Avionik “exportiert”. Fachbeiträge für diese Bereiche wurden in [PBH00, HP03a, HP00b, HP03b, PGHD04, BZL04] (Bahnanwendungen) und [PT02, PBJH06] (Luftfahrt) publiziert. Die sicherheitsrelevanten Aspekte der in HYBRIS erzielten Resultate sind weiterhin für die Forschungsgruppen von Interesse, die sich allgemein mit Sicherheit im Sinne von **Safety** befassen. Dies führte zu Publikationen bzw. Einreichungen von Beiträgen in den einschlägigen Journalen bzw. Konferenzen [BL05, BH06, PHK⁺06].

3.6 Anwendung

Das Projekt HYBRIS diente als “Anschubmotor” und “methodisches Rückgrat” für

- 2 internationale und 1 nationales anwendungsorientiertes Forschungsprojekte im Luftfahrtbereich:
 - EU-Projekt **VICTORIA** (2000 — 2004)⁷: Hier wurden die HYBRIS-Grundlagen über Testmethoden und das HL3-Framework eingebracht, um – zusammen mit Industriekooperationspartner Verified Systems International GmbH – eine neue Testplattform für moderne Avionik Systeme zu entwickeln. Die Plattform besteht aus einer Bibliothek generischer Testprozeduren zusammen mit einer auf PC-Clustertechnologie basierenden *System Integration-Testbench*, die inzwischen zum Test mehrerer Controller der Airbus-Familie – u. a. Airbus A380 – eingesetzt worden ist.
 - Nationales Projekt **KATO** (2004 – 2007), Forschungsprojekt im Rahmen des LUFO III Programms zur Luftfahrtforschung: Hier arbeitet die Forschungsgruppe des Antragstellers im Unterauftrag von Airbus und behandelt unter anderem die modellbasierte Entwicklung großer Simulationssysteme. Hier wurde von HYBRIS das HybridUML-Profil, sowie das HL3-Framework als Ausführungsumgebung für große verteilte Echtzeitsimulationen eingebracht. Weiterhin wird die in HYBRIS erarbeitete Konzeption, auf UML Profilbasis domänenspezifische Formalismen zu konstruieren, zur Entwicklung von Beschreibungssprachen für generisches Wissen in der Avionik-Domäne eingesetzt.
 - Integriertes EU-Projekt (IP) **E-CAB** (2006 — 2008): In diesem Projekt, welches gerade zum Start im Juni 2006 genehmigt worden ist, werden neue Kommunikationstechnologien zur Steigerung des Passagierkomforts in den Phasen Check-In, Flug, Ankunft am Zielflughafen entwickelt und erprobt. Die Forschungsgruppe des Antragstellers erweitert im Rahmen dieses Projektes die in HYBRIS und VICTORIA entwickelte Testtechnologie auf automatisierte Konformitätstests für zusätzliche neue Hardwareausprägungen des Testlings (Mobiltelefone, PDAs) und zugehörige Kommunikationsprotokolle.

⁷Validation platform for Integration of standardised Components, Technologies and tools in an Open, modulaR and Improved Aircraft electronic system siehe <http://www.informatik.uni-bremen.de/agbs/projects/victoria/>.

- Die Industriekooperation **AzSME** (2004 – 2006) im Bereich Bahnsteuerungssysteme (Siemens AG): In dieses Projekt werden die HYBRIS-Ergebnisse bzgl. modellbasierter Entwicklung, Testautomatisierung, sowie die domänenspezifische Systemspezifikation mittels des RCSD-Profiles eingebracht.
- Eine Industriekooperation mit Verified Systems International GmbH im Bereich **Testautomatisierung** (2002 — 2006). Dort werden Algorithmen, sowie das HL3-Framework zur Ausführung nebenläufiger Simulationen und Testorakel in das von Verified Systems entwickelte und u. a. von Airbus und Siemens eingesetzte Testwerkzeug RT-Tester integriert.
- Das **Doktorandenkolleg Eingebettete Systeme (GESy)**⁸ wurde 2005 durch den Antragsteller und den Kooperationspartner Rolf Drechsler an der Universität Bremen gegründet. Das Kolleg wird durch eine kombinierte Stipendiatenförderung aus Industriemitteln und einer Finanzierung durch die Universität Bremen realisiert. Bis Ende 2006 wird das Programm auf die Zielgröße von ca. 10 Stipendiaten gewachsen sein.
- Durch den Erfahrungsaustausch zwischen der Forschungsgruppe des Antragstellers und der Gruppe von Werner Damm an der Universität Oldenburg im Rahmen des Schwerpunktprogramms 1064 schloss sich der Antragsteller zusammen mit Rolf Drechsler der von Werner Damm 2005 initiierten Forschungsinitiative **Safety in Transportation Systems (SafeTRANS)**⁹ an. Hier werden zahlreiche Ansätze aus dem Schwerpunktprogramm auf Verkehrssysteme in den Bereichen Bahn, Avionik und Automobilelektronik erweitert. Der Antragsteller ist dort Mitglied des *Scientific Steering Board* und zusammen mit Jens Braband (Siemens AG und Technische Universität Braunschweig) fachlich für den Bereich *Safety and Certification Process* verantwortlich.

⁸<http://www.informatik.uni-bremen.de/gesy>

⁹<http://www.safetrans-de.org/>

3.7 Beteiligte Wissenschaftler

Beiträge des Antragstellers (Jan Peleska) zum Projekt HYBRIS: Der Antragsteller entwickelte die wissenschaftlichen Zielsetzungen des Projektes. Er lieferte die Hauptbeiträge zu den Bereichen *automatische Codegenerierung, Modellprüfung und Codevalidierung für Bahnsteuerungen* [HP98, HP99, PBH00, HP00a, HP00b, HP02, HP03b, HP03a, PGHD04, PHK⁺06] und Automatische Testfallerstellung und Test Orakel für (Hybride) Echtzeitsysteme [Pel02, Pel05, PT02, PBB⁺06, PSF06].

Seine Beiträge zu den weiteren oben beschriebenen Gebieten waren konzeptioneller Natur und wurden im Detail von den nachfolgend genannten Mitarbeitern ausgeführt. In Kooperation mit dem Industriepartner Verified Systems International GmbH leitet er verantwortlich die Integration der Testautomatisierungsalgorithmen in das Testwerkzeug RT-Tester.

Beiträge des von der DFG geförderten Wissenschaftlers (Stefan Bisanz) zum Projekt HYBRIS: Stefan Bisanz lieferte im Rahmen seiner Projektstätigkeit die Hauptbeiträge zum HybridUML Profil, der Hybrid Low-Level Language HL3, sowie zur Transformation von HybridUML nach HL3 und den zugehörigen Fallstudien [BBHP03, BBHP06, BBHP04a, BBHP04b, BZL04, Bis05]. Er entwickelte weiterhin maßgeblich einen Ansatz zur Entwicklung von Testfällen in virtueller Realität [PBIM99, BT02a, BT02b]

Beiträge weiterer Projektmitarbeiter: Holger Schlingloff¹⁰ leitete das Projekt zusammen mit dem Antragsteller in der ersten Projektphase und lieferte im Rahmen seiner Habilitation die wesentlichen Beiträge zum Bereich Partialordnungsverfahren. Bettina Buth und Jan Bredereke bearbeiteten in ihren Habilitationen ebenfalls Fragestellungen aus HYBRIS. Das gleiche gilt für die Dissertationen von Peter Amthor (Projektphase 1), Oliver Meyer, Detlef Kendelbacher, Markus Dahlweid und Uwe Schulze (Projektphasen 1 und 2); für weitere Details siehe Abschnitt 3.8. Kirsten Berkenkötter und Ulrich Hannemann (Projektphase 3) leisteten den Hauptbeitrag zum RCSD Profil für Bahnsteuerungsanwendungen [PBB⁺06, BH06] und arbeiteten auf den Gebieten HybridUML und HL3 mit. Elena Svetogorova (Phase 3) trug zum Bereich *Testorakel für Hybride Systeme* bei.

¹⁰Inzwischen Professor an der Humboldt Universität zu Berlin, <http://www2.informatik.hu-berlin.de/hs/>

Beiträge von Kooperationspartnern: Anne Haxthausen (Denmark Technical University) und Rolf Drechsler (Universität Bremen), zusammen mit Sebastian Kinder und Daniel Große aus seiner Forschungsgruppe kooperierten auf dem Gebiet der Bahnsteuerungssysteme und ihrer Verifikation. Zusammen mit dem Projektpartner im SPP 1064, Eckehard Schnieder und seiner Forschungsgruppe wurde die Umsetzung der Fallstudie im Bahnmodell der TU Braunschweig vorgenommen.

3.8 Weiterqualifikation

Im Projekt HYBRIS sind

- 3 Habilitationen,
- 5 Dissertationen,
- 10 Diplomarbeiten,
- 1 Projektbericht aus einem 4-semesterigen studentischen Projekt

entstanden. Titel und Autoren sind in Abschnitt 4.4 aufgeführt.

Die Habilitation von Holger Schlingloff [Sch01] behandelt unter anderem die Lösung der HYBRIS-Fragestellungen bzgl. Partialordnungsverfahren. Die Arbeit beschreibt verschiedene Methoden zur Verifikation und Validierung sicherheitskritischer Systeme, die im Rahmen des Projektes entstanden sind. Sie fasst dabei im ersten Teil zunächst einige auf dem Gebiet der zustandsorientierten Spezifikation gebräuchliche Formalismen zusammen: Automatenbasierte und algebraische Modellierungssprachen, temporale Logiken, sowie Methoden zur Transformation von Modellen. Darauf aufbauend werden im zweiten Teil die bekannten Modellprüfungs- und Abstraktionsverfahren in einem einheitlichen Rahmen und an praxisorientierten Beispielen dargestellt. Im dritten Teil werden diese Methoden auf Realzeit- und hybride Systeme ausgeweitet, die im Kontext des Projektes HYBRIS untersucht wurden. Es wird ein neuer Verifikationsalgorithmus entwickelt, bei dem der (potentiell unendliche) Zustandsraum eines Realzeitautomaten auf einen leicht analysierbaren Teil reduziert werden kann. Ein weiteres Ergebnis ist ein neuartiger Konformanzbegriff für hybride Systeme, mit dem sich Sicherheits- und Zeitanforderungen leicht verifizieren lassen. Im vierten Teil schließlich werden Ergebnisse vorgestellt, die eine hohe Relevanz für die Praxis ingenieurwissenschaftlicher Anwendungen haben, nämlich für das Debugging und den Test eingebetteter Systeme. Zum Debugging paralleler reaktiver Programme wird ein auf Modellprüfungsverfahren aufbauender Algorithmus vorgestellt, mit dem Fehler bei einer Programmausführung entdeckt werden können, selbst wenn sie dabei durch die Ablaufsteuerung maskiert waren. Für den Test eingebetteter Steuergeräte wird ein Hardware-in-the-Loop Ansatz vorgestellt, mit dem die Korrektheit einer Implementierung gegenüber einer formalen Spezifikation systematisch und automatisiert überprüft werden kann. Die in der Habilitationsschrift vorgestellten Ergebnisse wurden überwiegend in der ersten Phase des HYBRIS-Projektes

erzielt und bildeten eine Grundlage für die Arbeiten der zweiten und dritten Projektphase.

Die Habilitationen von Bettina Buth [But01] und Jan Brederke [Bre05b] behandeln verschiedene Anwendungen von Modellprüfungsverfahren für reaktive Systeme und insbesondere Fragestellungen der **Mode Confusion**¹¹ und **Feature Interaction**¹², deren Analyse für sicherheitsrelevante Systeme auf Grund der zunehmenden Systemkomplexität heute von besonderer Bedeutung sind.

Die Dissertationen von Peter Amthor [Amt99], Oliver Meyer [Mey01], Markus Dahlweid und Uwe Schulze [DS04] behandelten Fragestellungen bzgl. des Testens von Echtzeitsystemen gegen Hybrid CSP und Timed CSP Spezifikationen, sowie die optimale interne Repräsentation solcher Spezifikationsmodelle. Ihre Arbeiten führten zu weiteren Testautomatisierungsalgorithmen, die in [Pel02, Pel05, PT02] veröffentlicht wurden. Stefan Bisanz erarbeitete im Rahmen seiner Dissertation [Bis05] die entscheidenden Beiträge für HybridUML und das Hybrid Low-Level Framework HL3. Die Dissertation von Detlef Kendelbacher [Ken04] behandelt spezielle Fragestellungen bzgl. Entwurf und Verifikation eingebetteter Systeme im Bahnbereich, insbesondere die sichere Interaktion zwischen nicht sicheren und sicherheitsrelevanten Subsystemen.

¹¹In der Regel sicherheitskritische Inkonsistenzen zwischen dem mentalen Modell des für die Systemsteuerung verantwortlichen Menschen und dem in den technischen Steuerungssystemen codierten Zuständen.

¹²Ungewollte Wechselwirkungen zwischen unterschiedlichen Funktionen eines Systems.

4 Publikationen

Im folgenden werde alle im Rahmen des Projektes entstandenen Publikationen nach Kategorien geordnet und bei Bedarf in Bezug auf den darin dargestellten Projektbeitrag aufgeführt. Dabei wird in Bezug auf die vollständige Publikationsreferenz auf das nachfolgenden alphabetische Literaturverzeichnis verwiesen.

Die meisten Publikationen sind Online unter http://www.informatik.uni-bremen.de/agbs/jp/jp_papers_e.html verfügbar; weitere Web-Links sind im nachfolgenden Literaturverzeichnis angegeben. Daher wurde auf die Übersendung von Sonderdrucken im Rahmen des Abschlussberichtes verzichtet.

4.1 Publikationen in Fachzeitschriften

1. **IEEE Transaction on Software Engineering [HP00a]**: A. E. Haxthausen and J. Peleska: *Formal Development and Verification of a Distributed Railway Control System*.
2. **IEEE Communications Magazine [Bre02]**: Jan Brederke: Maintaining Telephone Switching Software Requirements. (HYBRIS-Bezug: Feature Interaction).
3. **atp – Automatisierungstechnische Praxis [BBHP04b]**: Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann and Jan Peleska: *Spezifikation von Echtzeit-Automatisierungssystemen mit HybridUML*.
4. **Journal of Reliability Engineering & System Safety [BL05]**: Jan Brederke and Axel Lankenau: Safety-Relevant Mode Confusions – Modelling and Reducing Them.
5. **International Journal on Software Tools for Technology Transfer (STTT) [BBHP06]**: Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann and Jan Peleska: The HybridUML Profile for UML 2.0.

4.2 Kongressbeiträge

1. **Kongressbeitrag 13th European Simulation Multiconference 1999 [PBIM99]**: Jan Peleska, Stefan Bisanz, Ingo Fiß and Manfred Endreß: *Non-Standard Graphical Simulation Techniques for Test Specification Development*.

2. **Kongressbeitrag Formal Methods World Congress FM'99 [HP99]:** Anne E. Haxthausen und Jan Peleska: *Formal Development and Verification of a Distributed Railway Control System.*
3. **Kongressbeitrag 9th IFAC Symposium on Control in Transportation Systems 2000 [PBH00]:** J. Peleska, A. Baer and A. E. Haxthausen: *Towards Domain-Specific Formal Specification Languages for Railway Control Systems.*
4. **Kongressbeitrag FORMS '99 - Formale Techniken für die Eisenbahnsicherung [HP00b]:** A. E. Haxthausen and J. Peleska: *Formal Methods for the Specification and Verification of Distributed Railway Control Systems: From Algebraic Specifications to Distributed Hybrid Real-Time Systems.*
5. **Kongressbeitrag FM-TOOLS 2002 – The 5th Workshop on Tools for System Design and Verification [BT02a]:** Stefan Bisanz and Aliko Tsiolakis: *Test Development in Virtual Environments.*
6. **Kongressbeitrag Formal Approaches to Testing of Software. FATES'02 – A Satellite Workshop of CONCUR'02 [BT02b]:** Stefan Bisanz and Aliko Tsiolakis: *Using a Virtual Reality Environment to Generate Test Specifications.*
7. **Kongressbeitrag Sixth Biennial World Conference on Integrated Design and Process Technology, (IDPT2002) [HP02]:** A. E. Haxthausen and J. Peleska: *A Domain Specific Language for Railway Control Systems.*
8. **Kongressbeitrag Sixth Biennial World Conference on Integrated Design and Process Technology, (IDPT2002) [Pel02]:** Jan Peleska: *Formal Methods for Test Automation - Hard Real-Time Testing of Controllers for the Airbus Aircraft Family.*
9. **Kongressbeitrag SVERTS Workshop at the «UML» 2003 Conference [Ber03]:** Kirsten Berkenkötter: *Using UML 2.0 in Real-Time Development - A Critical Review.*
10. **Kongressbeitrag SVERTS Workshop at the «UML» 2003 Conference [BBHP03]:** Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hanneemann and Jan Peleska: *HybridUML Profile for UML 2.0.*

11. **Kongressbeitrag Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'2003) [HP03b]:** A. E. Haxthausen and J. Peleska: *Generation of Executable Railway Control Components from Domain-Specific Descriptions.*
12. **Kongressbeitrag 10th IFAC Symposium on Control in Transportation Systems 2003 [HP03a]:** A. E. Haxthausen and J. Peleska: *Automatic Verification, Validation and Test for Railway Control Systems based on Domain-Specific Descriptions.*
13. **Kongressbeitrag FORMS/FORMAT 2004 - Formal Methods for Automation and Safety in Railway and Automotive Systems 2004 [PGHD04]:** Jan Peleska, Daniel Große, Anne E. Haxthausen and Rolf Drechsler: *Automated Verification for Train Control Systems.*
14. **Kongressbeitrag FORMS/FORMAT 2004. Formal Methods for Automation and Safety in Railway and Automotive Systems [BZL04]:** Stefan Bisanz, Paul Ziemann and Arne Lindow: *Integrated Specification, Validation and Verification with HybridUML and OCL applied to the BART Case Study.*
15. **Kongressbeitrag Communicating Sequential Processes: The First 25 Years. Symposium on the Occasion of 25 Years of CSP 2004 [Pel05]:** Jan Peleska: *Applied Formal Methods - From CSP to Executable Hybrid Specifications.*
16. **Kongressbeitrag 6th ICSTEST International Conference on Software Testing [OH05]:** Aliko Ott and Tobias Hartmann: *Domain Specific V&V Strategies for Aircraft Applications.*
17. **Kongressbeitrag TRain@SEFM2005 workshop, Koblenz, Germany, September, 5th - 6th [PBD⁺05]:** Jan Peleska, Kirsten Berkenkötter, Rolf Drechsler, Daniel Große, Ulrich Hannemann, Anne E. Haxthausen and Sebastian Kinder: *Domain-Specific Formalisms and Model-Driven Development for Railway Control Systems.*
18. **Eingereichter Beitrag für SAFECOMP'2006 [PHK⁺06]:** Jan Peleska, Anne E. Haxthausen, Sebastian Kinder, Daniel Große and Rolf Drechsler: *Model-Driven Development and Verification in the Railway Domain.*
19. **Eingereichter Beitrag für SAFECOMP'2006 [BH06]:** Kirsten Berkenkötter and Ulrich Hannemann: *Modeling the Railway Control domain Rigorously with a UML 2.0 Profile.*

20. **Eingereichter Beitrag für Memocode (Fourth ACM-IEEE International Conference on Formal Methods and Models for Codesign) 2006 [KPD06]:** Sebastian Kinder, Jan Peleska and Rolf Drechsler: *Modeling and Validation of Counting Heads for Railways using SystemC.*
21. **In Vorbereitung befindlicher Beitrag zu 1st Joint International Workshop on FORMAL ASPECTS OF TESTING AND RUNTIME VERIFICATION FATES/RV 2006 [PSF06]:** Jan Peleska, Elena Svetogorova und Martin Fränzle: *Model-Based Testing of Hybrid Systems – Challenges and Solutions.*

4.3 Buchbeiträge

1. **Lecture Notes in Computer Science [BBHP04a]:** Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann and Jan Peleska: *Executable HybridUML and Its Application to Train Control Systems.*
2. **Lecture Notes in Computer Science [Pel05]:** Jan Peleska: *Applied Formal Methods - From CSP to Executable Hybrid Specifications.*
3. **Feature Interactions in Telecommunications and Software Systems VIII [Bre05a]:** Jan Brederke: *Configuring Members of a Family of Requirements Using Features.*

4.4 Studien- und Diplomarbeiten, Dissertation, Habilitationen, Berichte, sonstige Publikationen

1. **Habilitation [But01]:** Bettina Buth: *Formal and Semi-Formal Methods for the Analysis of Industrial Control Systems.*
2. **Habilitation [Sch01]:** Holger Schlingloff: *Partial State Space Analysis of Safety-Critical Systems.*
3. **Habilitation [Bre05b]:** Jan Brederke: *Maintaining Families of Rigorous Requirements for Embedded Software Systems.*
4. **Dissertation [Amt99]:** Peter Amthor: *Structural Decomposition of Hybrid Systems. Test Automation for Hybrid Reactive Systems.*
5. **Dissertation [Mey01]:** Oliver Meyer: *Structural Decomposition of Timed CSP and its Application in Real-Time Testing.*

6. **Dissertation [Ken04]:** Detlef Kendelbacher: *Architekturkonzept und Designaspekte einer signaltechnisch nichtsicheren Kommunikationsplattform für sicherheitsrelevante Bahnanwendungen.*
7. **Dissertation [DS04]:** Markus Dahlweid and Uwe Schulze: *High Level Transition Systems of CSP Specifications and their Application in Automated Testing.*
8. **Dissertation [Bis05]:** Stefan Bisanz: *Executable HybridUML Semantics. A Transformation Definition.*
9. **Diplomarbeit [Ker00]:** Michael Kersten: *Die symbolische Ausführung von CSP-Spezifikationen.*
10. **Diplomarbeit [Geh01]:** Joerg Gehrken: *Automatische Life- und Deadlockanalyse von CSP-Spezifikation basierend auf Kommunikationsgraphen.*
11. **Diplomarbeit [Sta01]:** Torsten Stange: *Wiederverwendung von Software - Entwurfsmuster für sicherheitsrelevante Systeme.*
12. **Diplomarbeit [Hap02]:** Mark Hapke: *Treiberarchitektur für eine Echtzeiterweiterung eines Linux-Multiprozessorsystems.*
13. **Diplomarbeit [Zwe02]:** Klaas-Henning Zweck: *Kernelbasierte Echtzeiterweiterung eines Linux-Multiprozessorsystems.*
14. **Diplomarbeit [Efk05]:** Christof Efke: *Development and evaluation of a hard real-time scheduling modification for Linux 2.6.*
15. **Diplomarbeit [KS05]:** Tatiana Kotas und Christian Stelljes: *Automatisierte Testdatengenerierung für Transitionssysteme.*
16. **In Fertigstellung begriffene Diplomarbeit [Ada06]:** Johannes Adams: *Modell-basierte Codegenerierung für HybridUML Statecharts zum Einsatz bei sicherheitsrelevanten Anwendungen.*
17. **In Fertigstellung begriffene Diplomarbeit [Den06]:** Zhou Deng: *UML Profile and its application in Railway Control System Domain.*
18. **In Fertigstellung begriffene Diplomarbeit [Ngu06]:** Franck Ngueuleu: *Entwurf und Realisierung einer Demonstratoranbindung für die verteilte Simulation von HybridUML-Modellen am Beispiel Funkbasierter Fahrbetrieb.*

19. **Projektbericht [Löd05]**: Studentisches Projekt *TRACS - TRAIIn Control Systems*. – Projektbericht.

Literatur

- [Ada06] Johannes Adams. Modell-basierte Codegenerierung für Hybriduml Statecharts zum Einsatz bei sicherheitsrelevanten Anwendungen. Master's thesis, Universität Bremen, erscheint 2006.
- [ADE⁺03] R. Alur, T. Dang, J. Esposito, Y. Hur, F. Ivančić, V. Kumar, I. Lee, P. Mishra, G. Pappas, and O. Sokolsky. Hierarchical hybrid modeling and analysis of embedded systems. *Proceedings of the IEEE*, 91(1):11–28, January 2003.
- [Amt99] Peter Amthor. *Structural Decomposition of Hybrid Systems. Test Automation for Hybrid Reactive Systems*. PhD thesis, Universität Bremen, BISS Monographs, No. 13, October 1999.
- [BBHP03] Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann, and Jan Peleska. HybridUML Profile for UML 2.0. SVERTS Workshop at the UML 2003 Conference, October 2003. <http://www-verimag.imag.fr/EVENTS/2003/SVERTS/>.
- [BBHP04a] Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann, and Jan Peleska. Executable HybridUML and its Application to Train Control Systems. In Hartmut Ehrig, Werner Damm, Jörg Desel, Martin Große-Rhode, Wolfgang Reif, Eckehard Schnieder, and Engelbert Westkämper, editors, *SoftSpez Final Report*, volume 3147 of *Lecture Notes in Computer Science*, pages 145–173. Springer, September 2004.
- [BBHP04b] Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann, and Jan Peleska. Spezifikation von Echtzeit-Automatisierungssystemen mit HybridUML. *atp – Automatisierungstechnische Praxis*, 46(8):54–60, August 2004. ISSN 0178-2320.
- [BBHP06] Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann, and Jan Peleska. The HybridUML Profile for UML 2.0. *International Journal on Software Tools for Technology Transfer (STTT)*, January 2006. Special Section on Specification and Validation of Models of Real Time and Embedded Systems with UML.
- [Ber03] Kirsten Berkenkötter. Using UML 2.0 in real-time development - a critical review. SVERTS Workshop at the UML 2003 Conference, October 2003. <http://www-verimag.imag.fr/EVENTS/2003/SVERTS/>.
- [BH06] Kirsten Berkenkötter and Ulrich Hannemann. Modeling the railway control domain rigorously with a uml 2.0 profile. In *Submitted to the Safecom 2006 conference*, 2006.
- [Bis05] Stefan Bisanz. *Executable HybridUML Semantics. A Transformation Definition*. PhD thesis, University of Bremen, December 2005. Available as <http://nbn-resolving.de/urn:nbn:de:gbv:46-diss000101662>.
- [BL05] Jan Brederke and Axel Lankenau. Safety-relevant mode confusions – modeling and reducing them. *Reliability Engineering & System Safety*, 88(3):229–245, June 2005. Elsevier, ISSN 0951-8320.
- [Bre02] Jan Brederke. Maintaining telephone switching software requirements. *IEEE Communications Magazine*, 40(11):104–109, November 2002.
- [Bre05a] Jan Brederke. Configuring members of a family of requirements using features. In Stephan Reiff-Marganiec and Mark D. Ryan, editors, *Feature Interactions in Telecommunications and Software Systems VIII*, pages 96–113, Amsterdam, June 2005. IOS Press.

- [Bre05b] Jan Bredereke. *Maintaining Families of Rigorous Requirements for Embedded Software Systems*. Habilitation Thesis. University of Bremen, 2005.
- [BT02a] Stefan Bisanz and Aliko Tsiolakis. Test Development in Virtual Environments. In Dominik Haneberg, Gerhard Schellhorn, and Wolfgang Reif, editors, *FM-TOOLS 2002. The 5th Workshop on Tools for System Design and Verification*, pages 65–69, Augsburg, June 2002. Universität Augsburg, Institut für Informatik. Technical Report 2002-11.
- [BT02b] Stefan Bisanz and Aliko Tsiolakis. Using a Virtual Reality Environment to Generate Test Specifications. In Rob Hierons and Thierry Jéron, editors, *Formal Approaches to Testing of Software. FATES'02 – A Satellite Workshop of CONCUR'02*, pages 121–135. INRIA, August 2002.
- [But01] Bettina Buth. *Formal and Semi-Formal Methods for the Analysis of Industrial Control Systems*. Number 15 in BISS Monographs. University of Bremen, May 2001. Habilitation Thesis.
- [BZL04] Stefan Bisanz, Paul Ziemann, and Arne Lindow. Integrated Specification, Validation and Verification with HybridUML and OCL applied to the BART Case Study. In Eckehard Schnieder and Géza Tarnai, editors, *FORMS/FORMAT 2004. Formal Methods for Automation and Safety in Railway and Automotive Systems*, pages 191–203, Braunschweig, December 2004. Proceedings of Symposium FORMS/FORMAT 2004, Braunschweig, Germany, 2nd and 3rd December 2004. ISBN 3-9803363-8-7.
- [Den06] Zhou Deng. Uml profile and its application in railway control system domain. Master's thesis, Universität Bremen, erscheint 2006.
- [DS04] Markus Dahlweid and Uwe Schulze. *High Level Transition Systems of CSP Specifications and their Application in Automated Testing*. PhD thesis, University of Bremen, 2004. Available under <http://elib.suub.uni-bremen.de/publications/dissertations/E-Diss781.ciao.pdf>.
- [Efk05] Christof Efke. Development and evaluation of a hard real-time scheduling modification for linux 2.6. Master's thesis, Universität Bremen, April 2005.
- [Geh01] Joerg Gehrken. Automatische Life- und Deadlockanalyse von CSP-Spezifikation basierend auf Kommunikationsgraphen. Master's thesis, Universität Bremen, 2001.
- [Hap02] Mark Hapke. Treiberarchitektur für eine Echtzeiterweiterung eines Linux-Multiprozessorsystems. Master's thesis, Universität Bremen, 2002.
- [Hen96] Thomas A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS)*, pages 278–292. IEEE Computer Society Press, 1996.
- [HHK03] Thomas A. Henzinger, Benjamin Horowitz, and Christoph M. Kirsch. Giotto: A time-triggered language for embedded programming. *Proceedings of the IEEE*, 91:84–99, 2003.
- [HP98] A. Haxthausen and J. Peleska. Formal Development and Verification of a Distributed Railway Control System. In *Proceedings of First FMERail Workshop*, 1998.
- [HP99] A. E. Haxthausen and J. Peleska. Formal Development and Verification of a Distributed Railway Control System. In *Proceedings of Formal Methods World Congress FM'99*, number 1709 in Lecture Notes in Computer Science, pages 1546 – 1563. Springer-Verlag, 1999.

- [HP00a] A. E. Haxthausen and J. Peleska. Formal Development and Verification of a Distributed Railway Control System. *IEEE Transaction on Software Engineering*, 26(8):687–701, 2000.
- [HP00b] A. E. Haxthausen and J. Peleska. Formal Methods for the Specification and Verification of Distributed Railway Control Systems: From Algebraic Specifications to Distributed Hybrid Real-Time Systems. In *Forms '99 - Formale Techniken für die Eisenbahnsicherung Fortschritt-Berichte VDI, Reihe 12, Nr. 436*, pages 263–271. VDI-Verlag, Düsseldorf, 2000.
- [HP02] A. E. Haxthausen and J. Peleska. A Domain Specific Language for Railway Control Systems. In *Proceedings of the Sixth Biennial World Conference on Integrated Design and Process Technology, (IDPT2002), Pasadena, California, June 23-28 2002*.
- [HP03a] A. E. Haxthausen and J. Peleska. Automatic Verification, Validation and Test for Railway Control Systems based on Domain-Specific Descriptions. In *Proceedings of the 10th IFAC Symposium on Control in Transportation Systems*. Elsevier Science Ltd, Oxford, 2003.
- [HP03b] A. E. Haxthausen and J. Peleska. Generation of Executable Railway Control Components from Domain-Specific Descriptions. In *Proceedings of the Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'2003), Budapest/Hungary*, pages 83–90. LHarmattan Hongrie, May 15-16 2003.
- [Jif94] He Jifeng. *From CSP to Hybrid Systems*, pages 171–189 in [Ros94]. 1994.
- [Ken04] Detlef Kendelbacher. *Architekturkonzept und Designaspekte einer signaltechnisch nichtsicheren Kommunikationsplattform für sicherheitsrelevante Bahnwendungen*. PhD thesis, Universität Bremen, BISS Monographs, No. 13, December 2004. Dissertation, verfügbar unter http://elib.suub.uni-bremen.de/publications/dissertations/E-Diss835_dis_50b.pdf.
- [Ker00] Michael Kersten. *Die symbolische Ausführung von CSP-Spezifikationen*. Master's thesis, Universität Bremen, 2000.
- [KPD06] Sebastian Kinder, Jan Peleska, and Rolf Drechsler. Modeling and validation of counting heads for railways using systemc. In *Submitted to Memocode (Fourth ACM-IEEE International Conference on Formal Methods and Models for Codesign) 2006, Napa Valley, California, 27th – 29th July 2006*.
- [KS05] Tatiana Kotas and Christian Stelljes. *Automatisierte testdatengenerierung für transitionssysteme*. Master's thesis, University of Bremen, 2005.
- [Löd05] Helge Löding. *TRACS - TRAIIn Control Systems*. Projektbericht, September 2005. Studentisches Projekt an der Universität Bremen.
- [Mey01] Oliver Meyer. *Structural Decomposition of Timed CSP and its Application in Real-Time Testing*. PhD thesis, Universität Bremen, BISS Monographs, No. 16, July 2001.
- [Ngu06] Franck Ngueuleu. *Entwurf und Realisierung einer Demonstratoranbindung für die verteilte Simulation von HybridUML-Modellen am Beispiel Funkbasierter Fahrbetrieb*. Master's thesis, Universität Bremen, erscheint 2006.
- [OH05] Aliko Ott and Tobias Hartmann. Domain Specific V&V Strategies for Aircraft Applications. In *Proc. of the 6th ICSTEST International Conference on Software Testing*, Düsseldorf, April 2005.

- [PBB⁺06] Jan Peleska, Kirsten Berkenkötter, Stefan Bisanz, Ulrich Hannemann, and Elena Svetogorova. HYBRIS – Efficient Specification and Analysis of Hybrid Systems. Technical report, Universität Bremen, 2006. Available under <http://www.informatik.uni-bremen.de/agbs/jp/papers/hybris2006.html>.
- [PBD⁺05] Jan Peleska, Kirsten Berkenkötter, Rolf Drechsler, Daniel Große, Ulrich Hannemann, Anne E. Haxthausen, and Sebastian Kinder. Domain-specific formalisms and model-driven development for railway control systems. Presentation at TRain@SEFM2005 workshop, Koblenz, Germany, September, 5th - 6th 2005. Presentation slides available under http://www.tzi.de/agbs/jp/papers/peleska_et_al_train2005_slides.pdf.
- [PBH00] J. Peleska, A. Baer, and A. E. Haxthausen. Towards Domain-Specific Formal Specification Languages for Railway Control Systems. In *Proceedings of the 9th IFAC Symposium on Control in Transportation Systems 2000, June 13-15, 2000, Braunschweig, Germany*, pages 147–152, 2000.
- [PBIM99] J. Peleska, S. Bisanz, Fiß I., and Endreß M. Non-Standard Graphical Simulation Techniques for Test Specification Development. In H. Szczerbicka, editor, *Modelling and simulation: A tool for the next millenium. 13th European Simulation Multiconference 1999*, volume 1, pages 575–580, Delft, 1999. Society for Computer Simulation International.
- [PBJH06] Jan Peleska, Klemens Brumm, Gunnar Jonas, and Tobias Hartmann. Advancement in automated simulation and testing technology for safety-critical avionic systems. In *Presentation on Aerospace Testing 2006, Hamburg, April, 4 - 6, 2006.*, April 2006. Available under http://www.informatik.uni-bremen.de/agbs/jp/papers/peleska_et_al_AeroSpaceTesting2006-HH.html.
- [Pel02] Jan Peleska. Formal methods for test automation - hard real-time testing of controllers for the airbus aircraft family. In *Proc. of the Sixth Biennial World Conference on Integrated Design & Process Technology (IDPT2002), Pasadena, California*. Society for Design and Process Science, June 2002. To appear.
- [Pel05] Jan Peleska. Applied formal methods - from CSP to executable hybrid specifications. In Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders, editors, *Communicating Sequential Processes: The First 25 Years. Symposium on the Occasion of 25 Years of CSP, London, UK, July 7-8, 2004. Revised Invited Papers*, volume 3525 of LNCS, pages 293–320. Springer-Verlag GmbH, 2005.
- [PGHD04] Jan Peleska, Daniel Große, Anne E. Haxthausen, and Rolf Drechsler. Automated verification for train control systems. In E. Schnieder and G. Tarnai, editors, *Proceedings of the FORMS/FORMAT 2004 - Formal Methods for Automation and Safety in Railway and Automotive Systems*, pages 252–265. Technical University of Braunschweig, December 2004. ISBN 3-9803363-8-7.
- [PHK⁺06] Jan Peleska, Anne E. Haxthausen, Sebastian Kinder, Daniel Große, and Rolf Drechsler. Model-driven development and verification in the railway domain. In *Submitted to SAFECOMP'2006, The 25th International Conference on Computer Safety, Reliability and Security. Gdansk, Poland, 26-29 September 2006*.
- [PSF06] Jan Peleska, Elena Svetogorova, and Martin Fränzle. Model-based testing of hybrid systems – challenges and solutions. In *1st Joint International Workshop on Formal Aspects of Testing and Runtime Verification, FATES/RV, Seattle, Washington, USA, August 15/16, 2006*, 2006. In preparation for submission.

- [PT02] J. Peleska and A. Tsiolakis. Automated Integration Testing for Avionics Systems. In *Proceedings of the 3rd ICSTEST – International Conference on Software Testing*, April 2002.
- [Ros94] A.W. Roscoe, editor. *A Classical Mind, Essays in Honour of C.A.R. Hoare*. International Series in Computer Science. Prentice Hall, 1994.
- [Sch01] Holger Schlingloff. *Partial State Space Analysis of Safety-Critical Systems*. Habilitation Thesis. University of Bremen, 2001.
- [Sta01] Torsten Stange. Wiederverwendung von Software - Entwurfsmuster für sicherheitsrelevante Systeme. Master's thesis, Universität Bremen, 2001.
- [ZRH93] Chaochen Zhou, A. P. Ravn, and M. R. Hansen. An extended duration calculus for hybrid real-time systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 763 of *Lecture Notes in Computer Science*, pages 36–59. The Computer Society of the IEEE, 1993. Extended abstract.
- [Zwe02] Klaas-Henning Zweck. Kernelbasierte Echtzeiterweiterung eines Linux-Multiprozessorsystems. Master's thesis, Universität Bremen, 2002.