**ANNUAL REPORT**

# FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

**NSF/ITR PROJECT – AWARD NUMBER: CCR-0225610**

**UNIVERSITY OF CALIFORNIA, BERKELEY**
**VANDERBILT UNIVERSITY**
**UNIVERSITY OF MEMPHIS**

**MAY 31, 2006**
**Revised 5 June**

**PERIOD OF PERFORMANCE COVERED: JUNE 1, 2005 – MAY 31, 2006**

## Contents


Contents 2
1. Participants 6
1.1. People 6
1.2. Partner Organizations: 9
1.3. Collaborators: 9
2. Activities and Findings 12
2.1. Project Activities 12
2.1.1. Hybrid Systems Theory 12
2.1.1.a. Deep Compositionality 13
*Trading Latency for Composability* *13*
*Non-Zero-Sum Games as Compositional System Models* *13*
*Hybrid Systems Modeling Tools: a Survey* *14*
*Graphs and games* *14*
2.1.1.b. Robust Hybrid Systems 14
*Design and Verification of Robust System Models* *14*
*A Homology Theory for Hybrid Systems* *14*
2.1.1.c. Computational Hybrid Systems 15
*Algorithms for the Control of Stochastic Systems* *15*
*Reach Set Calculations using Ellipsoidal Approximations* *15*
*A Deterministic Operational Semantics for Hybrid System Simulations* *15*
*Building Efficient Simulations from Hybrid Bond Graph Models* *16*
*Going Beyond Zeno* *18*
2.1.1.d. Stochastic Hybrid Systems 18
*Stochastic Approximations of Hybrid Systems* *18*
*Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems* *19*
*Adjoint-based Optimal Control of the Expected Exit Time for Stochastic Hybrid Systems* *19*
*Inference Methods for Autonomous Stochastic Linear Hybrid Systems* *19*
2.1.2. Model-Based Design 20
2.1.2.a. Composition of Domain Specific Modeling Languages 20
*Metamodeling* *22*
*Compositional Metamodeling* *23*
*Semantic Foundations for Heterogeneous Systems* *23*
*Causality analysis of dataflow components for deadlock* *24*
2.1.2.b. Extensions to Distributed Models of Embedded systems 24
*Compositional Theory of Heterogeneous Reactive Systems* *24*
2.1.2.c. Model Transformation 25
2.1.2.d. Real-Time Programming Models 27
*Advanced Tool Architectures* *27*
*Trading Latency for Composability* *28*
2.1.3.a. Syntax and Semantics 28
*Modularity Mechanisms in Actor-Oriented Design* *28*

*Code Generation from Actor-Oriented Models* *29*
*Agent Algebra Theory for Platform-Based Design* *29*
*Synthesis for Platform-Based Design* *30*
*Metropolis Framework* *30*
*Reviews on Tools and Methodologies* *32*
2.1.3.b. Interface Theories 33
*Counting Interface Automata* *33*
*A Component Model for Heterogeneous Systems* *33*
2.1.3.c. Virtual Machine Architectures 33
*Types for Real-Time Programs* *33*
2.1.3.d. Components for Embedded Systems 34
*Mapping Network Applications to Multiprocessor Embedded Platforms* *34*
2.1.3.e Verification of Embedded Software 34
*Model Checking Quantitative Properties of Systems* *34*
*Run-Time Error Handling* *34*
*Memory Safety Enforcement in Assembly Code* *34*
2.1.4 Experimental Research 34
2.1.4.a. Embedded Control Systems 35
*Automated Landing for Unmanned Aerial Vehicles (UAVs)* *35*
*Pursuit Evasion Games* *35*
*Vision-based Landing of an Autonomous Rotorcraft* *35*
2.1.4.b. Embedded Software for National and Homeland Security 36
*Aerial Pursuit Evasion Games for Fixed-wing Aircraft* *36*
*Softwalls for Collision Avoidance* *36*
*Dirty Bomb Detection and Localization* *36*
2.1.4.c. Networks of Distributed Sensors 37
*VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems* *37*
*Viptos: a Programming Models for Sensor Networks* *38*
*Control of Communication Networks* *38*
2.1.4.d. Fault-Driven Applications 39
*Online Hierarchical Fault-Adaptive Control* *39*
*Development of engine models for combustion engine for controller synthesis* *42*
*Fault tolerant distributed systems* *42*
2.2. Project Findings 43
2.3. Project Training and Development 88
2.4. Outreach Activities 88
2.4.1. Curriculum Development for Modern Systems Science (MSS) 89
Undergrad Course Insertion and Transfer 89
*Course: Structure and Interpretation of Signals and Systems (UCB, EECS 20N) http://ptolemy.eecs.berkeley.edu/eecs20/* *90*
*Course: Bipedal Robotic Walking: From Theory to Practice (UCB, EECS Special Topics) http://chess.eecs.berkeley.edu/bipeds/* *90*
Graduate Courses 91
*Course: Concurrent Models of Computation for Embedded Software (UCB, EECS 290N) http://embedded.eecs.berkeley.edu/concurrency/* *91*

*Course: Hybrid Systems: Computation and Control (UCB, EECS 291E/ME 290S) http://robotics.eecs.berkeley.edu/~sastry/ee291e/HSCC05.htm* 91
*Course: Embedded System Design: Models, Validation, and Synthesis (UCB EE249)* 91
*Course: Foundations of Hybrid and Embedded Systems (VU, CS 376)* 92
*Course: Control Systems I (VU, EECE 257-01)* 92
*Course: Model Integrated Computing (VU, CS 388 / EE 395)* 92
*Course: Real-Time Systems (VU, EECE 353-01)* 92
*Course: Automated Verification (VU, EECE 315)* 93
*Course: Automated Verification (VU, EECE 375)* 93
2.4.2. SUPERB-IT Program 93
*Project: Visual Target Segmentation and Identification* 94
*Project: Modeling of Distributed Camera Networks* 94
*Project: Hybrid Reduction of a Bipedal Walker from Three to Two Dimensions* 95
*Project: A Hybrid Systems Approach to Communication Networks: Zeno Behavior and Guaranteed Simulations* 96
*Project: Modeling, Simulation, and Analysis of a Bipedal Walker* 96
*Project: Modeling and Analysis of On-Chip Networks* 97
Plans for 2006 97
*Project: Highway traffic flow analysis and control* 98
*Project: Tool for probabilistic safety verification of stochastic hybrid systems* 98
*Project: Autopilot for ultra-light flying wing* 98
*Project: Viptos: A graphical development and simulation environment for* 99
2.4.3. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program 99
*Project: Process Control Systems with Simulink/Stateflow* 101
*Project: Smart Structures* 101
*Project: Wireless Sensor Networks* 101
*Project: Autonomous Robot Path Planning and Mapping* 101
*Project: Embedded real-time operating systems* 101
Plans for 2006 102
3. Publications and Products 103
3.1. Journal Publications 103
3.2. Conference Papers 104
3.3. Books, Reports, and Other One-Time Publications 111
3.4. Dissemination 114
3.4.1. Software Maturation 114
Industry Technology Transition 114
3.4.2. Working Groups and Standards 115
3.4.3. "After Theory?" The 2005-2006 Chess seminar series 115
3.4.4. Workshops and Invited Talks 117
*Hybrid and Embedded Systems: Technologies and Applications* 117
*International Embedded Systems Forum* 118
*5th OOPSLA Workshop on Domain-Specific Modeling* 118
*ARTEMIS 2006 Annual Conference* 118
*OSD Workshops on the Software Producibility Initiative* 119
3.4.5. General Dissemination 119

3.5. Other Specific Product 119
4. Contributions 123
4.1. Within Discipline 123
4.1.1. Hybrid Systems Theory 123
4.1.2. Model-Based Design 124
4.1.3. Advanced Tool Architectures 125
4.1.4. Experimental Research 125
4.2. Other Disciplines 126
4.3. Human Resource Development 126
4.4. Integration of Research and Education 127
4.5. Beyond Science and Engineering 127

# 1. Participants

## 1.1. People

PRINCIPAL INVESTIGATORS:

THOMAS HENZINGER (UC BERKELEY, EECS)
EDWARD A. LEE (UC BERKELEY, EECS)
ALBERTO SANGIOVANNI-VINCENTELLI (UC BERKELEY, EECS)
SHANKAR SASTRY (UC BERKELEY, EECS)
JANOS SZTIPANOVITS (VANDERBILT, ECE)
CLAIRE TOMLIN (UC BERKELEY, EECS)

FACULTY INVESTIGATORS:

ALEX AIKEN (UC BERKELEY, CS)
RUZENA BAJCSY (UC BERKELEY, EECS)
GAUTAM BISWAS (VANDERBILT, CS)
RASTISLAV BODIK (UC BERKELEY, EECS)
BELLA BOLLOBAS (MEMPHIS, MATHEMATICS)
JEROME A. FELDMAN (UC BERKELEY)
KENNETH FRAMPTON (VANDERBILT, ME)
J. KARL HEDRICK (UC BERKELEY, ME)
GABOR KARSAI (VANDERBILT, ECE)
KURT KEUTZER (UC BERKELEY, EECS)
T. JOHN KOO (VANDERBILT)
WAGDY H. MAHMOUD (TENNESSEE TECH. UNIVERSITY)
GEORGE NECULA (UC BERKELEY, EECS)
SRINI RAMASWAMY (TENNESSEE TECH. UNIVERSITY)
PRAVIN VARAIYA (UC BERKELEY, EECS)
MASAYOSHI TOMIZUKA (UC BERKELEY, ME)

POST DOCTORAL RESEARCHERS:

MASSIMO FRANCESHETTI (UC BERKELEY)
CHRISTOPHER KIRSH (UC BERKELEY)
CLAUDIO PINELLO (UC BERKELEY)
MARCO SANVIDO (UC BERKELEY)
JONATHAN SPRINKLE (UC BERKELEY)

GRADUATE STUDENTS:

ALESSANDRO ABATE (UC BERKELEY)
AARON AMES (UC BERKELEY)

SAURABH AMIN (UC BERKELEY)
DANIEL BALASUBRAMANIAN (UC BERKELEY)
CHRIS BEERS (VANDERBILT)
ADAM CATALDO (UC BERKELEY)
ARINDAM CHAKRABARTI (UC BERKELEY)
DENNIS CHANG (UC BERKELEY)
KRISHNENDU CHATTERJEE (UC BERKELEY)
KAI CHEN (VANDERBILT UNIVERSITY)
ELAINE CHEONG (UC BERKELEY)
ABHIJIT DAVARE (UC BERKELEY)
DOUGLAS DENSMORE (UC BERKELEY)
ANDREW DIXON (VANDERBILT)
MATTHEW EMERSON (VANDERBILT)
HUINING FENG (UC BERKELEY)
SUMITRA GANESH (UC BERKELEY)
JOYTI GANDHE (VANDERBILT)
ARKEDEB GHOSAL (UC BERKELEY)
MATTHEW HARREN (UC BERKELEY)
GRAHAM HEMMINGWAY (VANDERBILT)
ETHAN JACKSON (VANDERBILT)
FARINAZ KOUSHANFAR (UC BERKELEY)
NARAYANAN KRISHNAN (UC BERKELEY)
BRANISLAV KUSHY (VANDERBILT)
MANISH KUSHWAHA (VANDERBILT)
ALEXANDER KURZHANSKIY (UC BERKELEY)
JONGHO LEE (UC BERKELEY)
XIAOJUN LIU (UC BERKELEY)
GABOR MADL (VANDERBILT)
DAVID P. MANDELIN (UC BERKELEY)
SLOBODAN MATIC (UC BERKELEY)
ELEFTHERIOS MATSIKOUDIS (UC BERKELEY)
MARK MCKELVIN (UC BERKELEY)
TREVOR MEYEROWITZ (UC BERKELEY)
BHARATHWAJ MUTHUSWARMY (UC BERKELEY)
TAKASHI NAGATA (UC BERKELEY)
STEPHEN NEUENDORFFER (UC BERKELEY)
SONGHWAI OH (UC BERKELEY)
ALESSANDRO PINTO (UC BERKELEY)
WILLIAM PLISHKER (UC BERKELEY)
VINAYAK PRABHU (UC BERKELEY)
KAUSHIK RAVINDRAN (UC BERKELEY)
JANOS SALLAI (VANDERBILT)
PANNAG SANKETI (UC BERKELEY)
PETER SCHMIDT (VANDERBILT)
TIVADAR SZEMETHY (VANDERBILT)
TAO TAO (VANDERBILT)

TODD TEMPLETON (UC BERKELEY)
GUOGIANG WANG (UC BERKELEY)
GUANG YANG (UC BERKELEY)
YANG YANG (UC BERKELEY)
JOSE CARLOS ZAVALA (UC BERKELEY)
HAIBO ZENG (UC BERKELEY)
YANG ZHAO (UC BERKELEY)
HAIYANG ZHENG (UC BERKELEY)
GANG ZHOU (UC BERKELEY)
YE ZHOU (UC BERKELEY)
QI ZHU (UC BERKELEY)

UNDERGRADUATE STUDENTS:

LANA CARNEL (UC BERKELEY)
MURPHY GANT (UC BERKELEY)
ROBERT GREGG (UC BERKELEY)
SHAMS KARIMKHAN (UC BERKELEY)
SIMON NG (UC BERKELEY)
REINALDO ROMERO (UC BERKELEY)
PHILLIP BALDWIN (UC BERKELEY)
CHRIS BEERS (VANDERBILT)
TREVOR BROWN (VANDERBILT)
COLIN COCHRAN (UC BERKELEY)
NICKOLIA COOMBS (VANDERBILT)
RACHAEL DENNISON (VANDERBILT)
BASIL ETEFIA (UC BERKELEY)
ELIZABETH FATUSIN (UC BERKELEY)
DAVID GARCIA (VANDERBILT)
RAFAEL GARCIA (UC BERKELEY)
SHANTEL HIGGINS (VANDERBILT)
MARY HILLIARD (VANDERBILT)
IYIBO JACK (UC BERKELEY)
JOHN KILBY (VANDERBILT)
SHIRLEY (XUE) LI (VANDERBILT)
PRAVEEN MUDINDI (VANDERBILT)
ANTONIO YORDAN-NONES (UC BERKELEY)
EFOSA OJOMO (VANDERBILT)
MIKE OKYERE (UC BERKELEY)
JAMESON PORTER (VANDERBILT)
RAKESH REDDY (UC BERKELEY)
MICHAEL RIVERA-JACKSON (VANDERBILT)
ISMAEL SARMIENTO (UC BERKELEY)
BINA SHAH (VANDERBILT)
SINITHRO TAVERAS (VANDERBILT)
EDWIN VARGAS (VANDERBILT)

TIRONE VINCENT (VANDERBILT)
JOHN WILLIAMSON (VANDERBILT)

TECHNICAL STAFF, PROGRAMMERS:

GYORGY BALOGH (VANDERBILT)
CHRISTOPHER HYLANDS BROOKS (UC BERKELEY)
NATHAN JEW (UC BERKELEY)
BRADLEY A. KREBS (UC BERKELEY)
PHILLIP LOARIE (UC BERKELEY)
ZOLTAN MOLNAR (VANDERBILT)
MARVIN MOTLEY (UC BERKELEY)
GUNNAR PROPPE (UC BERKELEY)
MARY STEWART (UC BERKELEY)
AARON WALBURG (UC BERKELEY)
BRIAN WILLIAMS (UC BERKELEY)

BUSINESS ADMINISTRATORS:

ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)
SUSAN GARDNER (UC BERKELEY)
TRACEY RICHARDS (UC BERKELEY)

## 1.2. Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY
VANDERBILT
MEMPHIS

## 1.3. Collaborators:

ROB ENNALS (INTEL RESEARCH, CAMBRIDGE, UK)
DAVID GAY (INTEL RESEARCH BERKELEY)
JUHA-PEKKA TOLVANEN (METACASE CONSULTING)
MATTI ROSSI (UNIVERSITY OF HELSINKI, ECONOMICS)
IAN M. MITCHELL (UNIVERSITY OF BRITISH COLUMBIA)
JAMES L. PAUNICKA (BOEING PHANTOM WORKS)
DAVID E. CORMAN (BOEING PHANTOM WORKS)
THOMAS RISGAARD HANSEN (UNIVERSITY OF AARHUS, CS)
ADAM DONLIN (XILINX RESEARCH)
SHINJIRO KAKITA (SONY CORPORATION)
JASON CONG (UNIVERSITY OF CALIFORNIA, LOS ANGELES)
INSEOK HWANG (PURDUE UNIVERSITY)
KAUSHIK ROY (STANFORD UNIVERSITY)

ASHISH TIWARI (SRI INTERNATIONAL, MENLO PARK, CA)
CAROLYN TALCOTT (SRI INTERNATIONAL, MENLO PARK, CA)
MARIA PRANDINI (POLITECNICO DI MILANO)
JOHN LYGEROS (UNIVERSITY OF PATRAS)
DR. DIRK BEYER (EPFL)
PROF. ORNA KUPFERMAN (HEBREW UNIVERSITY)
MARK WILCUTTS (TOYOTA MOTOR ENGINEERING & MANUFACTURING NORTH AMERICA)
TOMOYUKI KAGA (TOYOTA MOTOR ENGINEERING & MANUFACTURING NORTH AMERICA)
ANOUCK GIRARD (COLUMBIA UNIVERSITY, ME)
LUCA P. CARLONI (COLUMBIA UNIVERSITY)
ROBERTO PASSERONE (UNIVERSITY OF TRENTO, ITALY)
PAULO TABUADA (UNIVERSITY OF NOTRE DAME)
XIAOJUN LIU (SUN MICROSYSTEMS)
YOSINORI WATANABE (CADENCE)
JOHN MOONDANOS (INTEL)
SAMPADA SONALKAR (GENERAL MOTORS, INDIA)
XI CHEN (UC RIVERSIDE)
HARRY HSIEH (UC RIVERSIDE)
FELICE BALARIN (CADENCE BERKELEY LABORATORIES)
XI CHEN (NOVAS SOFTWARE)
JIE LIU (MICROSOFT RESEARCH)
FENG ZHAO (MICROSOFT RESEARCH)
JENS HARNISCH (INFINEON)
ETHAN JACKSON (VANDERBILT)
LUCA DE ALFARO (UC SANTA CRUZ)
MARCIN JURDZINSKI (UNIVERSITY OF WARWICK)
RANJIT JHALA (UC SAN DIEGO)
PROF. KLARA NAHRSTEDT (UIUC URBANA-CHAMPAIGN)
LISA WYMORE (UCB DANCE DEPARTMENT)
KATHERINE MEZURE (MILLS COLLEGE)
YUHONG XIONG (HP LABORATORIES)
LIZHI ZHONG (STMICROELECTRONICS)
PHILIP BALDWIN (2 MITCHELL, ECE)
CHRISTOPH KIRSCH (UNIVERSITY OF SALZBURG)
MIN XU (UNIVERSITY OF WISCONSIN)
MARK HILL (UNIVERSITY OF WISCONSIN)
DENIS GOPAN (UNIVERSITY OF WISCONSIN)
S. SASTRY (UNIVERSITY OF WISCONSIN)
JIM SMITH (UNIVERSITY OF WISCONSIN)
KEMAL EBCIOUGLU (IBM)
DOUG KIMELMAN (IBM)
RODRIC RABBAH (MIT)
CLAUDIO PINELLO (GENERAL MOTORS)
SRI KANAJAN (GENERAL MOTORS)

JOE WYSOCKI (HRL)
ALBERT BENVENISTE (IRISA)
BENOIT CAILLAUD (IRISA)
PAUL CASPI (VERIMAG)
HERMANN KOPETZ (TECHNICAL UNIVERSITY OF VIENNA, AUSTRIA)
MANFRED MORARI (ETH, ZURICH, SWITZERLAND)
GABOR PECELI (TECHNICAL UNIVERSITY OF BUDAPEST, HUNGARY)
JOSEPH SIFAKIS (CNRS VERIMAG, GRENOBLE, FRANCE)
KIN LARSEN (UNIVERSITY OF AALBORG, AALBORG, DENMARK)
HENRIK CHRISTENSEN (ROYAL INSTITUTE OF TECHNOLOGY, STOCKHOLM, SWEDEN)

# 2. Activities and Findings

## 2.1. Project Activities

This is the fourth Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software." This year generally saw a great deal of synergy among various researchers. This research activity is primarily organized through CHESS at the University of California, Berkeley (Center for Hybrid and Embedded Systems and Software, http://chess.eecs.berkeley.edu ), ISIS at Vanderbilt University (Institute for Software Integrated Systems, http://www.isis.vanderbilt.edu), and the Department of Mathematical Sciences, (http://msci.memphis.edu) at the University of Memphis.

The web address for the overall ITR project is:

http://chess.eecs.berkeley.edu/projects/ITR/main.htm

This web site has links to the proposal and statement of work for the project.

Main events for the ITR project in its fourth year were:

- NSF Onsite Review, November 21, 2005, UC Berkeley. The program and the presentations are available at http://chess.eecs.berkeley.edu/conferences/05/NovReview/
- The Berkeley Electrical Engineering Annual Research Symposium (BEARS) featured an open house co-sponsored by Chess in order to display results for the benefit of our industrial partners and friends of the project. The program and presentations are available at http://chess.eecs.berkeley.edu/conferences/06/BEARS/
- A weekly Chess workshop was held at Berkeley. The speakers and topics are listed in Section 3.4.3, and presentations for the workshop are available at http://chess.eecs.berkeley.edu/seminar.htm

We organize this section by thrust areas that we established in the statement of work.

### 2.1.1. Hybrid Systems Theory

We have proposed to build the theory of mixed discrete and continuous hybrid systems into a mathematical foundation of embedded software systems. For this purpose we have been pursuing four directions:

1. We have been designing models of computation that permit the composition of non-functional properties. While previously we had focused on real-time and resource-constrained systems, in the past year we developed a general theory of composing quantitative aspects of systems. We also formalized composition as a non-zero-sum game where the players (components) have different objectives.

2. We have been designing robust models of computation, where small perturbations of the system description cause only small changes in the system behavior. Previously we had

identified discounting as a paradigm for achieving robustness in discrete and hybrid models, and in the past year we developed model checking algorithms for discounted properties. We also studied affine hybrid systems as an approach to robust modeling. We have also been studying different functorial representations of hybrid systems with a view to characterizing Zeno properties of hybrid systems.

3. We have been developing and evaluating several methods for the computational treatment of hybrid systems. In particular, we have matured our design and implementation of a deterministic operational semantics for the simulation of hybrid systems, as well as ellipsoid-based algorithms for the efficient reach-set analysis of hybrid systems.

4. We have been developing stochastic models that combine hybrid dynamics with sources of uncertainty. For controlling such stochastic systems, we improved the best known algorithms for solving stochastic games. We also pursued the application of stochastic hybrid models in systems biology and for other classes of small noise perturbation of deterministic hybrid systems.

### 2.1.1.a. Deep Compositionality

***Trading Latency for Composability***

In many cases it is more important to understand the composition of the system than to have low-latency—particularly when composition properties need to be analyzed. When examining resource constraints, the periodic resource model for hierarchical, compositional scheduling abstracts task groups by resource requirements. We studied this model in the presence of dataflow constraints between the tasks within a group (intragroup dependencies), and between tasks in different groups (intergroup dependencies). We have examined two natural semantics for dataflow constraints, namely, RTW (Real-Time Workshop) semantics and LET (logical execution time) semantics. We show that while RTW semantics offers better end-to-end latency on the task group level, LET semantics allows tighter resource bounds in the abstraction hierarchy and therefore provides better composability properties. This result holds both for intragroup and intergroup dependencies, as well as for shared and for distributed resources. For more information, see where this work appeared at RTSS 05 [63] and RTAS 06 [36].

***Non-Zero-Sum Games as Compositional System Models***

In 2-player non-zero-sum games, Nash equilibria capture the options for rational behavior if each player attempts to maximize her payoff. In contrast to classical game theory, we consider lexicographic objectives: first, each player tries to maximize her own payoff, and then, the player tries to minimize the opponent's payoff. Such objectives arise naturally in the verification of systems with multiple components. There, instead of proving that each component satisfies its specification no matter how the other components behave, it sometimes suffices to prove that each component satisfies its specification provided that the other components satisfy their specifications. We say that a Nash equilibrium is secure if it is an equilibrium with respect to the lexicographic objectives of both players. We prove that in graph games with Borel winning conditions, which include the games that arise in verification, there may be several Nash equilibria, but there is always a unique maximal payoff profile of a secure equilibrium. We show how this equilibrium can be computed in the case of ω-regular winning conditions, and we

characterize the memory requirements of strategies that achieve the equilibrium. For more information, see our paper in [53].

***Hybrid Systems Modeling Tools: a Survey***
We completed the evaluation of a set of tools, languages and formalisms for the simulation, verification and specification of hybrid systems. In this survey we evaluated a number of languages and tools including: Simulink, Modelica, HyVisual, Scicos, Charon, CheckMate, Masaccio, SHIFT, HSIF, Metropolis and Hysdel. We described their syntax and semantics and we showed their modeling capabilities through simple examples. The goal of this survey is to identify, among all the languages, a potential interchange format for hybrid systems, or to define a new language which has all the necessary semantic and syntactic properties to describe hybrid systems. The survey results (more than 120 pages) will be published as the first issue of the NOW journal on embedded systems

***Graphs and games***
We are systematically filling gaps in the complexity theory of games played on graphs, which still contains several major open problems. These games are a model for the control of discrete event systems. In the past year we obtained several new results, especially about stochastic games and about concurrent games. Perhaps the strongest recent result is that in concurrent games with so-called parity objectives (a general form of omega-regular objectives), the probability of winning can be computed in the intersection of NP and co-NP (see [52]). This work appeared at TACAS 06 [41], STACS 06 [47][48], SODA 06 [52], FSTTCS 05 [62], ICALP 05 [96], and LICS 05 [101].

### 2.1.1.b. Robust Hybrid Systems

***Design and Verification of Robust System Models***
In previous work, we had identified discounting as a mechanism for moving from a discrete, brittle paradigm of boolean-valued property satisfaction to a continuous, robust paradigm of real-valued property estimation. In this past year, we developed algorithms for computing the real value of discounted properties expressed in temporal logic over state transition systems. As a next step, we have defined quantitative similarity functions between timed transition systems that measure the degree of closeness of two systems as a real, in contrast to the traditional boolean yes/no approach to timed simulation and language inclusion. Two systems are close if for each timed trace of one system, there exists a corresponding timed trace in the other system with the same sequence of events and closely corresponding event timings. We show that timed Computation Tree Logic (CTL) is robust with respect to our quantitative version of bisimilarity, in particular, if a system satisfies a formula, then every close system satisfies a close formula. We also define a discounted version of CTL over timed systems, which assigns to every CTL formula a real value that is obtained by discounting real time. We prove the robustness of discounted CTL by establishing that close states in the bisimilarity metric have close values for all discounted CTL formulas. This work is reported in [80].

***A Homology Theory for Hybrid Systems***
Using a categorical framework defined in previous work, we have developed a homology theory for hybrid systems. This theory enables us to characterize some intrinsic properties of the hybrid systems structure, include whether or not it admits of Zeno behavior [64]. In addition, we have examined the stability properties of a class of Zeno equilibria, and look toward a necessary

paradigm shift in the study of hybrid stability. Motivated by the peculiarities of Zeno equilibria, we consider a form of asymptotic stability that is global in the continuous state, but local in the discrete state. We provide sufficient conditions for stability of these equilibria, resulting in sufficient conditions for the existence of Zeno behavior. For more information, see [65].

### 2.1.1.c. Computational Hybrid Systems

***Algorithms for the Control of Stochastic Systems***
The problem of designing a controller can be viewed as the problem of finding a winning strategy of the control player in a game against the plant player. We improved on the best known algorithms for finding such strategies in the case that there is also a probabilistic source of uncertainty in the game. In [58], we introduce a method for approximating the dynamics of deterministic hybrid systems. Within this setting, we shall consider jump conditions that are characterized by spatial guards. After defining proper penalty functions along these deterministic guards, corresponding probabilistic intensities are introduced and the deterministic dynamics are approximated by the stochastic evolution of a continuous-time Markov process. We will illustrate how the definition of the stochastic barriers can avoid ill-posed events such as "grazing," and show how the probabilistic guards can be helpful in addressing the problem of event detection. Furthermore, this method represents a very general technique for handling Zeno phenomena; it provides a universal way to regularize a hybrid system. Simulations will show that the stochastic approximation of a hybrid system is accurate, while being able to handle "pathological cases." Finally, further generalizations of this approach are motivated and discussed.

***Reach Set Calculations using Ellipsoidal Approximations***
Ellipsoidal methods can be used to perform operations with ellipsoids and hyperplanes of arbitrary dimensions. It computes the external and internal ellipsoidal approximations of geometric (Minkowski) sums and differences of ellipsoids, intersections of ellipsoids and intersections of ellipsoids with halfspaces and polytopes; distances between ellipsoids, between ellipsoids and hyperplanes, between ellipsoids and polytopes; and projections onto given subspaces. Ellipsoidal methods are used to compute forward and backward reach sets of continuous- and discrete-time piecewise affine systems. Forward and backward reach sets can be also computed for continuous-time piece-wise linear systems with disturbances. It can be verified if computed reach sets intersect with given ellipsoids, hyperplanes, or polytopes. The toolbox provides efficient plotting routines for ellipsoids, hyperplanes and reach sets. More information on this work by Kurzhanskiy and Varaiya can be found at http://www.eecs.berkeley.edu/~akurzhan/ellipsoids/.

***A Deterministic Operational Semantics for Hybrid System Simulations***
In this continuation of work from the previous years, we have developed a deterministic operational semantics for hybrid system simulations. We have implemented this semantics in HyVisual, a domain-specific hybrid system modeling framework built on Ptolemy II. HyVisual includes a simulator that gives a well-defined execution by removing unnecessary non-deterministic behaviors when dealing with the discontinuities of piecewise continuous signals and when dealing with simultaneous discrete events.

We interpret the piecewise continuous signals as a set of continuous signals and discrete events, and the discontinuities as the effects of discrete events. In particular, we developed a mechanism

to accurately detect the time point when a state transition is enabled and force the transition to take place immediately. By this mechanism, an enabled transition is treated as a discrete event. Multiple discrete events can occur at the same time point in a model, but the semantics gives them a well-defined ordering. For example, when a transient state is entered, where incoming transitions and outgoing transitions are enabled simultaneously, two ordered discrete events with the same stamp represent the transition in and the transition out.

Based on this signal interpretation, a simulation of hybrid system models is divided into two kinds of interleaved execution phases: a continuous phase and a discrete phase. Each phase uses its own fix-point semantics to find the model's behavior. In particular, the fix point of the discrete phase of execution is reached only when all events at the current time have been handled.

We resolved a few subtleties with this operational semantics, including ways to generate piecewise continuous signals, operations on continuous-time signals with discontinuities such as sampling and level-crossing detection, and execution of transitions between transient states.

We have developed a formal model of this operational semantics, studying its relationship with those of synchronous languages and discrete-event languages, and unifying these into a general operational semantics for executing heterogeneous models. This work is available in [51].

***Building Efficient Simulations from Hybrid Bond Graph Models***

Embedded systems and their corresponding hybrid models are pervasive in engineering applications; therefore, systematic mathematical analysis using these models has become an important research area. Our approach to hybrid modeling with Hybrid Bond Graphs (HBGs) allows for seamless integration of physical system principles with discrete computational structures, but simulating the hybrid behaviors can be difficult and computationally expensive. We have developed an efficient method for creating block diagram models from HBG structures, where run time changes in model configuration are handled by reconfiguring the data flow through the blocks of the model.

A particularly intuitive physics-based modeling paradigm is the Bond Graph (BG) language. BGs provide a uniform lumped parameter, energy-based topological framework across multiple physical domains (e.g., electrical, fluid, mechanical, and thermal). HBGs extend BGs by incorporating local switching functions that enable the reconfiguration of energy flow paths in the model. This allows for seamless integration of energetic modeling and model reconfiguration to handle hybrid behaviors. Discrete changes in HBG model configuration are handled by junctions switching on and off. A Finite State Machine (FSM) implements the junction control specification. When the controlled junction is on, it behaves like a conventional junction. In the off state, all bonds incident on the junction are de-activated by enforcing a 0 effort or flow at the junction. The system mode at any time is determined by a parallel composition of modes of the individual switched junctions.

The inherent causal structure in BG models provides the basis for efficient conversion of BGs to computation models. For HBGs, the computation model is more complex because junction switching during behavior generation results in dynamic updating of the causal assignments and the computational structure during execution. We introduce the notion of a determining bond associated with every active (i.e., on) HBG junction. By definition, every active 0- (1-) junction in a valid bond graph will have one bond that determines the value of the effort (flow) for that junction. We label that bond as the determining bond for the particular junction. All other bond

effort (flow) values are dependent and set equal to this effort (flow) value. Similarly, the flow (effort) value on a determining effort (flow) bond is an algebraic sum of the flow (effort) values of the other bonds that are connected to this 0- (1-) junction. The determining bond plays a crucial role in mapping a HBG to a block diagram structure.

Converting a BG model to a block diagram is a straightforward procedure when there are no algebraic loops and no elements are in derivative causality. First, each bond is replaced by two links, i.e., the effort and flow variables for the bond. Next, each junction is replaced by the algebraic constraints they impose. The individual blocks for the other elements are now connected using the algebraic constraints imposed by the junctions. The choice of block depends on the assigned causality. The 1-1 mapping from bond graph elements to corresponding block diagram fragment can be found in most bond graph texts. The determining bonds establish the independent effort (flow) variables and the form of the algebraic equation for the corresponding flow (effort) variables at 0- (1-junctions). For HBGs, the block diagram structure must handle junction switching. This is realized functionality as a control flow graph that dynamically reconfigures the computational block diagram when mode switches occur. If derivative causality was allowed, additional computational structures would be needed to update the system state at mode transitions.

Given a HBG with n switching junctions, there are potentially 2n unique possible switching junction configurations. Pre-enumeration of all block diagram configurations offline and then selecting the appropriate one at run-time when junction configurations change is exponential in the number of switching junctions, and clearly a waste of space. On-line construction of the complete block diagram after each junction switch is space-efficient but wasteful in terms of computation time. Our solution to this problem is to construct a structurally adaptable block diagram model, and update the data flow paths through this model to match the causal structure when a junction switch occurs. Since we make the assumption that the system remains in integral causality, we exploit the locality principle for the propagation of causality changes through the model. This scheme may be combined with a caching mechanism that avoids having to recalculate causal assignment updates for discrete modes that have occurred previously.

When junction switches occur in a HBG model the following changes are made to the existing block diagram to generate the block diagram for the new mode.

1. Update the active HBG structure based on the junctions that change state.
2. Evaluate the changes in the determining bonds for the junctions in the HBG structure, and propagate these changes to derive the block diagram structure for the new mode.

For this work we implement the block diagram simulation models using Simulink. The Simulink environment provides all the primitives to implement the block diagram structure for a bond graph, and the bond graph elements. For hybrid junctions, we must implement the control structure as well as the dataflow structure. Rather than implementing a switching junction using discrete Simulink blocks, or using Stateflow extensions to Simulink, we implement the switching junction as custom written S-functions in C/C++. The S-function implements the dataflow machinery for the junction, as well as the evaluation of the control specification for the junction. For each bond connected to the junction, the S-function adds an input/output signal pair. The mapping of these signals to the effort/flow variables is determined dynamically. Note that we rely directly on the capabilities of the Simulink environment to detect the zero crossings, which

define the mode changes. We have successfully tested this approach for developing a number of complex models for Advanced Life Support system applications.

In summary, we use physical system modeling semantics as defined by BGs and HBGs to impose semantic structure on hybrid computational models in Simulink. This builds upon foundations by other elegant computational approaches, such as Ptolemy and HyVisual possess these semantics in a mathematical framework, although they do not link these semantics to physical system principles. Therefore, we believe that our approach for building computational models from HBGs provides a comprehensive framework for starting from component-oriented physical system models and deriving efficient computational models for hybrid systems.

***Going Beyond Zeno***

We developed a technique to extend the simulation of a Zeno hybrid system beyond its Zeno time point. [112][113]. A Zeno hybrid system model is a hybrid system with an execution that takes an infinite number of discrete transitions during a finite time interval. We argue that the presence of Zeno behavior indicates that the hybrid system model is incomplete by considering some classical Zeno models that incompletely describe the dynamics of the system being modeled.

This motivates the systematic development of a method for completing hybrid system models through the introduction of new post-Zeno states, where the completed hybrid system transitions to these post-Zeno states at the Zeno time point. In practice, simulating a Zeno hybrid system is challenging in that simulation effectively halts near the Zeno time point. Moreover, due to unavoidable numerical errors, it is not practical to exactly simulate a Zeno hybrid system. Our method for constructing approximations of Zeno models by leveraging the completed hybrid system model, and is foundationally based on the semantics developed in the deterministic hybrid systems model. Using these approximations, we can simulate a Zeno hybrid system model beyond its Zeno point and reveal the complete dynamics of the system being modeled.

### 2.1.1.d. Stochastic Hybrid Systems

Stochastic hybrid systems are a natural extension of the deterministic counterpart. We develop strategies for characterizing the stability of stochastic hybrid systems, and we use stochastic hybrid systems as approximations to simulation models of deterministic hybrid systems with non-deterministic switching.

***Stochastic Approximations of Hybrid Systems***

In this work [58] we consider jump conditions of deterministic hybrid systems that are characterized by spatial guards. After defining proper penalty functions along these deterministic guards, corresponding probabilistic intensities are introduced and the deterministic dynamics are approximated by the stochastic evolution of a continuous-time Markov process. We illustrated how the definition of the stochastic barriers can avoid ill-posed events such as “grazing,” and showed how the probabilistic guards can be helpful in addressing the problem of event detection. This method represents a very general technique for handling Zeno phenomena; it provides a universal way to regularize a hybrid system. We build on this work to have results about error bounds.

***Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems***

Building on the work in [58] we have explored an integration-inspired methodology for the simulation and analysis of deterministic hybrid dynamical systems. When simulating hybrid systems, and thus unavoidably introducing some numerical error, a progressive tracking of this error can be exploited to discern the properties of the system, i.e., it can be used to introduce a stochastic approximation of the original hybrid system, the simulation of which would give a more complete representation of the possible trajectories of the system. Moreover, the error can be controlled to check and even guarantee (in certain special cases) the robustness of simulated hybrid trajectories. For more information, see [57].

***Adjoint-based Optimal Control of the Expected Exit Time for Stochastic Hybrid Systems***

We have considered the problem of controlling the expected exit time from a region for a class of stochastic hybrid systems. That is, we find the least costly feedback control for a stochastic hybrid system that can keep its state inside a prescribed region for at least an expected amount of time. The stochastic hybrid systems considered are quite general: the continuous dynamics are governed by stochastic differential equations, and the discrete mode evolves according to a continuous time Markov chain. Instead of adopting the usual Hamilton-Jacobi viewpoint, we study the problem directly by formulating it as a PDE constrained optimization problem, and propose a solution using adjoint-based gradient descent methods. The adjoint method computes the gradient of an objective function whose variables are subject to PDE constraints. It is a powerful method, due mainly to the flexibility with which the optimal control problem can be formulated. Indeed, once the governing PDE, encoding the dynamics of the system, has been derived, many types of optimization problems can be posed. For instance, any constraints on the control input or on the state variable can be handled contrary to Hamilton-Jacobi formulations. Numerical results of the proposed approach are presented for several representative examples, and, for the simple case, compared with analytical results.

***Inference Methods for Autonomous Stochastic Linear Hybrid Systems***

The modeling of systems as stochastic hybrid systems has applications in fields such as target-tracking, the statistical analysis of time-series data, and systems biology. These systems frequently exhibit behavior that is a combination of discrete switches and continuous evolution; in addition, the data available in these applications is usually corrupted by noise. Most target-tracking algorithms for maneuvering targets, as well as estimators for hybrid systems, depend on the prior knowledge of a good model for the plant dynamics and noise characteristics, as well as knowledge of the transition probabilities between the discrete modes. We have designed a parameter inference algorithm for autonomous stochastic linear hybrid systems, which computes a maximum-likelihood model, given only a set of continuous output data of the system. We overcome the potentially intractable problem of identifying the sequence of discrete modes by using dynamic programming; we compute the maximum-likelihood continuous models using an Expectation-Maximization technique. This allows us to find a maximum-likelihood model in time that is polynomial in the number of discrete modes as well as in the length of the data series. We prove local convergence of the algorithm. We propose methods to derive good initial conditions, so that the local maximum converged to is a suitable model for tracking the future behavior of the system. We have demonstrated our algorithm on some examples - two simple one-dimensional examples with simulated data, and an application to real flight test data from a dual-vehicle demonstration of the DragonFly Unmanned Aerial Vehicles.

### 2.1.2. Model-Based Design

Model-based design uses models, which are formal, composable and manipulable during the design process. The modeling languages are domain-specific, offering designers modeling concepts and notations that are tailored to characteristics of their application domain. Domain-specific modeling languages (DSML-s) represent the structural and behavioral aspects of embedded software and systems. Their semantics capture concurrency, communication abstractions, temporal and other physical properties. For example, a DSML framework (i.e. a set of related modeling aspects) for embedded systems might represent physical processes using ordinary differential equations, signal processing using dataflow models, decision logic using finite-state machines, and resource management using synchronous models.

The project team started off with three different notions for embedded system and software design: platform-based design (developed by Sangiovanni-Vincentelli's group), actor-based design (investigated by Lee's group) and model-based design (advocated by Sztipanovits' group). These approaches emphasize different, complementary aspects of the design process. Platform-based design focuses on the creation of abstraction layers in the design flow and investigates the semantic properties of mapping across these layers. Actor-based design investigates component interaction semantics and theories of composition on different layers of abstractions. Model-based design focuses on the specification and composition of domain-specific modeling languages (DSML-s) and model transformations via metamodeling. As a result of interaction among the research groups a synergistic view is emerging:

- Abstractions and design constraints play central role in the definition of platforms. DSML-s offer a formal way for capturing these abstractions and constraints in metamodels. The required semantic clarity for expressing the mapping across platforms challenges the DSML technology with the need of expanding the abstract syntax oriented metamodeling toward explicit representation of formal semantics.
- A core concept in actor-based design is component interaction semantics defined by models of computation (MoC). New results have been achieved in the semantic foundations for heterogeneous systems, which will be the underpinning for the safe composition of heterogeneous reactive systems.
- Mapping across platforms has fundamental role in platform-based design flow. A new development in the technology of model transformations will contribute to the platform-based design vision.

Our extensive experimental work on networked embedded systems have revealed new challenges in extending model-based design to distributed models of embedded systems. We have reached significant progress in developing a new theory for the design of this system category. Some recent new work in this area has been in the area of semantic anchoring of models, see for example [7][8][9].

#### 2.1.2.a. Composition of Domain Specific Modeling Languages

In all approaches to model-based design, modeling languages play major roles that fall into the following three categories:

1. Unified (or universal) modeling languages, such as UML and Modelica, are designed with goals similar to programming languages: they are designed to be generic and to offer

the advantage to remain in a single language framework independently from the domain and system category the users are concerned with. Necessarily, the core language constructs are tailored more toward an underlying technology (e.g. object-oriented programming) rather then to a particular domain (even if extension mechanisms, such as UML profiling, allow some form of customizability).

2. Interchange languages, such as the Hybrid System Interchange Format (HSIF), are designed for facilitating the use of models across different analysis tools (hybrid system analysis). Accordingly, they are optimized to cover concepts related to an analysis technology.
3. Domain-specific modeling languages (DSMLs) are tailored to the particular concepts, constraints and assumptions of application domains. They are optimized to the requirements of a well defined domain: the modeling language should offer the simplest possible formulation that is still sufficient for the modeling tasks.

Model-based design frameworks that aggressively use DSMLs, need to support the composition of modeling languages. For example, the MIC infrastructure uses abstract syntax metamodeling and meta-programmable tool suites for the rapid construction of DSMLs with well defined syntax and semantics.

While abstract syntax metamodeling has been a very important step in model-based design and is used now not only in MIC but also in various MDA and MDD frameworks, such as Eclipse and Software Factories, explicit and formal specification of semantics has been an unsolved problem whose significance not even recognized. For instance, the SPT profile (UML Profile for Schedulability, Performance and Time) does not have precisely defined semantics, which creates possibility for semantic mismatch between design models and modeling languages of analysis tools. This is particularly problematic in safety critical real-time and embedded systems domain, where semantic ambiguities may produce conflicting results across different tools.

There has been much effort in the research community to define semantics of modeling languages by means of informal mathematical text or using formal mathematical notations. In either case, precise semantics specification for DSMLs remains a challenge. To solve this problem, we proposed a method and tool infrastructure for semantic anchoring that facilitates the transformational specification of DSML semantics. The proposed semantic anchoring infrastructure includes a set of well-defined "semantic units" that capture the operational semantics of basic dynamic behaviors and models of computations using Abstract State Machines as an underlying formal framework. The semantics of a DSML is defined by specifying the transformation between the abstract syntax metamodel of the DSML and that of the semantic unit.

During the last year we achieved the following progress in this research:

1. We have developed the first release of a semantic anchoring tool suite [8] that comprises (1) the ASM-based AsmL tool suite from Microsoft Research for specifying semantic units and (2) the MIC modeling (GME) and model transformation (GReAT) tool suites that support the specification of transformation between the DSML metamodels and the Abstract Data Models used in the semantic units.
2. We have demonstrated the use of the tool infrastructure in specifying the semantics of hierarchical state automata [9].

3. Using various specifications of timed automata, we have examined approaches for defining semantic units. We demonstrated the concepts with developing a semantic unit for timed automata [10] and showed the anchoring of UPAAL and IF to this common semantic unit.
4. We started investigating the problems of defining semantics for heterogeneous modeling languages. This investigation has led us to work on establishing a composition theory for semantic units. The first results of this work were applied to the compositional specification of semantics for a complex DSML proposed by industry. Results of the work are published in [11].

The semantic anchoring research direction focuses on behavioral semantics of modeling languages. We have also progressed in formalizing structural semantics, as well. Understanding structural semantics and investigating mathematical formalisms for their representation is very important in correct-by-construction design approaches, where a set of easily testable well-formedness rules guarantee selected system level properties. The first results of this work have been published in [7] and [12].

***Metamodeling***

The modeling languages in which models are expressed are domain-specific, offering embedded system designers modeling constructs and syntax that are closer to their application domain. Domain-specific modeling languages (DSMLs) must capture the structural and behavioral aspects of embedded software and systems. Their semantics must emphasize concurrency, communication abstractions, temporal and other physical properties. For example, a DSML framework (i.e. a set of related modeling aspects) for embedded systems might represent physical processes using ordinary differential equations, signal processing using dataflow models, decision logic using finite-state machines, and resource management using synchronous models. The languages that are used for defining components of DSMLs are called *meta-languages* and the formal specifications of DSMLs are called *metamodels*. The specification of the abstract syntax of DSMLs requires a meta-language that can express concepts, relationships, and integrity constraints. The specification of the semantic domain and semantic mapping is more complicated, because models might have different interesting interpretations; therefore DSMLs might have several semantic domains and semantic mappings associated with them. For example, the *structural semantics* of a modeling language describes the meaning of the models in terms of the structure of model instances: all of the possible sets of components and their relationships, which are consistent with the well-formedness rules in defined by the abstract syntax. Accordingly, the semantic domain for structural semantics is defined by a *set-valued semantics*. The *behavioral semantics* may describe the evolution of the state of the modeled artifact along some time model. Hence, the behavioral semantics is formally captured by a mathematical framework representing the appropriate form of dynamics.

The specification of the abstract syntax of DSMLs requires a meta-language that can express concepts, relationships, and integrity constraints. In our work in Model-Integrated Computing (MIC), we first adopted UML class diagrams and the Object Constraint Language (OCL) as meta-language. This selection was consistent with UML's four layer meta-modeling architecture, which uses UML class diagrams and OCL as meta-language for the abstract syntax specification of UML. Last year, we have developed a MOF-based metamodeling approach and developed a new metamodeling environment using our GME tool suite. This year, our work on MIC (see [2])

has helped to clarify technical details on the Model Driven Architecture (MDA) concept of OMG and we have started up a meaningful interaction with the OMG MDA community. Additional work that was performed this year is on domain specific visual languages for domain model evolution and on model "correctness" (see [4][7]).

***Compositional Metamodeling***

The GME-based metamodeling environment provides support for specifying DSML-s via metamodel composition. There are three characteristics of the GME that make it a valuable tool for the construction of domain-specific modeling environments. First, the GME provides generic modeling primitives that assist an environment designer in the specification of new graphical modeling environments. Second, these generic primitives are specialized to create the domain-specific modeling concepts through meta-modeling. The meta-models explicitly support composition enabling the creation of composite modeling languages supporting multiple paradigms. Third, several ideas from prototype-based programming languages have been integrated with the inherent model containment hierarchy, which gives the domain expert the ability to clone graphical models. Currently, we are exploring characteristics of the new MOF-based meta-modeling environment for DSML composition. See the work in [11][12] for the new results here.

***Semantic Foundations for Heterogeneous Systems***

We continued to work on our approach to the semantic foundations using an agent algebra framework. Agent Algebra is a formal framework that can be used to uniformly present and reason about the characteristics and the properties of the different models of computation used in a design, and about their relationships. This is accomplished by defining an algebra that consists of a set of denotations, called agents, for the elements of a model, and of the main operations that the model provides to compose and to manipulate agents. Different models of computation are constructed as distinct instances of the algebra. However, the framework takes advantage of the common algebraic structure to derive results that apply to all models in the framework, and to relate different models using structure-preserving maps.

Relationships between different models of computation are described as conservative approximations and their inverses. A conservative approximation consists of two abstractions that provide different views of an agent in the form of an over- and a under-approximation. When used in combination, the two mappings are capable of preserving refinement verification results from a more abstract to a more concrete model, with the guarantee of no false positives. Conservative approximations and their inverses are also used as a generic tool to construct a correspondence between two models. Because this correspondence makes the correlation between an abstraction and the corresponding refinement precise, conservative approximations are useful tools to study the interaction of agents that belong to heterogeneous models. A detailed comparison also reveals the necessary and sufficient conditions that must be satisfied for the well established notions of abstract interpretations and Galois connections (in fact, for a pair thereof) to form a conservative approximation. Conservative approximations are illustrated by several examples of formalization of models of computation of interest in the design of embedded systems.

While the framework of Agent Algebra is general enough to encompass a variety of models of computation, the common structure is sufficient to prove interesting results that apply to all models. In particular, we focus on the problem of characterizing the specification of a component of a system given the global specification for the system and the context surrounding the

component. This technique, called Local Specification Synthesis, can be applied to solve synthesis and optimization problems in a number of different application areas. The results include sufficient conditions to be met by the definitions of system composition and system refinement for constructing such characterizations. The local specification synthesis technique is also demonstrated through its application to the problem of protocol conversion.

***Causality analysis of dataflow components for deadlock***
Causality properties of components in a hybrid system model are important to ensuring that a unique behavior is defined by the model. By introducing a functional dependency, which describes the causality relationship between the inputs and outputs of a component, we have developed a mechanism to analyze the causality properties of a hybrid system model without flattening the hierarchies. These causality properties guide execution of a simulator, ensuring deterministic behavior for deterministic models. Because the causality properties of a hybrid system may change dynamically due to the state change, our mechanism supports a dynamic re-calculation of the causality properties.

Dynamic re-calculation of causality properties can be costly and not practical for some applications. We are developing a static analysis mechanism that infers the common causality properties of a modal model from those of its modes. The result of the static analysis is conservative, but provides safety guarantees. One of our objectives is to analyze the tradeoffs between the conservative static analysis and the more costly run-time analysis. This work is surveyed in the papers [25][33].

### 2.1.2.b. Extensions to Distributed Models of Embedded systems

***Compositional Theory of Heterogeneous Reactive Systems***
We have been working on a compositional theory of heterogeneous reactive systems in collaboration with A. Benveniste (INRIA), B. Caillaud (INRIA), and P. Caspi (VERIMAG). The approach is based on the concept of tags marking the events of the signals of a system. Tags can be used for multiple purposes from indexing evolution in time (time stamping) to expressing relations among signals like coordination (e.g., synchrony and asynchrony), and causal dependencies. The theory provides flexibility in system modeling because it can be used both as a unifying mathematical framework to relate heterogeneous models of computations and as a formal vehicle to implement complex systems by combining heterogeneous components.

In this period of the grant, we focused on the problem of choosing a distributed implementation architecture using this framework. Choosing a communication architecture that can be formally analyzed and/or guaranteed to maintain the ideal behavior of the system is an active research area and of great industrial interest. We follow in this work the Platform-based Design paradigm.

When the implementation platform includes one or more processors, functions are in general packaged into tasks which must be scheduled and/or distributed. When the resources of the architecture are limited, the distribution of the tasks to the architectural elements requires a careful scheduling and assignment step. For example, if two or more concurrent functions are assigned to the same sequential processing element, we need to determine an order according to which the functions must be executed. By the same token, if a number of communication requests are made to a limited interconnect structure such as a bus, an arbitration protocol determines the order with which the communication requests are served. In a realistic scenario,

architectural elements do take time to compute and to serve communication requests. Satisfying timeliness constraints requires clever assignment of functions to computing and communication elements. In addition, if the scheduling algorithm is not carefully designed, we may run into a dead-lock situation that would impact in a catastrophic manner system behavior.

Schedulability analysis, a very hard problem in the general case, aims at answering questions related to the correct behavior of the implementation when compared to the functional specification. Because of its conservative nature and of its computational complexity, engineers are used to performing approximate analysis but in doing so there is no guarantee that the final implementation would be always executing correctly, a very serious problem indeed for safety critical systems. Since the duration of tasks may vary depending on the execution platform characteristics, the functional semantics can be lost, unless rigid policies such as TTA or the one advocated by Giotto are used. An alternative approach to schedulability analysis as advocated by Kopetz with its Time-Triggered Architecture (TTA) is to use physical time to coordinate communication allowing the implementation of the real-time periodic synchronous model in a distributed way. Using this approach, correctness of a distributed implementation can be analyzed rigorously with formal techniques. However, this approach carries cost and timing penalties that at times are non acceptable for the application considered. For this reason, there has been growing interests in less constrained architectures such as the Loosely Time-Triggered Architecture (LTTA) used in the aerospace industry. All modern real-time distributed architectures share the viewpoint that communications should not be blocking. One way to achieve this is by triggering actions and communications by dates, thus resulting in what we call time-sensitive systems. Recent work has considered, for these architectures, the problem of maintaining proper functional semantics while performing task scheduling. Tracking how functional semantics may be skewed in this context requires a formal approach that captures causality dependencies and logical delays across the tasks of the functional specification as well as the resource availability and effective execution times that characterize its implementation on a given platform. Ultimately, this translates into the problem of guaranteeing that all the individual inter-task data exchanges occurring in the final implementation are consistent with those defined in the original specification. We addressed the hybrid nature of this problem using the framework of tag systems. We formally derived an operational protocol that guarantees the preservation of data semantics as we move from the specification to a particular implementation. This is accomplished through the insertion of a proper number of compensating logical delays in the inter-task communication channels. A subtle but important point is that to perform this operation correctly and optimally, we need to account for the possible presence of original logical delays in the specification. Furthermore, sometimes it may be necessary to revisit the original specification in order to correct it by increasing the “delay budget” between some tasks to match the constraint imposed by a given implementation platform. A practical contribution of this approach is to provide formal means to guide the designers through this process.

### 2.1.2.c. Model Transformation

In the model-based design area, the development of the model transformation toolsuite has continued. We have improved the model transformation language: GReAT with the following new capabilities:

- Global containers. During constructing complex model transformation programs we have observed that elements of the "state" of the transformation often had to be passed across many rules, without those rules contributing to the transformation on that part of the state space. This was caused by the purely functional nature of the transformation rules, and it was very inconvenient for the practical developer. Global containers solve the problem by introducing the equivalent of "global variables" where one can insert new objects and links in one rule, and fetch them in a later rule. While this feature violates the pure functional nature of transformation programs, it increases usability.
- Sorted objects. In some applications (e.g. a Stateflow/Simulink code generator) we have found that the result of the pattern matching or the products of transformation rule firing should be sorted according to some criteria. This sorting is difficult (sometimes impossible) to implement with existing GReAT constructs. We have extended the language with a sorting capability that is applied to the outputs of the transformation rules: when required, the results are sorted according to a user-supplied function.
- Distinguished cross-product. If there are multiple matches for $n > 2$ pattern elements then the pattern matcher generates the full Cartesian product of the matching elements. However, in some applications we needed that for every pattern element a specific host graph element appeared only once in the result. Now this capability is supported by the language and for every transformation rule such restrictions can be enforced.
- Match-any associations. In some applications we needed a simple way to test whether any association exists between two objects. The type of the association was irrelevant, only the existence mattered. We have added supported for this in the GReAT language and run-time system and code generator.

We have used the GReAT toolsuite during the year in developing model transformations for this ITR project, as well as for other sponsored research projects. In the context of this ITR, we have used GReAT to develop a model transformer that maps Giotto programs into E-code programs: this transformation program proved the usability of the language for non-trivial applications.

During the year, we have made several improvements to and releases of the GReAT and UDM packages. We have recently created a fully automated, nightly "build and test" framework that ensures software quality.

Another related effort focused on platform modeling and the connection between modeling languages and analysis tools. In earlier publications we have shown how platform models are relevant and could be used to generate analysis models from design models of embedded systems. Recently, this work has been extended and generalized to a new "platform modeling language" (PML) that allows the rapid construction of "design model -> analysis model" transformation tools. In this language, one has to explicitly include an extended metamodel for the platform that is assumed by the design modeling language. This extended metamodel captures how the "components" and the "run-time kernel" of the target run-time system look like in terms of concepts of language of the analysis tool. Additionally, the language supports a high-level specification of the mapping of design language structures into the "components" and the "run-time kernel". This mapping is specified using a higher-order, graph-transformation-based language that is simpler than GReAT. We have developed several examples for small-scale component-oriented, dataflow-driven design languages that were using an asynchronous

dataflow scheduler for component execution, and built “platform models” for the analysis of models using UPPAAL and IF. The results are documented in an upcoming PhD thesis.

### 2.1.2.d. Real-Time Programming Models

***Advanced Tool Architectures***

A premise of this project is that many foundational results are best expressed through software. Academic papers can gloss over scalability, practicality, and design issues, and frequently are much harder to understand than a software application that embodies the concepts. We view software as a publication medium that for some research results is more complete, more understandable, and more rigorous than papers. To maximize impact, we distribute source code, and we put considerable effort into making sure that code is readable. Moreover, our copyright policies encourage re-use of the code, even in commercial products, in order to maximize the probability of significant impact.

Institutionally, we have a long history of producing high-quality pioneering tools (such as Spice, Espresso, MIS, Ptolemy, Polis, and HyTech from UCB, and GME, SSAT, and ACE from Vanderbilt) to disseminate the results of our research. The conventional notion of “tool,” however, does not respond well to the challenges of deep compositionality, rapid construction and composition of DSMLs, and model-based transformation and generation. In this project, we have shifted the emphasis to tool architectures and tool components—that is, software modules that can be composed in flexible ways to enable researchers with modest resources to rapidly and (most importantly) correctly construct and experiment with sophisticated environments for hybrid and embedded systems.

We currently have three key frameworks that we use for this purpose, GME, which emphasizes metamodeling, Metropolis, which emphasizes codesign of architecture and functionality, and Ptolemy II, which emphasizes concurrent models of computation. The cores of these three frameworks predate this project. They are evolving together into a more coherent view of what frameworks and toolkits for hybrid and embedded software systems require.

All three frameworks share a focus on what we call "actor-oriented design," where components are conceptually concurrent and interaction between components is via the flow of data through ports. This contrasts with (and complements) prevailing object-oriented methods, where components bundle data with methods and interaction between components is through procedure calls (method invocations, which semantically involve a transfer of control). Many commercial tools, such as Simulink, Labview, Modelica, Opnet, VHDL, and many others, use actor-oriented componentization (and some, like Modelica, use it together with object-oriented componentization). Thus, our work has promise of building the fundamentals behind high-profile and high-impact tools. For example, one key innovation of the last year is the introduction of "classes" and "inheritance" in an actor-oriented sense to complement such mechanisms that have long existed in the object-oriented sense. We report on some of the work from [91][92][93].

Conceptual concurrence between our frameworks is evolving. The work in GME produced technology which mastered metamodeling of abstract syntactic properties of actor-oriented models, and today it is moving aggressively towards coupling this metamodeling to anchored semantic properties, focusing initially on the concurrent models of computation that have been implemented in Ptolemy II. Metropolis and Ptolemy II are both seeking to abstract these semantic properties in a somewhat different (and complementary) way than metamodeling. They

both define an "abstract semantics" that represents the common features of families of models of computation, and they both realize models of computation through specialization (concretization) of this abstract semantics. They do so, however, in different ways, and by pursuing these different ways, the team is gaining an understanding of the fundamentals behind the use of such abstract semantics in frameworks.

A number of more specialized tools (vs. frameworks) are also being built to illustrate, refine, and publish research results. For example, Chic supports definitions of interfaces and interface theories and provides compatibility checking with respect to these interface theories. We have demonstrated that Chic can be used as a component within the Ptolemy II framework, and are using this integration to try to identify which interface theories are most productive and useful to designers. NP-Click, which was first prototyped within Ptolemy II, explores models of computation that appear to be particularly well-suited to high-speed networking systems design. Giotto, which has both a stand-alone textual syntax and a graphical syntax built in Ptolemy II, elevates the semantic principles of time-triggered architectures to the programming language and modeling level. GReAT builds on GME's metamodeling of abstract syntax to synthesize model transformers that bridge distinct abstract syntaxes. Desert builds on GME to provide systematic exploration of families of designs where components have several distinct available implementations. Blast and CCured are focused on improving the reliability of the embedded C code that ultimately emerges from these tools. Streambit explores a model of computation for computation on streams of bits, such as that typically found in networking applications. As these tools evolve, the most useful ones will be integrated into one or more of our frameworks, providing the community with a coherent view of best practices methods.

***Trading Latency for Composability***
We have formally proved the benefits that the Giotto model offers in terms of composability over traditional real-time models. In particular, our comparisons of the LET (Giotto) semantics and RTW (Simulink) semantics, show that the former offers better composability properties (while the latter offers better latencies). This work also ties in to interface theory for real-time components. For more information, see where this work appeared at RTSS 05 [63]and RTAS 06 [36].

### 2.1.3.a. Syntax and Semantics

***Modularity Mechanisms in Actor-Oriented Design***
Concurrent, domain-specific languages such as Simulink, LabVIEW, Modelica, VHDL, SystemC, and OPNET are widely used in the design of embedded software. They provide modularization mechanisms that are significantly different from those in prevailing object-oriented languages such as C++ and Java. In these languages, components are concurrent objects that communicate via messaging, rather than abstract data structures that interact via procedure calls. Although the concurrency and communication semantics differ considerably between languages, they share enough common features that we consider them to be a family. Included in this family are our own hybrid systems modeling languages (like HyVisual) and embedded software design languages (like Giotto). We call them actor-oriented languages, and have been studying their properties as a family of languages.

***Code Generation from Actor-Oriented Models***

We have been developing technology for code generation from actor-oriented models in Ptolemy II. This code generation system, called *Copernicus*, is designed to make maximum use of the same generic actor specifications used for simulation. The system is based on the concept of component specialization: generic actor specifications are transformed according the model context in which they are used. This model context includes information such as the connections between actor ports, assignments of values to actor parameters, and the model of computation that governs component interaction. Each aspect of a component's context, which doesn't change presents an opportunity for specialization to improve the execution performance of the component.

Recently we have focused on analysis techniques for analyzing the reconfiguration of parameter values that goes beyond simply determining whether parameter value changes or not. The analysis determines a bound on how often during the execution of a model particular parameters are reconfigured. We interpret this analysis as a behavioral type system capable of checking constraints on reconfiguration. Although reconfigured parameters cannot be specialized during code generation, behavioral type constraints on reconfiguration can enable scheduling optimization of the interaction between components. During code generation, this optimization allows threads and dynamic communication buffers to be replaced with statically scheduled code and statically allocated communication buffers.

***Agent Algebra Theory for Platform-Based Design***

The motivation for developing this theory described in [44] is the hope to obtain an algorithm to perform the mapping process automatically and in an optimized fashion. An agent algebra represents a domain, a model of computation, that is a natural model to express a specification. This algebra is instrumental in describing the PBD design flow. The function to be implemented is described in a domain that we call the Function Domain that is chosen depending on the application to model. Notice that, while the agent algebras change depending on the application domain and on the level of abstraction, at each level and for each application the description that we give here is fixed. The same formalism can be used to represent platforms. To obtain an appropriate domain of agents to model a platform, we start from the set of library elements D0. The domain of agents D is then constructed as the closure of D0 under the operation of parallel composition: The set of agents D represents all possible legal compositions of the library elements. To map the function on a platform instance we define a common domain, a new agent algebra, where both specification and platform instance can be “refined”. Given a platform QP and specification domain QS, a common semantic domain is an agent algebra QC related to QP and QS through conservative approximations. In the common semantic domain both function and architecture instances are described using the same mathematical objects. A single function that is mapped from the function domain to the common domain defines a set of ordered agents representing all those agents that refine the same functionality. Similarly, a platform instance that is mapped from the architecture platform to the common domain defines a set of ordered agents that represent all those configuration of the platform with the same properties. The intersection of the two sets of agents in the common domain is a new set of agents each implementing the function on the platform instance. A synthesis, or automatic mapping, tool selects a platform instance in the architecture platform and its parameters to meet the functional requirements and minimize a user-defined cost function (power consumption, area or simply monetary cost). The result of a mapping is the selection of an agent in the common semantic domain. The selected agent turns into the function specification for the lower level of abstraction. Since both the

behavior and the architectural components of the platform are expressed in the common semantic domain, the automatic mapping tool can be seen as a generalized optimal covering problem where we seek to "cover" the agents representing the behavior of the design with the agents representing behaviors of the architectural components.

***Synthesis for Platform-Based Design***

The agent algebra approach to synthesis in PBD has to be complemented with a refinement process that takes the functionality that is assigned to a software programmable processing element and generates optimized code. When the processing element is a single processor that executes sequential code, the mathematical representation of the concurrent functionality has to be converted in a sequence of operations of the processor. We have developed a theory that takes a data-flow like representation of the functionality and maps it into a Petri net. Then the Petri net is manipulated and optimized to produce sequential code. The theory of schedulability on Petri nets has received the attention of the formal community for some time. We have developed a method that generates a finite sequential program even when the Petri net is proven to be unschedulable. We also have found sufficient structural conditions for Petri nets to be schedulable that are tighter than any other published method.

***Metropolis Framework***

The Metropolis framework is based on the unified representation of designs and of the design processes using a particular modeling approach that we called the Metropolis MetaModel (MMM). The term "metamodel" has been used in the OMG world and by Sztipanovits' group to indicate a particular abstract syntax; in the Metropolis domain "metamodel" refers to *abstract semantics*, i.e., semantics that can be used to express in principle all models of computation used in design. The central role of the MMM plays to its extensibility to a variety of design domains from system-on-chips to distributed systems such as automobiles and elevators.

The activities in this domain have four prongs:

- Continuous improvement of the performance and usability of the basic infrastructure (simulation, Metropolis metamodel editing and compilation);
- Addition of tools and methodologies for communication-based design;
- Moving towards Metropolis II, a new version of the framework that is intended to improve substantially the ease of use and the ease of integration of native and foreign tools by focusing on the coordination aspects of the modeling approach.
- Application of the framework, tools and PBD design methodology to a number of different industrial domains with the support of our industrial partners.

<u>Continuous improvement of the performance and usability of the basic infrastructure</u>

In the design of highly complex, heterogeneous, and concurrent systems, deadlock detection and resolution remains an important issue. In [103], we systematically analyzed the synchronization dependencies in concurrent systems modeled in Metropolis, where system functions, high level architectures and function-architecture mappings can be modeled and simulated. We proposed a data structure called the dynamic synchronization dependency graph, which captures runtime (blocking) dependencies. A loop-detection algorithm is then used to detect deadlocks and help designers quickly isolate and identify modeling errors that cause the deadlock problems. We demonstrated our approach through a real world design example, which is a complex functional model for video processing and a high level model of function-architecture mapping.

Addition of tools and methodologies for fault-tolerant and communication-based design

*Fault Tree Synthesis*. We refined a fault tree synthesis methodology for generating accurate fault trees and covering more fault events than informal methods. We implemented an automotive case study that extends multiple energy domains (i.e. electrical, mechanical) and integrates with system controller data-flow models.

*NOC Design.* We developed a methodology for performance improvement of network-based on-chip communication via long-range link customization. We proposed a deadlock-free routing scheme which works with application-specific long-range links. We also developed a FPGA-based prototype using a 4-by-4 NoC communication architecture and evaluated various power/performance tradeoffs.

Moving towards Metropolis II

To handle complexity, IC designers are moving towards higher abstraction levels, such as transaction level or behavioral level. However, the abstraction gap prohibits easy communication and synchronization in IP integration. Another challenge is the co-simulation among IPs written in different design languages. Up to now, there are attempts on co-simulation between HDLs and C/C++/SystemC; however, there does not exist a generic co-simulation framework for arbitrary design languages. In [45], a communication infrastructure for the new version of Metropolis, Metropolis II was presented that enables co-design and co-simulation of heterogeneous design components specified at different abstraction levels and in different languages. The core of the approach is abstracting different communication interfaces or protocols to a common high-level communication semantics. Designers only need to specify the interfaces of the design components using extended regular expressions; communication adapters can then be automatically generated for the co-simulation or other co-design and co-verification purposes. We plan to build upon this work to develop Metropolis II.

Applications

*Sensor Network Platform* (Pirelli and Telecom Italia). We defined the Sensor Network Ad-hoc Protocol Platform (SNAPP), intended to complete the mapping of applications into hardware nodes for control systems based on Wireless Sensor networks. Proof of concept and test bed development of the overall synthesis flow was carried out for a periodic control application in an industrial environment

*Automotive Architecture Exploration* (General Motors). Automotive control applications are implemented over distributed platforms consisting of a number of electronic control units (ECUs) connected by communication buses. During system development, the designer can explore a number of design alternatives: for example, software distribution, software architecture, hardware architecture, and network configuration. Exploring design alternatives efficiently and evaluating them to optimize metrics such as cost, time, resource utilization, and reliability provides an important competitive advantage to OEMs and helps minimize integration risks. We presented how [38] a methodology (Platform-Based Design) and a framework (Metropolis) can be used to support efficient architecture exploration. We exercised the methodology and the capabilities of Metropolis for developing a library of automotive architecture components and performed design space exploration on a chassis control sub-system.

*Multimedia Application Specification and Multiprocessor Model* (Infineon and Intel). In [81], we applied the Platform-based design (PBD) methodology to tackle design issues for multimedia

design by recommending the use of formal models, carefully defined abstraction layers and the separation of concerns. Models of computation (MoCs) can be used within this methodology to enable specialized synthesis and verification techniques. In this work, these concepts are leveraged in an industrial case study: the JPEG encoder application deployed on the Intel MXP5800 imaging processor. The modeling was carried out in the Metropolis design framework. We showed that the system-level model using our chosen model of computation allows performance estimation within 5% of the actual implementation. Moreover, the chosen MoC is amenable to automation, which enables future synthesis techniques.

*FPGA Architecture Characterization* (Xilinx). We presented in [39] a modular and scalable approach for automatically extracting actual performance information from a set of FPGA-based architecture topologies. This information is used dynamically during simulation to support performance analysis in a System Level Design environment. The topologies capture systems representing common designs using FPGA technologies of interest. Their characterization is done only once; the results are then used during simulation of actual systems being explored by the designer. Our approach allows a rich set of FPGA architectures to be explored accurately at various abstraction levels to seek optimized solutions with minimal effort by the designer. To offer an industrial example of our results, we describe the characterization process for Xilinx CoreConnect-based platforms and the integration of this data into the Metropolis modeling environment.

*Functional Model exploration for Multi-media Applications* (Sony). An optimized functional design space exploration method for multimedia applications was proposed in previous work. The basis of the method is a way of representing the dependency and the concurrency of an application in a compact form exploiting algebraic operators and expressions. The optimized design process consists of mapping one of the possible expressions in the application space onto a concurrent architecture. We used the Metropolis design framework to demonstrate the effectiveness of the procedure using an FPGA architecture as the target implementation platform. The advantage of using this platform is the availability of models that approximate well the performance of the final implementation when performing the mapping from function to architecture thus yielding a robust design methodology.

***Reviews on Tools and Methodologies***

In addition to the technical work presented above, we also spent time reviewing and categorizing the available methodologies and tools in system design. The research findings were used to place the literature and the tools in a unified framework.

*Electronic System Level Tools: A Platform-Based Taxonomy* [43]. At this relatively early stage in the development of Electronic System Level Design, there is no agreed upon definition of what the space is let alone what a design flow and tools should include. This research is an attempt at defining a conceptual framework for ESL design (platform-based design) where more than 90 present and future offerings can be uniformly described and analyzed.

*Hybrid Systems Modeling Tools: a Survey*. We completed the evaluation of a set of tools, languages and formalisms for the simulation, verification and specification of hybrid systems. In this survey we evaluated a number of languages and tools including: Simulink, Modelica, HyVisual, Scicos, Charon, CheckMate, Masaccio, SHIFT, HSIF, Metropolis and Hysdel. We described their syntax and semantics and we showed their modeling capabilities through simple examples. The goal of this survey is to identify, among all the languages, a potential interchange

format for hybrid systems, or to define a new language which has all the necessary semantic and syntactic properties to describe hybrid systems. The survey results (more than 120 pages) will be published as the first issue of the NOW journal on embedded systems.

### 2.1.3.b. Interface Theories

***Counting Interface Automata***
We present an interface theory based approach to static analysis of actor models. We first introduce a new interface theory, which is based on Interface Automata, and which is capable of counting with numbers. Using this new interface theory, we can capture temporal and quantitative aspects of an actor interface as well as an actor's token exchange rate. We will show how to extract this information from actors written in the Cal Actor Language (Cal), and we also present a method to capture the interface information as well as the structure of dataflow models into an interface automaton. This automaton acts as glue between the automata of all actors in the model, and by successfully composing all actor automata with it, we can prove interface compatibility of all actors with the composition framework. After successful composition, the resulting automaton will contain information that can be used for further static analysis of the composite actor model.

***A Component Model for Heterogeneous Systems***
We have developed a new component model for timed models of computation such as discrete event, continuous time, hybrid systems, and synchronous/reactive models. Using the tagged signal model (developed by Lee and Sangiovanni-Vincentelli) as the basis to analyze the computational requirements of these models of computation, we developed a unified scheme to simulate heterogeneous timed models. The scheme relies on the proposed component model that aims to minimize the interface complexity between components and their operating environment. A generic component in this model is similar to a Mealy state machine, having an output function that computes the input-output relation at the current simulation time, and a next state function that computes the new state of the component. The simulation of a timed model goes through a sequence of time steps. In each step the system of equations formed by the components in the model is solved. This unified scheme provides a solid foundation for building correct simulators of heterogeneous timed models. Extensions to the generic component model are developed to satisfy the requirements of specific models of computation, while still keeping the interface complexity of the components minimal. A small component interface makes component composition easier and more flexible.

### 2.1.3.c. Virtual Machine Architectures

***Types for Real-Time Programs***
We developed a type system for Embedded Machine code, which is assembly-like hard real-time code, with the property that well-typed programs are efficiently schedulable. Work is ongoing on the use of embedded machine code for time triggered controllers for UAVs.

### 2.1.3.d. Components for Embedded Systems

***Mapping Network Applications to Multiprocessor Embedded Platforms***
We formulated and solved the task allocation problem for a popular multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor. This method proves to be computationally efficient and produces results that are within 5% of aggregate egress bandwidths achieved by hand-tuned implementations on two representative applications: IPv4 Forwarding and Differentiated Services. We are currently exploring extensions to this work by considering multiple target platforms: a reconfigurable multiprocessor system on the Xilinx Virtex-II Pro and the IXP2400. The results are reported in the papers [82][87].

### 2.1.3.e Verification of Embedded Software

***Model Checking Quantitative Properties of Systems***
We developed model checking algorithms for automata whose states are not labeled with boolean-valued propositions, but with natural-number valued quantities. These numbers might express, for example, power consumption or memory usage. More information is available in the publication at [73][80].

***Run-Time Error Handling***
It is difficult to write programs that behave correctly in the presence of run-time errors. Existing programming language features often provide poor support for executing clean-up ode and for restoring invariants in such exceptional situations. e present a data flow analysis for finding a certain class of error-handling mistakes: those that arise from a failure to release resources or to clean up properly along all paths. Many real-world programs violate such resource safety policies because of incorrect error handling. Our flow-sensitive analysis keeps track of outstanding obligations along program paths and does a precise modeling of control flow in the presence of exceptions. Using it, we have found over 800 error handling mistakes almost 4 million lines of Java code. Among the systems that we have debugged with our tool is the Ptolemy software.

***Memory Safety Enforcement in Assembly Code***
There are many source-level analyses or instrumentation tools that enforce various safety properties. In this paper we present an infrastructure that can be used to check independently that the assembly output of such tools has the desired safety properties. By working at assembly level we avoid the complications with unavailability of source code, with source-level parsing, and we certify the code that is actually deployed. The novel feature of the framework is an extensible dependently-typed framework that supports type inference and mutation of dependent values in memory. The type system can be extended with new types as needed for the source-level tool that is certified. Using these dependent types, we are able to express the invariants enforced by CCured, a source-level instrumentation tool that guarantees type safety in legacy C programs. We can therefore check that the x86 assembly code resulting from compilation with CCured is in fact type-safe [76].

### 2.1.4 Experimental Research

The main emphasis of our research is on the foundations of hybrid systems theory and of embedded system design. However, in the best tradition of our groups, a strong application

program is necessary to verify the viability of the theory and to uncover difficult problems that provide appropriate motivation to develop new methods and theories. Most of the applications studied are distributed systems where scarce and fragile resources have to be used to provide reliable behavior. Wireless sensor networks, distributed systems for automotive electronics, embedded systems for national and homeland defense, are but a few examples that attracted the attention of our research groups because of their complexity and of their objective importance. We argue that the distributed nature of the applications poses additional challenges to overcome with an appropriate design methodology and supporting tools.

In particular, during this period, we have focused on the application to UAVs for air borne combat, Unmanned Underwater Vehicles, wireless sensor networks to control and monitoring, on fault-diagnosis, fault-adaptive and fault-tolerant approaches for distributed systems, and finally, on a multi-media problem as a test vehicle for the methodology and the tools embedded in Metropolis.

### 2.1.4.a. Embedded Control Systems

***Automated Landing for Unmanned Aerial Vehicles (UAVs)***
In work using hybrid control, we have partnered with Northrop Grumman to devise and actually fly (on a surrogate UAV, a Boeing T-33 with avionics to mimic the Unmanned Combat Air Vehicle (UCAV)) algorithms for automated landing of UAVs with special contingency maneuvers for waving off landing. This work is reported in [100], and in [75] a more detailed vision of how to bring about these kinds of applications about. This work depends heavily on research performed in reachable sets and embedded controller synthesis, and is being transitioned to the Automated Aerial Refueling (AAR) effort through the Certification Techniches for Flight Critical Systems (CerTA FCS) project through AFRL.

***Pursuit Evasion Games***
We have presented final simulation and flight test results for a Non-linear Model Predictive Controller (NMPC) used in evasive maneuvers in three dimensions on a fixed wing UAV for the purposes of pursuit/evasion games with a piloted F-15 aircraft. Such capabilities have been under development through Software Enabled Control (SEC) program and were recently tested in the Capstone Demonstration of that program. This work in [99] is critical to showing how layered control and embedded software can lead to more widespread use of intelligent vehicles. Through foundational support from the ITR we were able to show rapid results though the development cycle was on the order of months.

***Vision-based Landing of an Autonomous Rotorcraft***
We successfully demonstrated the use of computer vision to scope landing zones for an autonomous rotorcraft using a single camera, and with control of the landing in the loop. This was shown on a Robinson R22 helicopter (a surrogate for the Boeing Hummingbird A160) which is a human-carrying helicopter. Ours was the first university-developed landing controller to do this, according to our Boeing sponsors. The work was done in conjunction with the DARPA SEC Program.

### 2.1.4.b. Embedded Software for National and Homeland Security

***Aerial Pursuit Evasion Games for Fixed-wing Aircraft***
We have presented final simulation and flight test results for a Non-linear Model Predictive Controller (NMPC) used in evasive maneuvers in three dimensions on a fixed wing UAV for the purposes of pursuit/evasion games with a piloted F-15 aircraft. Such capabilities have been under development through Software Enabled Control (SEC) program and were recently tested in the Capstone Demonstration of that program. There were several instances of our controller defeating the skilled test pilot which led him to quip that the UAVs was just as good as a human pilot. This work in [99] is critical to showing how layered control and embedded software can lead to more widespread use of intelligent vehicles. Through foundational support from the ITR we were able to show rapid results though the development cycle was on the order of months.

***Softwalls for Collision Avoidance***
In the last year, we have studied several methods for the Soft Walls controller, looking for a method which will work for a more realistic method of the aircraft. We have looked at a discrete-time formulation of the game theory formulation. Unfortunately, the first theoretical result did not lend itself to a practical, computable algorithm. With Claire Tomlin of Stanford and George Pappas of the University of Pennsylvania, we have begun to investigate collision avoidance methods based on computational geometry methods. Some intriguing connections are to be made between model predictive control and the solution of Hamilton Jacobi equations.

***Dirty Bomb Detection and Localization***
We have showcased several exciting technologies in an integrated demonstration. The demo scenario is as follows: a plain cloth security guard walks around the stadium with a cell phone-integrated radiation detector. The person also carries an XBow XSM mote that we continuously track using an enhanced version of the radio interferometric positioning technique introduced at the ACM SenSys conference in 2005. There are 12 XSMs acting as infrastructure nodes deployed at known positions to enable tracking with ~1m accuracy throughout the stadium. When the radiation detector is in close proximity to a source, it sends an alarm using the mobile phone network. This causes a remote controlled camera to automatically zoom in on the position of the policeman. The large Jumbotron display in the stadium shows the video as well as the Google Earth-based user interface.

The tracking is done using a novel radio interferometric technique developed previously under the DARPA NEST program. The tracked node selects a neighbor and the pair acts as transmitters. They emit radio waves in the 400MHz band that have ~350Hz separation. All other infrastructure nodes measure the phase of the low frequency envelope signal that is the result of the two signals interfering. The relative phase offset of pairs of nodes provides information on the relative location of the four nodes involved. It is possible to determine the position of one node if at least four other nodes at known positions participate in the measurement. Additional nodes are used to provide better coverage of the large area and to compensate for errors introduced by RF multipath effects.

The security component (MultiMAC) provides group-based peer authentication for sensor nodes. We use the SkipJack implementation in TinySec as symmetric cipher. Each sensor stores a different set of keys in its ROM which is pre-defined by a key mapping scheme. Multiple message authentication code (MAC)s of every message are calculated in SkipJack, using the key

set assigned to the sensor node. The receiver authenticates the message by recomputing MACs using its common keys with the sender.

The radiation detector is a small, battery-powered gamma detector connected to a mobile phone running a Java application reading the detector output. The output is sent nearly continuously over the phone's mobile data network to a server which checks the received gamma count for a threshold crossing. If an elevated gamma reading is seen, the server interfaces with the XSM system to determine the current location of the policeman and sends a control message to the camera system.

The camera system consists of a controller unit communicating with one or more cameras via a wireless network using the IEEE 1451 protocol. The system can be used to point the camera(s) on demand, based on alert conditions. The controller unit accepts position commands to slew, zoom, and focus a camera on a specific location. The locations are specified as coordinates (in any well-defined coordinate reference system) as well as "field of view," which determines the zooming degree of interest. Additionally, each camera's pan, tilt and zoom capabilities can be directly accessed and controlled over the network.

This integrated demonstration showcases important technologies and potential homeland security applications of sensor networks:

- highly accurate positioning and tracking using wireless sensor networks (ISIS-VU),
- sophisticated radiation detection capabilities (ORNL),
- secure sensor network architecture (ISIS-VU / TRUST),
- early example of the application of federated sensor networks (ISIS-VU and ORNL),
- highly accurate fine grained camera control (ORNL),
- modular micro-operating system for sensor networks (UC Berkeley),
- low-power mote design (UC Berkeley, Crossbow, OSU)

### 2.1.4.c. Networks of Distributed Sensors

***VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems***
VisualSense is a modeling and simulation framework for wireless and sensor networks that builds on and leverages Ptolemy II. Modeling of wireless networks requires sophisticated representation and analysis of communication channels, sensors, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. This modeling framework is designed to support a component-based construction of such models. It supports actor-oriented definition of network nodes, wireless communication channels, physical media such as acoustic channels, and wired subsystems. The software architecture consists of a set of base classes for defining channels and sensor nodes, a library of subclasses that provide certain specific channel models and node models, and an extensible visualization framework. Custom nodes can be defined by subclassing the base classes and defining the behavior in Java or by creating composite models using any of several Ptolemy II modeling environments. Custom channels can be defined by subclassing the WirelessChannel base class and by attaching functionality defined in Ptolemy II models. It is intended to enable the research community to share models of disjoint aspects of the sensor nets problem and to

build models that include sophisticated elements from several aspects. VisualSense can be downloaded from http://ptolemy.eecs.berkeley.edu/visualsense/.

***Viptos: a Programming Models for Sensor Networks***

Viptos (Visual Ptolemy and TinyOS) is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. Viptos allows developers to create block and arrow diagrams to construct TinyOS programs from any standard library of nesC/TinyOS components. The tool automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target hardware. In particular, Viptos includes the full capabilities of VisualSense, which can model communication channels, networks, and non-TinyOS nodes. Viptos is compatible with nesC 1.2 and includes tools to harvest existing TinyOS components and applications and convert them into a format that can be displayed as block (and arrow) diagrams and simulated.

Viptos is based on TOSSIM and Ptolemy II. TOSSIM is an interrupt-level simulator for TinyOS programs. It runs actual TinyOS code but provides software replacements for the simulated hardware and models network interaction at the bit or packet level. Ptolemy II is a graphical software system for modeling, simulation, and design of concurrent, real-time, embedded systems. Ptolemy II focuses on assembly of concurrent components with well-defined models of computation that govern the interaction between components. VisualSense is a Ptolemy II environment for modeling and simulation of wireless sensor networks at the network level.

Viptos provides a bridge between VisualSense and TOSSIM by providing interrupt-level simulation of actual TinyOS programs, with packet-level simulation of the network, while allowing the developer to use other models of computation available in Ptolemy II for modeling various parts of the system. While TOSSIM only allows simulation of homogeneous networks where each node runs the same program, Viptos supports simulation of heterogeneous networks where each node may run a different program. Viptos simulations may also include non-TinyOS-based wireless nodes. The developer can easily switch to different channel models and change other parts of the simulated environment, such as creating models to generate simulated traffic on the wireless network.

Viptos inherits the actor-oriented modeling environment of Ptolemy II, which allows the developer to use different models of computation at each level of simulation. At the lowest level, Viptos uses the discrete-event scheduler of TOSSIM to model the interaction between the CPU and TinyOS code that runs on it. At the next highest level, Viptos uses the discrete-event scheduler of Ptolemy II to model interaction with mote hardware, such as the radio and sensors. This level is then embedded within VisualSense to allow modeling of the wireless channels to simulate packet loss, corruption, delay, etc. The user can also model and simulate other aspects of the physical environment including those detected by the sensors (e.g., light, temperature, etc.), terrain, etc. Viptos can be downloaded from http://ptolemy.eecs.berkeley.edu/viptos/.

***Control of Communication Networks***

In a series of papers Abate and co-authors have explored using stochastic hybrid systems congestion control schemes for both wired and wireless networks. These methods have tremendous applicability to other classes of network embedded systems as well.

### 2.1.4.d. Fault-Driven Applications

***Online Hierarchical Fault-Adaptive Control***

This research extends model based predictive control (MPC) methods to a class of hybrid systems that are regulated by a finite control set. We call this type of hybrid system *switching hybrid system*. The effectiveness of proposed methods are examined by the mathematical analysis as well as simulation experiments and real-time applications. We are currently investigating the following issues:

- Use hybrid bond graphs to build the models of switching hybrid systems
- Develop MPC algorithms for controlling the switching hybrid systems using limited look-ahead control (LLC) schemes. Set point and utility-based algorithms have been developed. These schemes have also been applied to Fault-Adaptive control by including diagnosers in the control loop.
- Design and implement multi-level control framework for distributed control of interacting hybrid subsystems. This combines resource management and scheduling at the higher levels of the hierarchy with utility-based and set-point control at the subsystem and component levels of the system.
- Study practical stability analysis issues for the developed control algorithms using reachability analysis method.
- Experimental studies. We have implemented the set-point based LLC method on the three tank test-bed in the EHS laboratory at Vanderbilt University. We are currently developing a multi-level control algorithm that will be applied to the air and water recovery systems of the Advanced Life Support (ALS) system being developed for long-duration manned NASA missions.

Our approach to fault-adaptive control is centered on model-based approaches for fault detection, fault isolation and estimation, and hierarchical online supervisory control for hybrid systems. The plant is assumed to be a hybrid system. The control approach proposed in this papers targets a special class of hybrid systems in which the controlled input to the system is characterized by a finite control set. The state space equations describing the continuous dynamics of this class of systems is:

$$x(k+1) = f(x(k), u(k), \lambda(k)),$$

where $k$ is the time index, $x(k) \in X \subseteq \Re^n$ is the sampled form of the continuous state vector, $u(k) \in U \subseteq \Re^m$ is the discrete valued input vector, and $\lambda(k) \in \Re^r$ is the environment input at time $k$. The set $U$ is finite. The above model is general enough to describe a wide class of hybrid systems, including nonlinear systems and piecewise linear systems. The requirement that the input set is finite is not uncommon in practical computer-controlled systems, where the control inputs are usually discrete and take values from a finite set.

Controller specifications are classified into two categories. The first is set-point specification and the second is performance specifications. The objective of the control structure is to achieve the desired level of the set-point specifications in "reasonable" time, maintain the system stable at the desired value, and optimize the given performance function. Note that, due to the nature of the system environment, it is common that the variables used to optimize the performance functions are evaluated over a quantized finite domain. In certain situations, the optimal operation point can be computed at design time, and used as a set-point objective for the system controller. In this case, the performance function can be translated into a linear or integer

programming problem. We assume that optimal points for performance functions can be computed; therefore, the specification is given as one or more set-points, or a state-space region. The specifications may change during operation, and the proposed approach can accommodate the changes.

In the LLC approach, the controller explores only a limited forward horizon in the system state space and selects the next event based on the available information. Any system designed for a particular purpose must achieve specific objectives, and, at the same time not violate resource constraints and interactions with the environment. In general, *cost optimization* can be used to optimize a given performance measure represented as a function of system states and inputs. A weighted norm of the form,

$$J(k) = \| x(k) - r(k) \|_Q + \| u(k) \|_R$$

is typically used as a performance function in which a weighted sum of relevant variables is computed, with the weights reflecting their contribution to the system utility and operation cost. In this paper, we consider the case when $r(k)$ is a point, say $x^*$. This form of specification is generally called a set-point specification.

Operational requirements for distributed computation systems may involve additional strict and soft constraints on the system variables and control inputs. In general, strict constraints can be expressed as a feasible domain for the composite space of a set of system variables and control inputs, and they can be represented in general by a set of inequalities, $h(x, u') \leq 0$ and a restricted set of inputs $U' \subseteq U$. The optimizing component to safety control is introduced as a utility function, $\sum_i V_i(p_i)$, where each $V_i$ corresponds to a value function associated with performance parameter, $P_i$. The parameters, $p_i$, can be continuous or discrete-valued, and they are derived from the system state variables, i.e., $P_i(t) = p_i(x(t))$. The value functions employed is a simple weighted functions of the form $V_i(P_i) = w_i P_i$.

**The Limited Lookahead Control (LLC) Approach** – The objective of the control algorithm is to achieve the desired set-point specifications, maintain the system stable at the desired value, and optimize the given performance function while satisfying a set of constraints. In general, there is no closed-form solution for such non-linear optimization problems. We propose an approximate solution approach based on limited-lookahead. To this end, the control problem is defined as a minimization of the weighted norm across time. Given that the control set is finite, the control problem is solvable, though it may not produce the optimal trajectory solution for the original control problem. However, in many practical situations, the main concern is the feasibility of the online controller, namely, its ability to drive the system towards the desired operation domain ”quickly" and maintain it in this region under typical variations in the system or environmental conditions. The feasibility of the online control approach is discussed briefly in the next section.

Based on the above settings, the online controller aims to satisfy the desired set-point specifications by continuously monitoring the current system state and selecting the inputs that best satisfy them. In this setting, the controller is simply considered an agent that applies a given sequence of events in order to achieve a certain objective. The controller explores only a limited forward horizon, $N$ time steps, in the system state space. The controller then selects the trajectory that minimizes the cost function while satisfying the constraints, $h$. The input at the first look-ahead step in this trajectory is chosen as the next input, and this process is repeated at each subsequent time step.

The above control policy takes into account the effect of possible variations in the environment inputs by requiring that the selected input satisfy a worst case scenario constraint with respect the estimation bounds. Relevant parameters of the operating environment are estimated and used by the system model to forecast future behavior over a limited look-ahead horizon. The controller optimizes the system behavior by selecting the best control inputs while making sure the specified constraints are not violated.

**Control Stability** - Giving the limited exploration nature of the online algorithm, it is important to obtain a measure of feasibility to determine if the online control will be able to reach the desired region in a finite time. The controller is feasible for a given set-point and tolerance domain containing the set-point if it can drive the system (in finite time) from any initial state in a given operation region to a neighborhood (contained in the tolerance domain) of the set-point and maintain the system within this neighborhood. The feasibility of the proposed online control approach is formulated as a joint containability and attraction problem. A novel computational procedure based on nonlinear programming is presented to compute a containable region.

**Multi-level Control for Distributed Systems** – Since a detailed behavioral model of the underlying distributed system may be very complex, the global controller uses an abstract (simplified) model to describe the composite behavior of the system components that is relevant to the overall requirements and operational constraints. The abstract model uses a set of global variables that are related by the input-output interactions between the individual systems. Moreover, the global controller's decisions are based on aggregate behaviors, which are determined over longer time frames compared to the individual systems. We assume here that these time-frames are harmonically related, i.e., $T_g = MT_l$ where $T_g$ and $T_l$ are the global and local time steps, respectively. Consequently, for a set of systems, the global state vector, $y(k_g)$, at global time instance $k$ can be represented as,

$$y(k_g) = \Omega(x^1(k_l,M), \ldots, x^L(k_l,M)),$$

where $k_l = M\, k_g$ and $M$ is a positive integer, and $x^j(k_l,M) = \{x^j(k_l - M + 1), \ldots, x^j(k_l)\}$ is the set of states for the $i$th system, and $\Omega$ is the abstraction map defining the relationship between the global state vector y at the global time instance $k_g$ and the local state variables over the local time instances spanning $[k_{g-1}\ k_g]$. Similarly, we can define the global environment inputs, $\mu(k_g)$, for the global controller at time $k_g$ as an aggregation of the local environment inputs $\lambda^j(k_l)$, over the global time frame, namely, $\mu(k_g) = \Gamma(\lambda^1(k_l;M), \ldots, \lambda^L(k_l;M))$, where $\lambda^j(k_l;M) = \{\lambda^j(k_l-M+1), \ldots, \lambda^j(k_l)\}$. The global model is represented by

$$y(k+1) = g(y(k), v(k), \mu(k)),$$

where $v(k) \in V$ and $V$ is the set of global control inputs, which represents a set of local control settings for the local modules. We assume that the set of such local control settings that can be manipulated by the system controller is finite. The map $g$ defines how the global state variables respond to relevant changes in environment inputs with respect to the global control inputs. This abstract behavior can be obtained analytically (in case of simple local dynamics) or more likely through simulation where the arguments are the input set $V$ and a quantized approximation of the domain of $\mu$. It is typical that an initial model is built through simulation and then adjusted through continuous observation of the actual system behavior. The objective of the system controller is to minimize a given cost function $J_g(y,v)$ over the operation span of the system. We also assume that $J_g$ takes the form of the set point specification described earlier for local controllers. Based on the assumption that global specification is of higher priority than local

ones, the outcome of the system controller is communicated to local modules. The local controllers then try to optimize the performance of the local components while ensuring that conditions imposed by the system controller are not violated. To summarize, in the hierarchical control scheme, the system controller performs the following functions:

- Forecasts long-term trends of the environment and based on the abstract system model examines the effect on the overall performance of the system.
- Optimizes the system performance by changing the operational settings of local module, or the distribution of loads and resources among these modules
- Obtains performance feedback from local modules, which then used to identify the current global state.

We have successfully demonstrated the use of this system for an online real time application that involves our three tank system testbed at Vanderbilt University and for a NASA application that involves components of the Advanced Life Support Systems for long-term manned missions.

***Development of engine models for combustion engine for controller synthesis***

In this work we have derived efficient engine controls using the model based strategy and in that sense, having a good plant model is crucial. We also developed controllers for the engine that have been tested in simulation, giving good result. The papers [27][98] show the work in detail.

Automotive engine models vary in their complexity depending on the intended application. Pre-prototype performance prediction models can be very complex in order to make accurate predictions. Controller design models need to be as simple as possible since model-based controllers must operate in real time. This paper develops hybrid models for engine control that incorporate time and events in their formulation. The resulting hybrid controllers have the capability of switching between two alternative control modes. The first mode is designed to reduce the raw hydrocarbon (HC) emissions while the second mode tries to increase the temperature of the catalytic converter as rapidly as possible during the initial transient or ‘‘cold start’’ period. Reachability, as a tool for system analysis, is used to verify the properties of the closed loop system [27].

The control of emissions has been addressed in the past to comply with environmental regulations. In particular air-to-fuel ratio control is key to reach the allowed pollution levels. The aim of this work is to present an alternative approach which allows for more flexibility to account for the type of signals and requirements of automotive applications, specifically, handling of time and event triggered tasks. An Air-Fuel Ratio nonlinear controller is developed for an automobile engine. The controller is then implemented using the event-driven real-time programming language xGiotto on the OSEK platform provided by WindRiver. Special attention is given to show the advantage that can be gained from using an event driven paradigm for implementing automotive controllers [98].

***Fault tolerant distributed systems***

Designing cost-sensitive real-time control systems for safety-critical applications requires a careful analysis of both performance versus cost aspects and fault coverage of fault tolerant solutions. This further complicates the difficult task of deploying the embedded software that implements the control algorithms on a possibly distributed execution platform (for instance in automotive applications). In this work [111], we present a novel technique for constructing a fault tree that models how component faults may lead to system failure. The fault tree enables us to use existing commercial analysis tools to assess a number of dependability metrics of the

system. Our approach is centered on a model of computation, Fault Tolerant Data Flow (FTDF), that enables the integration of formal verification techniques. This new analysis capability is added to an existing design framework, also based on FTDF, that enables a synthesis-based, correct-by-construction, design methodology for the deployment of real-time feedback control systems in safety critical applications.

## 2.2. Project Findings

Abstracts for key publications representing project findings during this reporting period, are provided here. A complete list of publications that appeared in print during this reporting period is given in Section 3 below, including publications representing findings that were reported in the previous annual report.

**[1] A visually-specified code generator for Simulink/Stateflow**

*Neema, S.; Kalmar, Z.; Feng Shi; Vizhanyo, A.; Karsai, G., 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, pp. 275- 277, 20-24 Sept. 2005*

> Visual modeling languages are often used today in engineering domains, Mathworks' Simulink/Stateflow for simulation, signal processing and controls being the prime example. However, they are also becoming suitable for implementing other computational tasks, like model transformations. In this paper we briefly introduce GReAT: a visual language with simple, yet powerful semantics for implementing transformations on attributed, typed hypergraphs with the help of explicitly sequenced graph transformation rules. The main contribution of the paper is a Simulink/Stateflow code generator that generates executable code (running on a distributed platform) from the visual input models. The paper provides an overview of the algorithms used and their realization in GReAT.

**[2] Developing Applications Using Model-Driven Design Environments**

*Balasubramanian, K.; Gokhale, A.; Karsai, G.; Sztipanovits, J.; Neema, S., Computer, vol.39, no.2, pp. 33- 40, Feb. 2006*

> Model-driven development is an emerging paradigm that improves the software development lifecycle, particularly for large software systems, by providing a higher level of abstraction for system design than is possible with third-generation programming languages.

**[3] Applying a Model Transformation Taxonomy to Graph Transformation Technology**

*Tom Mens, Pieter Van Gorp, Dániel Varró and Gabor Karsai, Electronic Notes in Theoretical Computer Science, Volume 152, Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005), 27 March 2006, Pages 143-159.*

> A taxonomy of model transformations was introduced in T. Mens, P.V. Gorp, A taxonomy of model transformation, in: Proc. Int'l Workshop on Graph and Model Transformation (GraMoT 2005), Electronic Notes in Computer Science (2005)]. Among others, such a taxonomy can help developers in deciding which language, forma lism,

tool or mechanism is best suited to carry out a particular model transformation activity. In this paper we apply the taxonomy to the technique of graph transformation, and we exemplify it by referring to four representative graph transformation tools. As a byproduct of our analysis, we discuss how well each of the considered tools carry out the activity of model transformation.

**[4] Improving the Usability of a Graph Transformation Language**

*Attila Vizhanyo, Sandeep Neema, Feng Shi, Daniel Balasubramanian and Gabor Karsai, Electronic Notes in Theoretical Computer Science, Volume 152, Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005), 27 March 2006, Pages 207-222.*

Model transformation tools implemented using graph transformation techniques are often expected to provide high performance. For this reason, in the Graph Rewriting and Transformation (GReAT) language we have supported two techniques: pre-binding of selected pattern variables and explicit sequencing of transformation steps to improve the performance of the transformation engine. When applied to practical situations, we recognized three shortcomings in our approach: (1) no support for the convenient reuse of results of one rewriting step in another, distant step, (2) lack of a sorting capability for ordering the results of the pattern matching, and (3) absence of support for the distinguished merging of results of multiple pattern matches. In this paper we briefly highlight the relevant features of GReAT, describe three motivating examples that illustrate the problems, introduce our solutions: new extensions to the language, and compare the approaches to other languages.

**[5] Reusable Idioms and Patterns in Graph Transformation Languages**

*Aditya Agrawal, Attila Vizhanyo, ZsoltKalmar, Feng Shi, Anantha Narayanan and Gabor Karsai, Electronic Notes in Theoretical Computer Science, Volume 127, Issue 1, Proceedings of the International Workshop on Graph-Based Tools (GraBaTs 2004), 30 March 2005, Pages 181-192.*

Software engineering tools based on Graph Transformation techniques are becoming available, but their practical applicability is somewhat reduced by the lack of idioms and design patterns. Idioms and design patterns provide prototypical solutions for recurring design problems in software engineering, but their use can be easily extended into software development using graph transformation systems. In this paper we briefly present a simple graph transformation language: GReAT, and show how typical design problems that arise in the context of model transformations can be solved using its constructs. These solutions are similar to software design patterns, and intend to serve as the starting point for a more complete collection.

**[6] Case Study: Model Transformations for Time-triggered Languages**

*Tivadar Szemethy, Electronic Notes in Theoretical Computer Science, Volume 152, Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005), 27 March 2006, Pages 175-190.*

In this study, we introduce a model transformation tool for a time-triggered language: Giotto. The tool uses graphs to represent the source code (Giotto) and the target (the schedule-carrying code) of the transformation, and has been implemented entirely using graph rewriting techniques. The meta-models of the input and the output were specified using standard (UML) technology, and the transformation itself as a programmed graph rewriting system (in GReAT). The approach illustrates how a non-trivial model transformation can be implemented using graph transformations, and how the results obtained here could be used for the formal verification of embedded systems models. The transformation developed here forms the first step towards translating high-level, domain-specific models (that use concepts of the time-triggered language) into analysis models (that use concepts from the language of the analysis, e.g. timed automata). Using a formal approach such as graph transformation helps ensure the correctness of this transformation process.

**[7] Corrected Through Construction: A Model-Based Approach to Embedded Systems Reality**

*Jackson, E., Sztipanovits, J. Proceedings of ECBS'06, pp. Potsdam, Germany, March 27-30, 2006*

We detail a scalable and formal specification language for embedded systems called SMOLES2. This specification language is build on top of the metaprogrammable tool GME and uses modern model-based techniques like multi-aspects, generative actions, and constraint checking to auto-generate parts of a specification and to approximate the correctness of the specification without invoking verification tools.

**[8] Toward a Semantic Anchoring Infrastructure for Domain-Specific Modeling Languages**

*Chen K., Sztipanovits J., Neema S., Emerson M., Abdelwahed S. Proceedings of the Fifth ACM International Conference on Embedded Software (EMSOFT'05), pp. 35-44, Jersey City, New Jersey, September 19, 2005.*

Metamodeling facilitates the rapid, inexpensive development of domain-specific modeling languages (DSML-s). However, there are still challenges hindering the wide-scale industrial application of model-based design. One of these unsolved problems is the lack of a practical, effective method for the formal specification of DSML semantics. This problem has negative impact on reusability of DSML-s and analysis tools in domain specific tool chains. To address these issues, we propose a formal well founded methodology with supporting tools to anchor the semantics of DSML-s to precisely defined and validated "semantic units". In our methodology, each of the syntactic and semantic DSML components is defined precisely and completely. The main contribution of our approach is that it moves toward an infrastructure for DSML design that integrates

formal methods with practical engineering tools. In this paper we use a mathematical model, Abstract State Machines, a common semantic framework to define the semantic domains of DSML-s.

**[9] Semantic Anchoring with Model Transformations**

*Chen K., Sztipanovits J., Abdelwahed S., Jackson E. European Conference on Model Driven Architecture -Foundations and Applications (ECMDA-FA), Nuremberg, Germany, November 7, 2005. Lecture Notes in Computer Science, (LNCS 3748) pp. 115-129, Springer 2005*

Model-Integrated Computing (MIC) places strong emphasis on the use of domain-specific modeling languages (DSML-s) and model transformations. A metamodeling process facilitated by the Generic Modeling Environment (GME) tool suite enables the rapid and inexpensive development of DSML-s. However, the specification of semantics for DSML-s is still a hard problem. In order to simplify the DSML semantics, this paper discusses semantic anchoring, which is based on the transformational specification of semantics. Using a mathematical model, Abstract State Machine (ASM), as a common semantic framework, we have developed formal operational semantics for a set of basic models of computations, called semantic units. Semantic anchoring of DSML-s means the specification of model transformations between DSML-s (or aspects of complex DSML-s) and selected semantic units. The paper describes the semantic anchoring process using the meta-programmable MIC tool suite.

**[10] A Semantic Unit for Timed Automata Based Modeling Languages**

*Chen K., Sztipanovits J., Abdelwahed S. Proceedings of RTAS'06 pp. 347-360, San Jose, CA, April 4-7, 2006*

Model-Integrated Computing (MIC) is an infrastructure for model-based design of real-time and embedded software and systems. MIC places strong emphasis on the use of domain-specific modeling languages (DSMLs) and model transformations in design flows. Building on our earlier work on transformational specification of semantics for DSMLs, the paper proposes a "semantic unit" - a common semantic model - for timed automata behavior. The semantic unit is defined using Abstract State Machine (ASM) formalism. We show that the precise semantics of a wide range of timed automata based modeling languages (TAMLs) can be defined through specifying model transformations between a domain-specific TAML and the semantic unit. The proposed method that we call semantic anchoring is demonstrated by developing the transformation rules from the UPPAAL and IF languages to the semantic unit.

**[11] Compositional Specification of Behavioral Semantics**

*Chen K., Sztipanovits J., Neema, S. Technical Report 2006-14, ISIS, Vanderbilt University May, 2006 (submitted paper)*

Domain-Specific Modeling Languages (DSMLs) play fundamental role in the model-based design of embedded software and systems. In previous work, we have developed methods and tools for the semantic anchoring of DSMLs. Semantic anchoring introduces a set of reusable "semantic units" that provide reference semantics for basic behavioral categories using the Abstract State Machine framework. In this paper, we extend the

semantic anchoring framework to heterogeneous behaviors by developing method for the composition of semantic units. Semantic unit composition reduces the required effort from DSML designers and improves the quality of the specification. The proposed method is demonstrated through a case study.

**[12] Towards A Formal Foundation For Domain Specific Modeling Languages**

*Jackson, E., Sztipanovits, J. Technical Report 2006-15, ISIS, Vanderbilt University May, 2006 (submitted paper)*

Embedded system design is inherently domain specific and typically model driven. As a result, design methodologies like OMG's model driven architecture (MDA) and model integrated computing (MIC) evolved to support domain spe- cific modeling languages (DSMLs). The success of the DSML approach has encouraged work on the heterogeneous composition of DSMLs, model transformations between DSMLs, approximations of formal properties within DSMLs, and reuse of DSML semantics. However, in the effort to produce a mature design approach that can handle both the structural and behavioral semantics of embedded system design, many foundational issues concerning DSMLs have been overlooked. In this paper we present a formal foundation for DSMLs and for their construction within metamodeling frameworks. This foundation allows us to algorithmically decide if two DSMLs or metamodels are equivalent, if model transformations preserve properties, and if metamodeling frameworks have meta-metamodels. These results are key to building correct embedded systems with DSMLs.

**[13] Using Separation of Concerns in Embedded Systems Design**

*Jackson, E., Sztipanovits, J. Proceedings of the Fifth ACM International Conference on Embedded Software (EMSOFT'05), pp. 25-34, Jersey City, New Jersey, September 19, 2005.*

Embedded systems are commonly abstracted as collections of interacting components. This perspective has lead to the insight that component behaviors can be defined separately from admissible component interactions. We show that this separation of concerns does not imply that component behaviors can be defined in isolation from their envisioned interaction models. We argue that a type of behavior/interaction co-design must be employed to successfully leverage the separation of these concerns. We present formal techniques for accomplishing this co-design and describe tools that implement these formalisms.

**[14] Online Fault-Adaptive Control for Efficient Resource Management in Advanced Life Support Systems**

*S. Abdelwahed, J. Wu, G. Biswas, J. Ramirez, and E.J. Manders, Habitation: International Journal of Human Support Research, vol. 10, no. 2, pp. 105-115, 2005.*

This paper presents the design and implementation of a controller scheme for efficient resource management in Advanced Life Support Systems. In the proposed approach, a switching hybrid system model is used to represent the dynamics of the system components and their interactions. The operational specifications for the controller are represented as a utility function, and the corresponding resource management problem is

formulated as a safety control problem. A limited-horizon online supervisory controller is used for this purpose. The online controller explores a limited region of the state-space of the system at each time step and uses the utility function to decide on the best action. The feasibility and accuracy of the online algorithm can be assessed at design time. We demonstrate the effectiveness of the scheme by running a set of experiments on the Reverse Osmosis (RO) subsystem of the Water Recovery System (WRS).

**[15] Introducing Embedded Software and Systems Education and Advanced Learning Technology in an Engineering Curriculum**

*J. Sztipanovits, G. Biswas, K. Frampton, A. Gokhale, L. Howard, G. Karsai, J. Koo, X. Koutsoukos, and D. Schmidt, Special issue, ACM Trans. on Embedded Systems (TECS), vol. 4, no. 3, pp. 549-568, August 2005.*

Embedded software and systems are at the intersection of electrical engineering, computer engineering, and computer science, with, increasing importance, in mechanical engineering. Despite the clear need for knowledge of systems modeling and analysis (covered in electrical and other engineering disciplines) and analysis of computational processes (covered in computer science), few academic programs have integrated the two disciplines into a cohesive program of study. This paper describes the efforts conducted at Vanderbilt University to establish a curriculum that addresses the needs of embedded software and systems. Given the compartmentalized nature of traditional engineering schools, where each discipline has an independent program of study, we have had to devise innovative ways to bring together the two disciplines. The paper also describes our current efforts in using learning technology to construct, manage, and deliver sophisticated computer-aided learning modules that can supplement the traditional course structure in the individual disciplines through out-of-class and in-class use.

**[16] Building Efficient Simulations from Hybrid Bond Graph Models**

*C. Beers, E.J. Manders, G. Biswas, and P. Mosterman, 2nd IFAC Conference on Analysis and Design of Hybrid Systems, Sardinia, Italy, June 2006.*

Embedded systems and their corresponding hybrid models are pervasive in engineering applications, therefore, systematic mathematical analysis using these models has become an important research area. Our approach to hybrid modeling with Hybrid Bond Graphs allows for seamless integration of physical system principles with discrete computational structures, but simulating the hybrid behaviors can be difficult and computationally expensive. In this paper, we develop a methodology that transforms Hybrid Bond Graphs into computational block diagrams and incrementally modifies the block diagram when mode changes occur. This forms the basis for a computationally efficient hybrid simulation algorithm.

**[17] Hierarchical Online Control Design for Autonomous Resource Management in Advanced Life Support Systems**

*S. Abdelwahed, J. Wu, G. Biswas, and E. J.-Manders, International Conference on Environmental Systems, Paper no. 2005-01-2965, Rome, Italy, July 2005.*

This paper presents a distributed, hierarchical control scheme for autonomous resource management in complex embedded systems that can handle dynamic changes in resource constraints and operational requirements. The developed hierarchical control structure handles the interactions between subsystem and system-level controllers. A global coordinator at the root of the hierarchy ensures resource requirements for the duration of the mission are not violated. We have applied this approach to design a three-tier hierarchical controller for the operation of a lunar habitat that includes a number of interacting life support components.

**[18] Requirements for an Autonomous Control Architecture for Advanced Life Support Systems**

*G. Biswas, P. Bonasso, S. Abdelwahed, E.J. Manders, J. Wu, D. Kortenkamp, and S. Bell, International Conference on Environmental Systems, Paper no.2005-01-3010, Rome, Italy, July 2005.*

This paper builds upon a series of life support control experiments conducted at NASA Johnson Space Center and at Vanderbilt University. The experiments used two distinct, layered control architectures: (i) a model-based approach, and (ii) a procedural approach to control complex, distributed systems. Both sets of experiments produced good results, but they also brought out the strengths and weaknesses of the two in the context of requirements for long duration human missions. Our goal in this paper is to come up with the design requirements for an integrated architecture for autonomous controller design that combines the best of the two approaches. This paper provides a framework for integrated design and discusses the advantages it offers.

**[19] Multi-scale Modeling of Advanced Life Support Systems**

*E.J.-Manders, S. Bell, G. Biswas, and D. Kortenkamp, International Conference on Environmental Systems, Paper no.2005-01-113, Rome, Italy, July 2005.*

Regenerative life support systems for long duration human space exploration missions present unique design challenges that are also reflected in constructing behavior models of these systems for analysis purposes. These systems have multiple modes of operation and complex non-linear dynamics that occur at multiple time scales. Coarse grained analysis of the complete system over long duration and fine grained analysis of smaller system elements while avoiding computational intractability can be achieved by using multiple modeling and simulation paradigms. We describe a multi-level simulation model of an advanced life support system. The simulation model couples a discrete-event approach at the system level, with more detailed hybrid (continuous/discrete) physical system modeling at the sub-system level.

**[20] Component-oriented modeling of hybrid dynamic systems using the Generic Modeling Environment**

*Eric-J. Manders, Gautam Biswas, Nagabhushan Mahadevan, Gabor Karsai, pp. 159-168, Fourth Workshop on Model-Based Development of Computer-Based Systems and Third International Workshop on Model-Based Methodologies for Pervasive and Embedded Software (MBD-MOMPES'06), 2006.*

This paper presents a component oriented modeling environment for building hybrid dynamic models of physical systems. The modeling environment is created using the Generic Modeling Environment (GME), a meta programmable visual modeling application developed at the Institute for Software Integrated Systems (ISIS). The core of the modeling language itself is a hybrid extension of the bond graph modeling language. The advantages of an object-oriented approach to physical system modeling combined with the advanced features of GME for managing model complexity are illustrated by building a library of hydraulic system components. A simulation model can be automatically generated from the physical system model using a model translator. As an example application we use the component library to build the model of a coupled multi-tank system with controlled and autonomous hybrid behaviors, and illustrate this with a simulation example.

**[21] A Hybrid Control System Design and Implementation for a Three-tank Testbed**

*J. Wu, G. Biswas, S. Abdelwahed, and E.J. Manders, IEEE Conference on Control Applications, Toronto, CA, pp. 645-650, August 2005.*

This paper discusses methodologies for designing online supervisory controllers for a class of embedded systems that can be modeled as switching hybrid system (SHS). In the online control approach, a limited forward horizon of possible behaviors is explored at each time step. The controller decides the best action using a cost function to determine the *best* state on the horizon. We discuss the controller design and implementation for a three-tank system test-bed with distributed sensor and actuation units. A set of real-time fault adaptive control experiments demonstrate the effectiveness of the approach.

**[22] Cascaded Control Design for a Quadrotor Aerial Robot**

*T. J. Koo, C. A. Clifton and G. Hemingway: In Proceedings of the Asian Control Conference, Bali, Indonesia, July 2006.*

Aerial robot can swiftly react to changes in dynamical environment, by properly executing a sequence of controllers. In this paper, a cascaded controller design based on an outer-inner model of the quadrotor aerial robot is presented. The cascaded control structure enables the design of multi-modal controller in the outer loop for executing autonomous mission while reducing control design complexity by using a single controller for the inner loop. An outer controller is designed by using the differential flatness property of the outer system and the backstepping design technique in order to take advantage of the nonlinear nature of the system. A robust linear inner controller is designed based on an identified inner model to deal with model uncertainty and disturbance. Given a desired output trajectory, the outer controller can generate desired inner trajectory for the inner controller to track. The controllers are designed to guarantee

bounded output tracking and bounded state performance for bounded desired output trajectory in the presence of anticipated disturbance. The control design is implemented and the experimental results are presented.

**[23] A Semantic Anchoring Infrastructure for Model-based Embedded System Design**

*G. Hemingway, H. Su, K. Chen, and T. J. Koo: Technical Report 2006-16, ISIS, Vanderbilt University May, 2006 (submitted paper)*

Model-Integrated Computing (MIC) is an infrastructure for model-based design of embedded software and systems. MIC places strong emphasis on the use of Domain Specific Modeling Language (DSMLs) and model transformations in design flows. Practical and effective development of formal specifications for DSML semantics within model-based tools can be challenging, but could positively impact adoption and reuse of these tools. The semantic anchoring methodology was developed to address this challenge by formally tying DSMLs to a ``semantic unit'', which is a formal specification that captures the operational semantics of a specific model of computation. Leveraging our prior work with semantic units, we develop a semantic unit for hybrid automata. Hybrid automata can be used to model system-level behaviors for embedded systems that exhibit strong couplings between discrete and continuous dynamics. In this paper, we explicitly specify the operational semantics of hybrid automata, and develop the corresponding semantic unit and model transformation rules. We demonstrate the effectiveness of the infrastructure in a practical case study involving the hybrid automata DSMLs, HyVisual and ReachLab.

**[24] Advantages and challenges of distributed active vibro-acoustic control**

*Frampton, K.: The Journal of the Acoustical Society of America -- September 2005 -- Volume 118, Issue 3, p. 1950*

As active control technologies reach their performance limits in large scale systems, many investigators have looked toward decentralized control as a means of expanding the application horizons. Decentralized control is defined here as numerous independent controllers operating on a single system. These decentralized approaches have been shown to be effective, but not as effective as traditional centralized control. In an effort to achieve control performance approaching centralized control while maintaining the scalability benefits of decentralized control, the use of distributed control is considered. Distributed control consists of numerous control processors operating on a system that are capable of communicating with each other over a network. This work discusses the application of distributed active control to vibroacoustic problems. Key elements in distributed control systems will be presented along with the state of each of the key technologies involved. Of particular interest are the limitations in enabling technologies that limit the application of distributed control: namely real-time network communications; distributed control algorithms and design tools; software infrastructure for system management; and other factors. Results will demonstrate that distributed control can perform nearly as well as centralized control, but that it can also be ``scaled up'' for use in large complex systems.

### [25] A Causality Interface for Deadlock Analysis in Dataflow

*Ye Zhou and Edward A. Lee. A Causality Interface for Deadlock Analysis in Dataflow. Submitted to EMSOFT, May, 2006.*

> In this paper, we consider a concurrent model of computation called dataflow, where components (actors) communicate via streams of data tokens. Dataflow semantics has been adopted by experimental and production languages used to design embedded systems. The execution of a data-flow actor is enabled by the availability of its input data. One important question is whether a dataflow model will deadlock (i.e., actors cannot execute due to a data dependency loop). Deadlock in many cases can be determined, although it is generally not decidable. We develop a causality interface for dataflow actors based on the general framework we introduced in [1] and show how this causality information can be algebraically composed so that composition of components acquire causality interfaces that are inferred from their components and the interconnections. We illustrate the use of these causality interfaces to statically analyze for deadlock.

### [26] A Formalism for Higher-Order Composition Languages that Satisfies the Church-Rosser Property

*Adam Cataldo, Elaine Cheong, Thomas Huining Feng, Edward A. Lee and Andrew Mihal. Technical report, EECS Dept., University of California, Berkeley, 48, May, 2006.*

> In actor-oriented design, programmers make hierarchical compositions of concurrent components. As embedded systems become increasingly complex, these compositions become correspondingly complex in the number of actors, the depth of hierarchies, and the connections between ports. We propose higher-order composition languages as a way to specify these actor-oriented models. The key to these languages is the ability to succinctly specify configurations with higher-order parameters---parameters that themselves might be configurations. We present a formalism which allows us to describe arbitrarily complex configurations of components with higher-order parameters. This formalism is an extension of the standard lambda calculus.

### [27] Automotive engine hybrid modelling and control for reduction of hydrocarbon emissions

*P.R. Sanketi, J.C. Zavala and J.K. Hedrick. International Journal of Control, 79(5):449-464, May 2006.*

> Automotive engine models vary in their complexity depending on the intended application. Pre-prototype performance prediction models can be very complex in order to make accurate predictions. Controller design models need to be as simple as possible since model-based controllers must operate in real time. This paper develops hybrid models for engine control that incorporate time and events in their formulation. The resulting hybrid controllers have the capability of switching between two alternative control modes. The first mode is designed to reduce the raw hydrocarbon (HC) emissions while the second mode tries to increase the temperature of the catalytic converter as rapidly as possible during the initial transient or ‘‘cold start’’ period. Reachability, as a tool for system analysis, is used to verify the properties of the closed loop system.

**[28] New real-time embedded software for an autonomous helicopter system using Giotto**

*Jongho Lee. New real-time embedded software for an autonomous helicopter system using Giotto. Master's thesis, UC Berkeley, May, 2006.*

A new design of embedded software for an autonomous helicopter control system is presented. A helicopter is a practical hard real-time system. Its timing behaviors of embedded software are crucial for safety. Readable and maintainable control software is essential for complex systems. In this thesis, a time-based control domain specific programming language is used to fulfill timing constraints and needs of reusable software. A design procedure along with implementation and flight tests is described for a radio controlled helicopter system.

**[29] Accelerating Applications Using TIPI Sub-RISC Processing Elements**

*Scott Weber, Kaushik Ravindran, Andrew Mihal and Kurt Keutzer. Unpublished article, May, 2006.*

We introduce the TIPI sub-RISC architecture and a supporting infrastructure for designing, programming, analyzing, and implementing TIPI sub-RISC processing elements. The TIPI sub-RISC architectural abstraction encapsulates programmable architectures ranging from hard-wired data paths to RISC/VLIW processors. Although RTL design can capture TIPI processing elements, the TIPI infrastructure substantially increases design entry productivity and provides a 100-1000x increase in simulation performance when compared to RTL design. We have used the TIPI infrastructure to design two processors to accelerate an MP3 decoder. The TIPI processors achieved speedups up to 26x over the corresponding functions in software. The TIPI framework also provides a path to implementation using RTL synthesis. We used this path to implement the system on the Xilinx ML-310 FPGA platform. The main program was executed on the PowerPC core and the TIPI accelerators were built on the Virtex-II Pro 2VP30 FPGA.

**[30] Modeling Timed Concurrent Systems using Generalized Ultrametrics**

*Xiaojun Liu, Eleftherios Matsikoudis and Edward A. Lee. Modeling Timed Concurrent Systems using Generalized Ultrametrics. Technical report, EECS Department, UC Berkeley, May, 2006.*

Timed concurrent systems are used in concurrent and distributed real-time software, modeling of hybrid systems, design of hardware systems (using hardware description languages), discrete-event simulation, and modeling of communication networks. They consist of concurrent components that communicate using timed signals, which are sets of (semantically) time-stamped events. The denotational semantics of such systems is traditionally formulated in a metric space. In this metric space, causal components are modeled by contracting functions. We show that this formulation excessively restricts the models of time that can be used. In particular, it cannot handle super-dense time, commonly used in hardware description languages and hybrid systems modeling, finite time lines, and time with no origin. Moreover, if we admit continuous-time and mixed signals (essential for hybrid systems modeling) or certain Zeno signals, then causality is no longer equivalent to its formalization in terms of contracting functions. In this paper,

we offer an alternative semantic framework using a generalized ultrametric that overcomes these limitations.

**[31] The Problem with Threads**

*Edward A. Lee. The Problem with Threads. IEEE Computer, 39(5):33-42, May 2006.*

For concurrent programming to become mainstream, we must discard threads as a programming model. Nondeterminism should be judiciously and carefully introduced where needed, and it should be explicit in programs.

**[32] Automated Mapping from a Domain Specific Language to a Commercial Embedded Multiprocessor**

*William Plishker and Kurt Keutzer. Unpublished article, May, 2006; Submitted to CODES 2006.*

Application specific programmable systems are capable of high performance implementations while remaining flexible enough to support a range of applications. Architects of these systems achieve high performance through domain specific optimizations, often introduced at the expense of programming productivity. We examine one of the most performance critical and time consuming steps to arriving at efficient implementations on these platforms: the mapping of an application to a target architecture. We accelerate this step by constructing a model of the architecture which captures its key features while being amenable to automated mapping. With an analogous presentation of the application and mapping formulated as an integer linear programming (ILP) problem, we demonstrate this approach finds solutions that are guaranteed to be close to optimal solutions with respect to this model. These solutions enable efficient implementations of representative network applications on a commercial network processor family. We show this approach can produce an implementation comparable to a hand crafted design.

**[33] COP Semantics of Timed Interactive Actor Networks**

*Xiaojun Liu and Edward A. Lee. COP Semantics of Timed Interactive Actor Networks. Technical report, EECS Department, UC Berkeley, 67, May, 2006.*

We give a denotational framework for composing interactive components into closed or open systems and show how to adapt classical domain-theoretic approaches to open systems and to timed systems. For timed systems, instead of the usual metric-space-based approaches, we show that existence and uniqueness of behaviors are ensured by continuity with respect to a simply defined prefix order. Existence and uniqueness of behaviors, however, does not imply that a composition of components yields a useful behavior. The unique behavior could be empty or smaller than expected. We define liveness and show that appropriately defined causality conditions ensure liveness and freedom from Zeno conditions. In our formulation, causality does not require a metric and can embrace a wide variety of models of time.

### [34] Hierarchical Timing Language

*Arkadeb Ghosal, Thomas A. Henzinger, Daniel Iercan, Christoph Kirsch and Alberto L. Sangiovanni-Vincentelli. Hierarchical Timing Language. Technical report, EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-20, May, 2006.*

> We have designed and implemented a new programming language for hard real-time systems. Critical timing constraints are specified within the language, and ensured by the compiler. The main novel feature of the language is that programs are extensible in two dimensions without changing their timing behavior: new program modules can be added, and individual program task can be refined. The mechanism that supports time invariance under parallel composition is that different program modules communicate at specified instances of time. Time invariance under refinement is achieved by conservative scheduling of the top level. The language, which assembles real-time tasks within a hierarchical module structure with timing constraints, is called Hierarchical Timing Language (HTL). It is a coordination language, in that individual tasks can be implemented in other languages. We present a distributed HTL implementation of an automotive steer-by-wire controller as a case study.

### [35] Analysis of Low-Level Code Using Cooperating Decompilers

*Bor-Yuh Evan Chang, Matthew Harren, and George C. Necula. Analysis of Low-Level Code Using Cooperating Decompilers. April, 2006; In submission.*

> We present a modular framework for building assembly-language program analyzers by using a pipeline of decompilers that gradually lift the level of the language to something appropriate for source-level analysis tools. Each decompilation stage contains an abstract interpreter that encapsulates its findings about the program by translating the program into a higher-level intermediate language. For the hardest decompilation tasks a decompiler may request information from higher-level stages in the pipeline. We provide evidence for the modularity of this framework through the implementation of multiple decompilation pipelines for both x86 and MIPS assembly produced by gcc, gcj, and coolc (a compiler for a pedagogical mini-Java language) that share several low-level components. Finally, we discuss our experimental results that apply the BLAST model checker for C and the Cqual analyzer to decompiled assembly.

### [36] Incremental Checkpointing with Application to Distributed Discrete Event Simulation

*Huining Thomas Feng and Edward A. Lee. Technical report, EECS Dept., University of California Berkeley, 37, April, 2006.*

> Checkpointing is widely used in robust fault-tolerant applications. We present an efficient incremental checkpointing mechanism. It requires to record only the the state changes and not the complete state. After the creation of a checkpoint, state changes are logged incrementally as records in memory, with which an application can spontaneously roll back later. This incrementality allows us to implement checkpointing with high performance. Only small constant time is required for checkpoint creation and state recording. Rollback requires linear time in the number of recorded state changes, which is bounded by the number of state variables times the number of checkpoints. We

implement a Java source transformer that automatically converts an existing application into a behavior-preserving one with checkpointing functionality. This transformation is application-independent and application-transparent. A wide range of applications can benefit from this technique. Currently, it has been used for distributed discrete event simulation using the Time Warp technique.

**[37] An Interface Algebra for Real-time Components**

*Thomas Henzinger, Slobodan Matic. Proceedings of RTAS 2006, 253-263, April, 2006.*

We present an assume-guarantee interface algebra for real-time components. In our formalism a component implements a set of task sequences that share a resource. A component interface consists of an arrival rate function and a latency for each task sequence, and a capacity function for the shared resource. The interface specifies that the component guarantees certain task latencies depending on assumptions about task arrival rates and allocated resource capacities. Our algebra defines compatibility and refinement relations on interfaces. Interface compatibility can be checked on partial designs, even when some component interfaces are yet unknown. In this case interface composition computes as new assumptions the weakest constraints on the unknown components that are necessary to satisfy the specified guarantees. Interface refinement is defined in a way that ensures that compatible interfaces can be refined and implemented independently. Our algebra thus formalizes an interface-based design methodology that supports both the incremental addition of new components and the independent stepwise refinement of existing components. We demonstrate the flexibility and efficiency of the framework through simulation experiments.

**[38] Design Space Exploration of Automotive Platforms in Metropolis**

*Haibo Zeng, Abhijit Davare, Alberto Sangiovanni-Vincentelli, Sampada Sonalkar, Sri Kanajan, Claudio Pinello. Society of Automotive Engineers Congress, April, 2006.*

Automotive control applications are implemented over distributed platforms consisting of a number of electronic control units (ECUs) connected by communication buses. During system development, the designer can explore a number of design alternatives: for example, software distribution, software architecture, hardware architecture, and network configuration. Exploring design alternatives efficiently and evaluating them to optimize metrics such as cost, time, resource utilization, and reliability provides an important competitive advantage to OEMs and helps minimize integration risks. We present a methodology (Platform-Based Design) and a framework (Metropolis) to support efficient architecture exploration. We have exercised the methodology and the capabilities of Metropolis for developing a library of automotive architecture components and performed design space exploration on a chassis control sub-system.

**[39] Concurrent Embedded Design for Multimedia: JPEG encoding on Xilinx FPGA Case Study**

*Jike Chong, Abhijit Davare, Kelvin Lwin. Technical report, UC Berkeley, April, 2006.*

Parallel platforms are becoming predominant in the embedded systems space due to a variety of factors. These platforms can deliver high peak performance if they can be

programmed effectively. However, current sequential software design techniques as well as the Single Program Multiple Data (SPMD) programming models often used in the High Performance Computing (HPC) domain are insufficient. In this report, we experiment with a dataflow programming model for multimedia embedded systems. By applying this programming model to a common application and embedded platform, we get a better idea of the implementation challenges for this class of systems.

**[40] Composition Languages**

*James Adam Cataldo and Edward A. Lee. Technical report, EECS Dept., University of California, Berkeley, 24, March, 2006; Found at: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-24.pdf.*

We propose composition languages as a way to specify actor-oriented models, or hierarchical networks of concurrent components which communicate with one another through ports. The key to composition languages is the ability to succinctly specify higher-order models. As an example, a higher-order model may be a distributed sort model. The model may be parameterized by a divide component (or model), a conquer component, and the respective numbers of divide and conquer components. A programmer will specify this higher-order model once and can then use it for an arbitrary number of components with arbitrary divide and conquer components. We believe composition languages will become increasingly important in actor-oriented design, since they will enable rapid development of large systems.

**[41] Finitary Winning in \omega-Regular Games**

*Krishnendu Chatterjee and Thomas A. Henzinger. Finitary Winning in \omega-Regular Games. TACAS, March, 2006.*

Games on graphs with \omega-regular objectives provide a model for the control and synthesis of reactive systems. Every \omega-regular objective can be decomposed into a safety part and a liveness part. The liveness part ensures that something good happens eventually. Two main strengths of the classical, infinite-limit formulation of liveness are robustness (independence from the granularity of transitions) and simplicity (abstraction of complicated time bounds). However, the classical liveness formulation suffers from the drawback that the time until something good happens may be unbounded. A stronger formulation of liveness, so-called finitary liveness, overcomes this drawback, while still retaining robustness and simplicity. Finitary liveness requires that there exists an unknown, fixed bound b such that something good happens within b transitions. While for one-shot liveness (reachability) objectives, classical and finitary liveness coincide, for repeated liveness objectives, the finitary formulation is strictly stronger. In this work we study games with finitary parity and Streett (fairness) objectives. We prove the determinacy of these games, present algorithms for solving these games, and characterize the memory requirements of winning strategies. Our algorithms can be used, for example, for synthesizing controllers that do not let the response time of a system increase without bound.

**[42] Reachability Analysis of Controlled Discrete-Time Stochastic Hybrid Systems**

*S. Amin, A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Hybrid Systems: Computation and Control, Proceedings of the 9th International Workshop, Santa Barbara, CA, vol. 3927 of Lecture Notes in Computer Science, J. Hespanha and A. Tiwari, Springer-Verlag, pp. 49-63, March, 2006.*

In this research, a model for discrete time stochastic hybrid systems is proposed. With reference to the introduced class of systems, a methodology for probabilistic reachability analysis is studied, which can be useful for safety verification. This methodology is based on the interpretation of the safety verification problem as an optimal control problem for a certain controlled Markov process. In particular, this allows us to characterize through some optimal cost function the set of initial conditions for the system such that its state can be maintained within a given "safe" set with sufficiently high probability.

**[43] Interchange Formats for Hybrid Systems: Abstract Semantics**

*Alessandro Pinto, Luca P. Carloni, Roberto Passerone and Alberto Sangiovanni-Vincentelli. Hybrid Systems: Computation and Control, Joao Hespanha and Ashish Tiwari, 491-506, March, 2006.*

In previous work we advocated the need for an interchange format for hybrid systems that enables the integration of design tools coming from many different research communities. In deriving such interchange format the main challenge is to define a language that, while presenting a particular formal semantics, remains general enough to accommodate the translation across the various modeling approaches used in the existing tools. In this paper we give a formal definition of the syntax and semantics for the proposed interchange format. In doing so, we clearly separate the structure of a hybrid system from the semantics attached to it. The semantics can be considered an ``abstract semantics" in the sense that it can be refined to yield the model of computation, or ``concrete semantics", which, in turn, is associated to the existing languages that are used to specify hybrid systems. We show how the interchange format can be used to capture the essential information across different modeling approaches and how such information can be used in the translation process.

**[44] A Platform-based Design Flow for Kahn Process Networks**

*Abhijit Davare, Qi Zhu, Alberto Sangiovanni-Vincentelli. Technical report, UC Berkeley, 2006-30, March, 2006.*

Effectively implementing multimedia applications on multiprocessor architectures is a key challenge in system-level design. This work explores automated solutions to this problem by considering two separate directions of research. First, the problem is placed within the context of a generalized mapping strategy and the concept of a common semantic domain is developed which is capable of reasoning about the automation techniques that are to be applied. Second, a specialized design flow and associated algorithms are developed to solve this problem. The idea of a common semantic domain is described and its usefulness in other mapping problems is demonstrated. For this particular problem, a common semantic domain is identified and forms the basis of the algorithms which are developed in the design flow. The design flow is divided into four

clearly defined steps, to ensure the tractability of optimization problems while obtaining a good overall solution. The separation of the flow into these steps allows prior work from a variety of sources to be used. Efficient heuristics are developed for each step of the design flow. The effectiveness of the heuristics used in this design flow is demonstrated by applying them to an industrial case study.

**[45] Communication and Co-Simulation Infrastructure for Heterogeneous System Integration**

*Guang Yang, Xi Chen, Felice Balarin, Harry Hsieh, Alberto Sangiovanni-Vincentelli. Design Automation and Test in Europe, March, 2006.*

With the increasing complexity and heterogeneity of embedded electronic systems, a unified design methodology at higher levels of abstraction becomes a necessity. Meanwhile, it is also important to incorporate the current design practice emphasizing IP reuse at various abstraction levels. In the traditional design industry, there are many legacy IPs at the register transfer level or gate level. To handle the design complexity, people are now moving towards higher abstraction levels, such as the transaction level or behavior level. However, the abstraction gap prohibits easy communication and synchronization in IP integration. Another challenge is the co-simulation among IPs written in different design languages. Up to now, there are attempts on co-simulation between HDLs and C/C++/SystemC; however, there does not exist a generic co-simulation framework for arbitrary design languages. In this paper, we present a communication infrastructure for an integrated design framework that enables co-design and co-simulation of heterogeneous design components specified at different abstraction levels and in different languages. The core of the approach is to abstract different communication interfaces or protocols to a common high level communication semantics. Designers only need to specify the interfaces of the design components using extended regular expressions; communication adapters can then be automatically generated for the co-simulation or other co-design and co-verification purposes.

**[46] Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks**

*Elaine Cheong, Prof. Edward A. Lee, Yang Zhao. Technical report, EECS Dept. UC Berkeley, 15, February, 2006; Presented in conjunction with BEARS 2006.*

We describe Viptos (Visual Ptolemy and TinyOS), an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. TinyOS is a component-based, event-driven runtime environment designed for wireless sensor networks. Viptos allows networked embedded systems developers to construct block and arrow diagrams to create TinyOS programs from any standard library of TinyOS components written in nesC, a C-based programming language. Viptos automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target platform. Viptos is built on Ptolemy II, a modeling and simulation environment for embedded systems, and TOSSIM, an interrupt-level discrete event simulator for homogeneous TinyOS networks. In particular, Viptos includes the full capabilities of VisualSense, a Ptolemy II environment that can model communication channels, networks, and non-TinyOS nodes.

Viptos extends the capabilities of TOSSIM to allow simulation of heterogeneous networks. Viptos provides a bridge between VisualSense and TOSSIM by providing interrupt-level simulation of actual TinyOS programs, with packet-level simulation of the network, while allowing the developer to use other models of computation available in Ptolemy II for modeling the physical environment and other parts of the system. This framework allows application developers to easily transition between high-level simulation of algorithms to low-level implementation and simulation. This paper presents our experiences with integrating the nesC/TinyOS/TOSSIM and Ptolemy II programming and execution models.

**[47] Strategy Improvement and Randomized Subexponential Algorithms for Stochastic Parity Games**

*Krishnendu Chatterjee and Thomas A. Henzinger. STACS 06, February, 2006.*

A stochastic graph game is played by two players on a game graph with probabilistic transitions. We consider stochastic graph games with \omega-regular winning conditions specified as parity objectives. These games lie in NP and coNP. We present a strategy improvement algorithm for stochastic parity games; this is the first non-brute-force algorithm for solving these games. From the strategy improvement algorithm we obtain a randomized subexponential-time algorithm to solve such games.

**[48] Markov Decision Processes with Multiple Objectives**

*Krishnendu Chatterjee, Rupak Majumdar and Thomas A. Henzinger. STACS, February, 2006.*

We consider Markov decision processes (MDPs) with multiple discounted reward objectives. Such MDPs occur in design problems where one wishes to simultaneously optimize several criteria, for example, latency and power. The possible trade-offs between the different objectives are characterized by the Pareto curve. We show that every Pareto-optimal point can be achieved by a memoryless strategy; however, unlike in the single-objective case, the memoryless strategy may require randomization. Moreover, we show that the Pareto curve can be approximated in polynomial time in the size of the MDP. Additionally, we study the problem if a given value vector is realizable by any strategy, and show that it can be decided in polynomial time; but the question whether it is realizable by a deterministic memoryless strategy is NP-complete. These results provide efficient algorithms for design exploration in MDP models with multiple objectives.

**[49] A Constructive Fixed-Point Theorem and the Feedback Semantics of Timed Systems**

*James Adam Cataldo, Edward A. Lee, Xiaojun Liu, Eleftherios Dimitrios Matsikoudis and Haiyang Zheng. Technical report, EECS Dept. University of California, Berkeley, 4, January, 2006.*

Deterministic timed systems can be modeled as fixed point problems. In particular, any connected network of timed systems can be modeled as a single system with feedback, and the system behavior is the fixed point of the corresponding system equation, when it exists. For delta-causal systems, we can use the Cantor metric to measure the distance

between signals and the Banach fixed-point theorem to prove the existence and uniqueness of a system behavior. Moreover, the Banach fixed-point theorem is constructive: it provides a method to construct the unique fixed point through iteration. In this paper, we extend this result to systems modeled with the superdense model of time used in hybrid systems. We call the systems we consider eventually delta-causal, a strict generalization of delta-causal in which multiple events may be generated on a signal in zero time. With this model of time, we can use a generalized ultrametric instead of a metric to model the distance between signals. The existence and uniqueness of behaviors for such systems comes from the fixed-point theorem of Priess-Crampe, but this theorem gives no constructive method to compute the fixed point. This leads us to define petrics, a generalization of metrics, which we use to generalize the Banach fixed-point theorem to provide a constructive fixed-point theorem. This new fixed-point theorem allows us to construct the unique behavior of eventually delta-causal systems.

**[50] The Problem with Threads**

*Edward A. Lee. Technical report, EECS Dept., University of California, Berkeley, 1, January, 2006.*

Threads are a seemingly straightforward adaptation of the dominant sequential model of computation to concurrent systems. Languages require little or no syntactic changes to support threads, and operating systems and architectures have evolved to efficiently support them. Many technologists are pushing for increased use of multithreading in software in order to take advantage of the predicted increases in parallelism in computer architectures. In this paper, I argue that this is not a good idea. Although threads seem to be a small step from sequential computation, in fact, they represent a huge step. They discard the most essential and appealing properties of sequential computation: understandability, predictability, and determinism. Threads, as a model of computation, are wildly nondeterministic, and the job of the programmer becomes one of pruning that nondeterminism. Although many research techniques improve the model by offering more effective pruning, I argue that this is approaching the problem backwards. Rather than pruning nondeterminism, we should build from essentially deterministic, composable components. Nondeterminism should be explicitly and judiciously introduced where needed, rather than removed where not needed. The consequences of this principle are profound. I argue for the development of concurrent coordination languages based on sound, composable formalisms. I believe that such languages will yield much more reliable, and more concurrent programs.

**[51] HyVisual: A Hybrid System Modeling Framework Based on Ptolemy II**

*Edward A. Lee and Haiyang Zheng. IFAC Conference on Analysis and Design of Hybrid Systems, January, 2006;*

HyVisual is a hybrid systems modeling framework providing a block diagram visual syntax for specifying continuous dynamics and a bubble-and-arc syntax for specifying modal behavior. It is based on Ptolemy II, is written in Java, and is distributed open-source at http://ptolemy.eecs.berkeley.edu/hyvisual/. HyVisual has a rigorous operational semantics described in [1]. A key property is that it internally uses superdense time, where signals are modeled as partial functions of the form f:R+ × N * V , where R+ is the

non-negative real numbers and represents time, V is the value set (a data type, such as Rn), and N is the set of natural numbers. Continuous-time functions are total, whereas discrete-event functions are defined only on a discrete subset of R+. The N in the domain permits signals to have multiple values in a well-defined order at a particular time. Using this framework, HyVisual gives a rigorous semantics to discontinuous signals (which have multiple values at the point of discontinuity), to discrete-event signals with multiple events at the same time, and to transient states, where the time spent in the state is zero. An example of a HyVisual model that leverages this is shown in fig. 1, which shows many features of HyVisual. This models Newton's cradle, an apparatus with three (or more) balls hanging from strings (inspired by a one dimensional version in [2]). If the model is initialized with one of the balls displaced as shown in the HyVisual graphical animation at the lower left, then when the ball collides with the middle ball, a transient state results. At that time, the right ball transfers its momentum to the middle ball, and then, without any time elapsing, the middle ball transfers its momentum to the left ball. The two events (state transitions in the state machine at the middle left) are simultaneous but ordered. Other initial conditions can be chosen where two simultaneous events are unordered (e.g., starting with two balls appropriately displaced). HyVisual allows a model to permit nondeterministic choice of enabled transitions.

**[52] The Complexity of Quantitative Concurrent Parity Games**

*Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger. SODA, January, 2006.*

We consider two-player infinite games played on graphs. The games are concurrent, in that at each state the players choose their moves simultaneously and independently, and stochastic, in that the moves determine a probability distribution for the successor state. The value of a game is the maximal probability with which a player can guarantee the satisfaction of her objective. We show that the values of concurrent games with \omega-regular objectives expressed as parity conditions can be decided in NP and coNP. This result substantially improves the best known previous bound of 3EXPTIME. It also shows that the full class of concurrent parity games is no harder than the special case of turn-based stochastic reachability games, for which NP and coNP is the best known bound. While the previous, more restricted NP and coNP results for graph games relied on the existence of particularly simple (pure memoryless) optimal strategies, in concurrent games with parity objectives optimal strategies may not exist, and e-optimal strategies (which achieve the value of the game within a parameter e > 0) require in general both randomization and infinite memory. Hence our proof must rely on a more detailed analysis of strategies and, in addition to the main result, yields two results that are interesting on their own. First, we show that there exist e-optimal strategies that in the limit coincide with memoryless strategies; this parallels the celebrated result of Mertens-Neyman for concurrent games with limit-average objectives. Second, we complete the characterization of the memory requirements for e-optimal strategies for concurrent games with parity conditions, by showing that memoryless strategies suffice for e-optimality for coBÃ¼chi conditions.

**[53] Games with Secure Equilibria**

*Krishnendu Chatterjee, Thomas A. Henzinger and Marcin Jurdzinski. Theoretica Computer Science, January 2006.*

In 2-player non-zero-sum games, Nash equilibria capture the options for rational behavior if each player attempts to maximize her payoff. In contrast to classical game theory, we consider lexicographic objectives: first, each player tries to maximize her own payoff, and then, the player tries to minimize the opponent's payoff. Such objectives arise naturally in the verification of systems with multiple components. There, instead of proving that each component satisfies its specification no matter how the other components behave, it sometimes suffices to prove that each component satisfies its specification provided that the other components satisfy their specifications. We say that a Nash equilibrium is secure if it is an equilibrium with respect to the lexicographic objectives of both players. We prove that in graph games with Borel winning conditions, which include the games that arise in verification, there may be several Nash equilibria, but there is always a unique maximal payoff profile of a secure equilibrium. We show how this equilibrium can be computed in the case of \omega-regular winning conditions, and we characterize the memory requirements of strategies that achieve the equilibrium.

**[54] Ellipsoidal Toolbox**

*Alex A. Kurzhanskiy, Pravin Varaiya. Technical report, EECS, UC Berkeley, January, 2006.*

Ellipsoidal Toolbox (ET) implements in MATLAB the ellipsoidal calculus and its application to the reachability analysis of continuous- and discrete-time, possibly time-varying linear systems, and linear systems with disturbances, for which ET calculates both open-loop and close-loop reach sets. The ellipsoidal calculus provides the following benefits: - The complexity of the ellipsoidal representation is quadratic in the dimension of the state space, and linear in the number of time steps. - It is possible to exactly represent the reach set of linear system through both external and internal ellipsoids. - It is possible to single out individual external and internal approximating ellipsoids that are optimal to some given criterion (e.g. trace, volume, diameter), or combination of such criteria. - It gives simple analytical expressions for the control that steers the state to a desired target.

**[55] Languages and Tools for Hybrid Systems Design**

*Luca P. Carloni, Roberto Passerore, Alessandro Pinto and Alberto Sangiovanni-Vincentelli. Foundations and Trends in Design Automation, 1(1):1-204, January 2006.*

The explosive growth of embedded electronics is bringing information and control systems of increasing complexity to every aspects of our lives. The most challenging designs are safety-critical systems, such as transportation systems (e.g., airplanes, cars, and trains), industrial plants and health care monitoring. The difficulties reside in accommodating constraints both on functionality and implementation. The correct behavior must be guaranteed under diverse states of the environment and potential failures; implementation has to meet cost, size, and power consumption requirements. The design is therefore subject to extensive mathematical analysis and simulation. However, traditional models of information systems do not interface well to the

continuous evolving nature of the environment in which these devices operate. Thus, in practice, different mathematical representations have to be mixed to analyze the overall behavior of the system. Hybrid systems are a particular class of mixed models that focus on the combination of discrete and continuous subsystems. There is a wealth of tools and languages that have been proposed over the years to handle hybrid systems. However, each tool makes different assumptions on the environment, resulting in somewhat different notions of hybrid system. This makes it difficult to share information among tools. Thus, the community cannot maximally leverage the substantial amount of work that has been directed to this important topic. In this paper, we review and compare hybrid system tools by highlighting their differences in terms of their underlying semantics, expressive power and mathematical mechanisms. We conclude our review with a comparative summary, which suggests the need for a unifying approach to hybrid systems design. As a step in this direction, we make the case for a semantic-aware interchange format, which would enable the use of joint techniques, make a formal comparison between different approaches possible, and facilitate exporting and importing design representations.

**[56] A-Priori Detection of Zeno Behavior in Communication Networks Modeled as Hybrid Systems**

*A. Abate, A. D. Ames, and Shankar S. Sastry. Proc. 25th IEEE American Control Conference, Minneapolis, MN, Jun. 2006.*

(No abstract.)

**[57] Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems**

*A. Abate, A. D. Ames, and S. Sastry. Proc. 25th IEEE American Control Conference, Minneapolis, MN, Jun. 2006.*

This paper introduces, develops and discusses an integration-inspired methodology for the simulation and analysis of deterministic hybrid dynamical systems. When simulating hybrid systems, and thus unavoidably introducing some numerical error, a progressive tracking of this error can be exploited to discern the properties of the system, i.e., it can be used to introduce a stochastic approximation of the original hybrid system, the simulation of which would give a more complete representation of the possible trajectories of the system. Moreover, the error can be controlled to check and even guarantee (in certain special cases) the robustness of simulated hybrid trajectories.

**[58] Stochastic Approximations of Hybrid Systems**

*A. Abate, A. D. Ames, and S. S. Sastry. Proc. 24th IEEE American Control Conference, Portland, OR, 2005, Jun., 2005.*

This paper introduces a method for approximating the dynamics of deterministic hybrid systems. Within this setting, we shall consider jump conditions that are characterized by spatial guards. After defining proper penalty functions along these deterministic guards, corresponding probabilistic intensities are introduced and the deterministic dynamics are approximated by the stochastic evolution of a continuous-time Markov process. We will

illustrate how the definition of the stochastic barriers can avoid ill-posed events such as "grazing," and show how the probabilistic guards can be helpful in addressing the problem of event detection. Furthermore, this method represents a very general technique for handling Zeno phenomena; it provides a universal way to regularize a hybrid system. Simulations will show that the stochastic approximation of a hybrid system is accurate, while being able to handle "pathological cases." Finally, further generalizations of this approach are motivated and discussed.

**[59] Hybrid Lagrangian and Hamiltonian Reduction of Simple Hybrid Systems**

*A. D. Ames and S. Sastry. Unpublished article, January, 2006.*

This paper extends Lagrangian and Hamiltonian reduction to a hybrid setting, i.e., to systems that display both continuous and discrete behavior. We begin by considering Lagrangians and simple mechanical systems together with unilateral constraints on the set of admissible configurations; this naturally yields the notion of a hybrid Lagrangian and a simple hybrid mechanical system. From such systems we obtain, explicitly, simple hybrid systems. We give conditions on when it is possible to reduce a hybrid system obtained from a cyclic hybrid Lagrangian, and we explicitly compute the reduced hybrid system---we perform hybrid Routhian (or Lagrangian) reduction. We then turn our attention to a more general form of reduction: hybrid Hamiltonian reduction. The main result of this paper is explicit conditions on when it is possible to reduce the phase space of simple hybrid systems due to symmetries in the system; given a Hamiltonian G-space (which is the ingredient needed to reduce a continuous system), we find conditions on the hybrid system and G-space so that reduction can be carried out in a hybrid setting.

**[60] On the Partitioning of Syntax and Semantics For Hybrid Systems Tools**

*Jonathan Sprinkle, Aaron D. Ames, Alessandro Pinto, Haiyang Zheng, S. Shankar Sastry. 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), IEEE Controls Society, 4694-4699, December, 2005.*

Interchange formats are notoriously difficult to finish. That is, once one is developed, it is highly nontrivial to prove (or disprove) generality, and difficult at best to gain acceptance from all major players in the application domain. This paper addresses such a problem for hybrid systems, but not from the perspective of a tool interchange format, but rather that of tool availability in a toolbox. Through the paper we explain why we think this is a good approach for hybrid systems, and we also analyze the domain of hybrid systems to discern the semantic partitions that can be formed to yield a classification of tools based on their semantics. These discoveries give us the foundation upon which to build semantic capabilities, and to guarantee operational interaction between tools based on matched operational semantics.

**[61] Semantic Foundation of the Tagged Signal Model**

*Xiaojun Liu. Semantic Foundation of the Tagged Signal Model. University of California, Berkeley, December, 2005.*

The tagged signal model provides a denotational framework to study properties of various models of computation. It is a generalization of the Signals and Systems approach

to system modeling and specification. Having different models of computation or aspects of them specified in the tagged signal model framework provides the following opportunities. First, one can compare certain properties of the models of computation, such as their notion of synchrony. Such comparisons highlight both the differences and the commonalities among the models of computation. Second, one can define formal relations among signals and process behaviors from different models of computation. These relations have important applications in the specification and design of heterogeneous embedded systems. Third, it facilitates the cross-fertilization of results and proof techniques among models of computation. This opportunity is exploited extensively in this dissertation. The main goal of this dissertation is to establish a semantic foundation for the tagged signal model. Both order-theoretic and metric-theoretic concepts and approaches are used. The fundamental concepts of the tagged signal model--signals, processes, and networks of processes--are formally defined. From few assumptions on the tag sets of signals, it is shown that the set of all signals with the same partially ordered tag set and the same value set is a complete partial order. This leads to a direct generalization of Kahn process networks to tagged process networks. Building on this result, the order-theoretic approach is further applied to study timed process networks, in which all signals share the same totally ordered tag set. The order structure of timed signals provides new characterizations of the common notion of causality and the discreteness of timed signals. Combining the causality and the discreteness conditions is proved to guarantee the non-Zenoness of timed process networks. The metric structure of tagged signals is studied from the very specific--the Cantor metric and its properties. A generalized ultrametric on tagged signals is proposed, which provides a framework for defining more specialized metrics, such as the extension of the Cantor metric to super-dense time. The tagged signal model provides not only a framework for studying the denotational semantics of models of computation, but also useful constructs for studying implementations or simulations of tagged processes. This is demonstrated by deriving certain properties of two discrete event simulation strategies from the behavioral specifications of discrete event processes. A formulation of tagged processes as labeled transition systems provides yet another framework for comparing different implementation or simulation strategies for tagged processes. This formulation lays the foundation to future research in polymorphic implementations of tagged processes.

### [62] Semi-perfect Information Games

*Krishnendu Chatterjee and Thomas A. Henzinger. FSTTCS, December, 2005.*

Much recent research has focused on the applications of games with \omega-regular objectives in the control and verification of reactive systems. However, many of the game-based models are ill-suited for these applications, because they assume that each player has complete information about the state of the system (they are perfect-information games). This is because in many situations, a controller does not see the private state of the plant. Such scenarios are naturally modeled by partial-information games. On the other hand, these games are intractable; for example, partial-information games with simple reachability objectives are 2EXPTIME-complete. We study the intermediate case of semiperfect-information games, where one player has complete knowledge of the state, while the other player has only partial knowledge. This model is

appropriate in control situations where a controller must cope with plant behavior that is as adversarial as possible, i.e., the controller has partial information while the plant has perfect information. As is customary, we assume that the controller and plant take turns to make moves. We show that these semiperfect-information turn-based games are equivalent to perfect-information concurrent games, where the two players choose their moves simultaneously and independently. Since the perfect-information concurrent games are well-understood, we obtain several results of how semiperfect-information turn-based games differ from perfect-information turn-based games on one hand, and from partial-information turn-based games on the other hand. In particular, semiperfect-information turn-based games can benefit from randomized strategies while the perfect-information variety cannot, and semiperfect-information turn-based games are in NP and coNP for all parity objectives.

**[63] Trading End-to-End Latency for Composability**

*Slobodan Matic, Thomas Henzinger. Proceedings of RTSS 2005, 99-110, December, 2005.*

The periodic resource model for hierarchical, compositional scheduling abstracts task groups by resource requirements. We study this model in the presence of dataflow constraints between the tasks within a group (intragroup dependencies), and between tasks in different groups (intergroup dependencies). We consider two natural semantics for dataflow constraints, namely, RTW (Real-Time Workshop) semantics and LET (logical execution time) semantics. We show that while RTW semantics offers better end-to-end latency on the task group level, LET semantics allows tighter resource bounds in the abstraction hierarchy and therefore provides better composability properties. This result holds both for intragroup and intergroup dependencies, as well as for shared and for distributed resources.

**[64] Sufficient Conditions for the Existence of Zeno Behavior**

*A. D. Ames, A. Abate and S. Sastry. IEEE Conference on Decision and Control, December, 2005.*

In this paper, sufficient conditions for the existence of Zeno behavior in a class of hybrid systems are given; these are the first sufficient conditions on Zeno of which the authors are aware for hybrid systems with nontrivial dynamics. This is achieved by considering a class of hybrid systems termed {\em diagonal first quadrant (DFQ) hybrid systems}. When the underlying graph of a DFQ hybrid system has a cycle, we can construct an infinite execution for this system when the vector fields on each domain satisfy certain assumptions. To this execution, we can associate a single discrete time dynamical system that describes its continuous evolution. Therefore, we reduce the study of executions of DFQ hybrid systems to the study of a single discrete time dynamical system. We obtain sufficient conditions for the existence of Zeno by determining when this discrete time dynamical system is exponentially stable.

**[65] On the Stability of Zeno Equilibria**

*D. Ames, P. Tabuada and S. Sastry, 34-48, Lecture Notes in Com, 3927, Springer-Verlag, 2006.*

Zeno behaviors are one of the (perhaps unintended) features of many hybrid models of physical systems. They have no counterpart in traditional dynamical systems or automata

theory and yet they have remained relatively unexplored over the years. In this paper we address the stability properties of a class of Zeno equilibria, and we introduce a necessary paradigm shift in the study of hybrid stability. Motivated by the peculiarities of Zeno equilibria, we consider a form of asymptotic stability that is global in the continuous state, but local in the discrete state. We provide sufficient conditions for stability of these equilibria, resulting in sufficient conditions for the existence of Zeno behavior.

### [66] Viptos 5.1-alpha

*Elaine Cheong, Christopher Brooks, Edward A. Lee. Viptos 5.1-alpha. UC Berkeley, 1, November, 2005.*

Viptos is an interface between TinyOS and Ptolemy II. TinyOS is an event-driven operating system designed for sensor network nodes that have very limited resources (e.g., 8K bytes of program memory, 512 bytes of RAM). TinyOS, is used, for example, on the Berkeley MICA motes, which are small wireless sensor nodes. The Viptos5.1-alpha release is a source only release that works under Linux only. Under Windows, Viptos will not run TinyOS models, though the models are viewable.

### [67] Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks

*Elaine Cheong, Edward A. Lee, and Yang Zhao. Proceedings of the Third ACM Conference on Embedded Networked Sensory Systems, ACM, November, 2005.*

We are announcing the first release of Viptos (Visual Ptolemy and TinyOS), an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. Viptos allows developers to create block and arrow diagrams to construct TinyOS programs from any standard library of nesC/TinyOS components. The tool automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target hardware. In particular, Viptos includes the full capabilities of VisualSense [1], which can model communication channels, networks, and non-TinyOS nodes. This release of Viptos is compatible with nesC 1.2 and includes tools to harvest existing TinyOS components and applications and convert them into a format that can be displayed as block (and arrow) diagrams and simulated.

### [68] A Simplified Catalytic Converter Model for Automotive Coldstart Control Applications

*Pannag R Sanketi, J. K. Hedrick, Tomoyuki Kaga. Proceedings of 2005 ASME International Mechanical Engineering Congress and Exposition (IMECE2005), November, 2005; Orlando, Florida USA.*

More than three-fourths of the unburned hydrocarbon (HC) emissions in a typical drive cycle of an automotive engine are produced in the initial 2 minutes of operation, commonly known as the coldstart period. Catalyst light-off plays a very important role in reducing these emissions. Model-based paradigm is used to develop a control-oriented, thermodynamics based simple catalyst model for coldstart purposes. It is a modified version of an available model consisting of thermal dynamics and static efficiency maps,

the critical modification being in the thermal submodel. Oxygen storage phenomenon does not play a significant role during the warm-up of the engine. The catalyst is modeled as a second-order system consisting of catalyst brick temperature and temperature of the feedgas flowing through the catalyst as its states. Energy balance of an unsteady flow through a control volume is used to model the feedgas temperature, whereas energy balance of a closed system is used to model the catalyst brick temperature. Wiebe profiles are adopted to empirically model the HC emissions conversion properties of the catalyst as a function of the catalyst temperature and the air-fuel ratio. The static efficiency maps are further extended to include the effects of spatial velocity of the feedgas. Experimental results indicate good agreement with the model estimates for the catalyst warm-up. It is shown that the model represents the system more accurately as compared to the previous model on which it is based and offers a broader scope for analysis.

**[69] HyVisual 5.0.1**

*Haiyang Zheng, Christopher Brooks, Edward Lee, Jie Liu, Xiaojun Liu, Stephen Neuendorffer. UC Berkeley, 7, October, 2005.*

Hybrid systems are systems with continuous-time dynamics, discrete events, and discrete mode changes. HyVisual supports construction of hierarchical hybrid systems. It uses a block-diagram representation of ordinary differential equations (ODEs) to define continuous dynamics. It uses a bubble-and-arc diagram representation of finite state machines to define discrete behavior.

**[70] Ptolemy II 5.0.1**

*Christopher Brooks, Edward Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao, Haiyang Zheng, Gang Zhou, Ye Zhou. UC Berkeley, 5, October, 2005.*

Ptolemy II 5.0.1 includes

- A Dynamic Dataflow (DDF) domain, in which actors are fired in response to available input data.
- Modeling of Hybrid systems. Hybrid systems are a special case of modal models where finite-state machines (FSMs) are combined with the continuous-time (CT) models to get mixed continuous-time and discrete-event models.
- Stochastic hybrid systems, which add random behavior to continuous-time models mixed with discrete events.
- Heterochronous Dataflow (HDF), which is an extension of synchronous dataflow (SDF) that permits dynamically changing production and consumption patterns without sacrificing static scheduling.

Ptolemy II 5.0.1 includes the following changes since Ptolemy II 5.0:

- Fixed problem with selecting different user styles for parameters.
- shutdown.bat script removed from a demo directory. For details see Bat/sdwn3 Virus Warning
- Fixed a problem surround saving in a library. See the Limitations page.
- Minor documentation updates

**[71] Building Unreliable Systems out of Reliable Components: The Real Time Story**

*Technical report, Edward A. Lee. Technical report, EECS Dept., University of California, Berkeley, 5, October, 2005.*

Despite considerable progress in software and hardware techniques, when embedded computing systems absolutely must meet tight timing constraints, many of the advances in computing become part of the problem rather than part of the solution. The underlying technology for computation, synchronous digital logic, easily delivers precise timing determinacy (although certain deep submicron techniques threaten even this foundation). However, advances in computer architecture and software have made it difficult or impossible to estimate or predict the execution time of software. Moreover, networking techniques introduce variability and stochastic behavior, and operating systems rely on best effort techniques. Worse, programming languages lack time in their semantics, so timing requirements are only specified indirectly. I examine the following question: "if precise timeliness in a networked embedded system is absolutely essential, what has to change?" The answer, unfortunately, is "nearly everything."

**[72] 5th OOPSLA Workshop on Domain-Specific Modeling (DSM'05)**

*Juha-Pekka Tolvanen, Jonathan Sprinkle, Matti Rossi, Computer Science and Information System Reports, Technical Reports, TR-36, University of Jyväskylä, Finland, October, 2005.*

Domain-Specific Modeling aims at raising the level of abstraction beyond programming by specifying the solution directly using domain concepts. In a number of cases the final products can be generated from these high-level specifications. This automation is possible because of domain-specificity: both the modeling language and code generators fit to the requirements of a narrow domain only, often in a single company. This is the fifth workshop on Domain-Specific Modeling, following the encouraging experiences from the earlier workshops at past OOPSLA conferences (Tampa 2001, Seattle 2002, Anaheim 2003 and Vancouver 2004). During the time the DSM workshops have been organized, interest in domain-specific modeling languages, metamodeling and supporting tools has seen a revival. Today DSM approaches gain popularity and they are used by large software development organizations. Furthermore, development environments for DSM have been deployed by key tool manufacturers, especially Microsoft and IBM. The objective of this workshop series is to bring together practitioners and researchers on the field of DSM to discuss and share experiences, present new ideas on modeling and tools. The workshop follows the structure found effective during the past workshops: presentations of selected papers in the morning and early afternoon and group work and its reporting in the late afternoon. This year the papers are organized into three themes: cases of DSM language creation and use, DSM for special domains and foundations of DSM. Together all these contributions form a basis for fruitful discussions on creation, use and refinement of DSM and supporting tools. The electronic version of the proceedings, presentation slides and group work results is available at www.dsmforum.org/events. We thank our program committee who donated their time and energy to review the papers. We hope you find the results of DSM’05 beneficial and enjoyable.

**[73] Verifying Quantitative Properties Using Bound Functions**

*Arindam Chakrabarti, Krishnendu Chatterjee, Thomas A. Henzinger, Orna Kupferman and Rupak Majumdar. CHARME, 50--64, October, 2005.*

We define and study a quantitative generalization of the traditional boolean framework of model-based specification and verification. In our setting, propositions have integer values at states, and properties have integer values on traces. For example, the value of a quantitative proposition at a state may represent power consumed at the state, and the value of a quantitative property on a trace may represent energy used along the trace. The value of a quantitative property at a state, then, is the maximum (or minimum) value achievable over all possible traces from the state. In this framework, model checking can be used to compute, for example, the minimum battery capacity necessary for achieving a given objective, or the maximal achievable lifetime of a system with a given initial battery capacity. In the case of open systems, these problems require the solution of games with integer values. Quantitative model checking and game solving is undecidable, except if bounds on the computation can be found. Indeed, many interesting quantitative properties, like minimal necessary battery capacity and maximal achievable lifetime, can be naturally specified by quantitative-bound automata, which are finite automata with integer registers whose analysis is constrained by a bound function f that maps each system K to an integer f(K). Along with the linear-time, automaton-based view of quantitative verification, we present a corresponding branching-time view based on a quantitative-bound Âμ-calculus, and we study the relationship, expressive power, and complexity of both views.

**[74] Homogeneous Semantics Preserving Deployments of Heterogeneous Networks of Embedded Systems**

*A. D. Ames, A. Sangiovanni-Vincentelli and S. Sastry. Workshop on Networked Embedded Sensing and Control, October, 2005.*

Tagged systems provide a denotational semantics for embedded systems. A heterogeneous network of embedded systems can be modeled mathematically by a network of tagged systems. Taking the heterogeneous composition of this network results in a single, homogeneous, tagged system. The question this paper addresses is: when is semantics (behavior) preserved by composition? To answer this question, we use the framework of category theory to reason about heterogeneous system composition and derive results that are as general as possible. In particular, we define the category of tagged systems, demonstrate that a network of tagged systems corresponds to a diagram in this category and prove that taking the composition of a network of tagged systems is equivalent to taking the limit of this diagram---thus composition is endowed with a universal property. Using this universality, we are able to derive verifiable necessary and sufficient conditions on when composition preserves semantics.

### [75] Online Safety Calculations for Glideslope Recapture

*Jonathan Sprinkle, Aaron D. Ames, J. Mikael Eklund, Ian Mitchell, S. Shankar Sastry. Innovations in Systems and Software Engineering, 1(2):157-175, September 2005; This was an invited paper, and was published without peer review.*

As unmanned aerial vehicles (UAVs) increase in popularity and usage, an appropriate increase in confidence in their behavior is expected. This research addresses a particular portion of the flight of an aircraft (whether autonomous, unmanned, or manned): specifically, the recapture of the glide slope after a wave-off maneuver during landing. While this situation is rare in commercial aircraft, its applicability toward unmanned aircraft has been limited due to the complexity of the calculations of safety of the maneuvers. In this paper, we present several control laws for this glide-slope recapture, and inferences into their convergence to the glide slope, as well as reachability calculations which show their guaranteed safety. We also present a methodology which theoretically allows us to apply these offline-computed safety data to all kinds of unmanned fixed-wing aerial vehicles while online, permitting the use of the controllers to reduce wait times during landing. Finally, we detail the live aircraft application demonstration which was done to show feasibility of the controller, and give the results of offline simulations which show the correctness of online decisions at that demonstration.

### [76] Using Dependent Types to Certify the Safety of Assembly Code

*Matthew Harren and George C. Necula. Static Analysis Symposium (SAS), Springer-Verlag LNCS, 155-170, September, 2005; Available at http://www.cs.berkeley.edu/~matth/papers/sas05.pdf.*

There are many source-level analyses or instrumentation tools that enforce various safety properties. In this paper we present an infrastructure that can be used to check independently that the assembly output of such tools has the desired safety properties. By working at assembly level we avoid the complications with unavailability of source code, with source-level parsing, and we certify the code that is actually deployed. The novel feature of the framework is an extensible dependently-typed framework that supports type inference and mutation of dependent values in memory. The type system can be extended with new types as needed for the source-level tool that is certified. Using these dependent types, we are able to express the invariants enforced by CCured, a source-level instrumentation tool that guarantees type safety in legacy C programs. We can therefore check that the x86 assembly code resulting from compilation with CCured is in fact type-safe.

### [77] Semantics-Based Optimization Across Uncoordinated Tasks in Networked Embedded Systems

*Jie Liu, Elaine Cheong, and Feng Zhao. 5th ACM Conference on Embedded Software (EMSOFT 2005), EMSOFT '05, September, 2005.*

Microservers are networked embedded devices that accept user tasks on demand and execute them on real world information collected by sensors. Sharing intermediate sensing and computing results among these tasks is critical for optimal resource

utilization. This paper presents a service-oriented microserver runtime SHARE and its semantics-based task management design. Event semantics checking and conversion are based on a signal type system (STS) that captures both data values and service triggering. Based on the compatibility of event semantics, redundant computations in uncoordinated tasks are removed from the runtime. A prototype of SHARE has been experimented with a parking garage sensor network executing three uncoordinated user queries.

**[78] Counting Interface Automata and their Application in Static Analysis of Actor Models**

*E. Wandeler, J.W. Janneck, E.A. Lee, L. Thiele. 3rd International Conference on Software Engineering and Formal Methods - SEFB 2005, SEFM 2005, September, 2005.*

We present an interface theory based approach to static analysis of actor models. We first introduce a new interface theory, which is based on Interface Automata, and which is capable of counting with numbers. Using this new interface theory, we can capture temporal and quantitative aspects of an actor interface as well as an actor's token exchange rate. We will show how to extract this information from actors written in the Cal Actor Language (Cal), and we also present a method to capture the interface information as well as the structure of dataflow models into an interface automaton. This automaton acts as glue between the automata of all actors in the model, and by successfully composing all actor automata with it, we can prove interface compatibility of all actors with the composition framework. After successful composition, the resulting automaton will contain information that can be used for further static analysis of the composite actor model.

**[79] Information Technology for Assisted Living at Home: Building a Wireless Infrastructure for Assisted Living**

*J. Mikael Eklund, Thomas Risgaard Hansen, Jonathan Sprinkle, S. Shankar Sastry. 27th Annual International Conference of the IEEE Engineering In Medicine and Biology Society (EMBS), 3931-3934, September, 2005; .*

A heterogeneous wireless network to support a Home Health System is presented. This system integrates a set of smart sensors which are designed to provide health and security to the elder citizen living at home. The system facilitates privacy by performing local computation, it supports heterogeneous devices and it provide a platform and initial architecture for exploring the use of sensors with elderly people in the Information Technology for Assisted Living and Home project. The goal of this project is to provide alerts to care givers in the event of an accident or acute illness, and enable remote monitoring by authorized and authenticated care givers.

**[80] Quantifying similarities between timed systems.**

*Thomas A. Henzinger, Rupak Majumdar, and Vinayak Prabhu. Proceedings of the Third International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS), Lecture Notes in Computer Science 3829, Springer, 2005, 226-241, September, 2005.*

We define quantitative similarity functions between timed transition systems that measure the degree of closeness of two systems as a real, in contrast to the traditional

boolean yes/no approach to timed simulation and language inclusion. Two systems are close if for each timed trace of one system, there exists a corresponding timed trace in the other system with the same sequence of events and closely corresponding event timings. We show that timed CTL is robust with respect to our quantitative version of bisimilarity, in particular, if a system satisfies a formula, then every close system satisfies a close formula. We also define a discounted version of CTL over timed systems, which assigns to every CTL formula a real value that is obtained by discounting real time. We prove the robustness of discounted CTL by establishing that close states in the bisimilarity metric have close values for all discounted CTL formulas.

**[81] JPEG Encoding on the Intel MXP5800: A Platform-Based Design Case Study**

*Abhijit Davare, Qi Zhu, John Moondanos, Alberto Sangiovanni-VIncentelli. ESTIMedia 2005: 3rd Workshop on Embedded Systems for Real-time Multimedia, September, 2005.*

Multimedia systems are becoming increasingly complex and concurrent. The Platform-based design (PBD) methodology tackles these issues by recommending the use of formal models, carefully defined abstraction layers and the separation of concerns. Models of computation (MoCs) can be used within this methodology to enable specialized synthesis and verification techniques. In this paper, these concepts are leveraged in an industrial case study: the JPEG encoder application deployed on the Intel MXP5800 imaging processor. The modeling is carried out in the Metropolis design framework. We show that the system-level model using our chosen model of computation allows performance estimation within 5% of the actual implementation. Moreover, the chosen MoC is amenable to automation, which enables future synthesis techniques.

**[82] An automated exploration framework for FPGA-based soft multiprocessor systems**

*Yujia Jin, Nadathur Satish, Kaushik Ravindran, Kurt Keutzer. Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05, ACM Press, 273 - 278, September, 2005.*

FPGA-based soft multiprocessors are viable system solutions for high performance applications. They provide a software abstraction to enable quick implementations on the FPGA. The multiprocessor can be customized for a target application to achieve high performance. Modern FPGAs provide the capacity to build a variety of micro-architectures composed of 20-50 processors, complex memory hierarchies, heterogeneous interconnection schemes and custom co-processors for performance critical operations. However, the diversity in the architectural design space makes it difficult to realize the performance potential of these systems. In this paper we develop an exploration framework to build efficient FPGA multiprocessors for a target application. Our main contribution is a tool based on Integer Linear Programming to explore micro-architectures and allocate application tasks to maximize throughput. Using this tool, we implement a soft multiprocessor for IPv4 packet forwarding that achieves a throughput of 2 Gbps, surpassing the performance of a carefully tuned hand design.

### [83] Causality Interfaces and Compositional Causality Analysis

*Edward A. Lee, Haiyang Zheng, Ye Zhou. Foundations of Interface Technologies (FIT), CONCUR 2005, ENTCS TBD, August, 2005.*

In this paper, we consider concurrent models of computation where ”actors” (components that are in charge of their own actions) communicate by exchanging messages. The interfaces of actors principally consist of “ports,” which mediate the exchange of messages. Actor-oriented architectures contrast with and complement object-oriented models by emphasizing the exchange of data between concurrent components rather than transfer of control. Examples of such models of computation include the classical actor model, synchronous languages, dataflow models, and discrete-event models. Many of these models of computation benefit considerably from having access to causality information about the components. This paper augments the interfaces of such components to include such causality information. It shows how this causality information can be algebraically composed so that compositions of components acquire causality interfaces that are inferred from their components and the interconnections. We illustrate the use of these causality interfaces to statically analyze discrete-event models for uniqueness of behaviors, synchronous models for causality loops, and dataflow models for schedulability.

### [84] Computing Inverse MEG Signals in the Brain

*J. Mikael Eklund, Ruzena Bajcsy, Jonathan Sprinkle, Gregory V. Simpson. 2005 IEEE Computational Systems Bioinformatics Conference, Controlling Complexity, 332-335, August, 2005.*

This paper deals with the complexity of the inverse computation of brain currents from magnetoencephalography (MEG) signals. MEG measures the magnetic field outside the head: in effect, the resultant field from the flow of current inside the brain. We describe our current techniques to perform this inverse computation (called source estimation in much of the literature), which provides a view of brain activity that is less sensitive to disturbances which affect other kinds of brain activity measurements, though much more expensive to record.

### [85] Two-player Nonzero-sum \omega-Regular Games

*Krishnendu Chatterjee. Two-player Nonzero-sum \omega-Regular Games. CONCUR, August, 2005.*

We study infinite stochastic games played by two-players on a finite graph with goals specified by sets of infinite traces. The games are concurrent (each player simultaneously and independently chooses an action at each round), stochastic (the next state is determined by a probability distribution depending on the current state and the chosen actions), infinite (the game continues for an infinite number of rounds), nonzero-sum (the playersâ€™ goals are not necessarily conflicting), and undiscounted. We show that if each player has an \omega-regular objective expressed as a parity objective, then there exists an \epsilon-Nash equilibrium, for every \epsilon>0. However, exact Nash equilibria need not exist. We study the complexity of finding values (payoff profile) of an \epsilon-Nash equilibrium. We show that the values of an \epsilon-Nash equilibrium in nonzero-

sum concurrent parity games can be computed by solving the following two simpler problems: computing the values of zero-sum (the goals of the players are strictly conflicting) concurrent parity games and computing \epsilon-Nash equilibrium values of nonzero-sum concurrent games with reachability objectives. As a consequence we establish that values of an \epsilon-Nash equilibrium can be computed in TFNP (total functional NP), and hence in EXPTIME.

**[86] An Interface Formalism for Web Services**

*Dirk Beyer, Arindam Chakrabarti, Thomas A. Henzinger. Foundations of Interface Technologies (FIT), 2005, August, 2005.*

(No abstract.)

**[87] An FPGA-Based Soft Multiprocessor System for IPv4 Packet Forwarding**

*Kaushik Ravindran, Nadathur Satish, Yujia Jin, Kurt Keutzer. Proceedings of the 15th International Conference on Field Programmable Logic and Applications (FPL-05), 487-492, August, 2005.*

To realize high performance, embedded applications are deployed on multiprocessor platforms tailored for an application domain. However, when a suitable platform is not available, only few application niches can justify the increasing costs of an IC product design. An alternative is to design the multiprocessor on an FPGA. This retains the programmability advantage, while obviating the risks in producing silicon. This also opens FPGAs to the world of software designers. In this paper, we demonstrate the feasibility of FPGA-based multiprocessors for high performance applications. We deploy IPv4 packet forwarding on a multiprocessor on the Xilinx Virtex-II Pro FPGA. The design achieves a 1.8 Gbps throughput and loses only 2.6X in performance (normalized to area) compared to an implementation on the Intel IXP-2800 network processor. We also develop a design space exploration framework using Integer Linear Programming to explore multiprocessor configurations for an application. Using this framework, we achieve a more efficient multiprocessor design surpassing the performance of our hand-tuned solution for packet forwarding.

**[88] PtPlot 5.5**

*Edward A. Lee, Christopher Brooks. PtPlot 5.5. UC Berkeley, 28, July, 2005.*

Ptplot is a 2D signal plotter implemented in Java. Ptplot can be used in a standalone applet or application or used in your own applet or application. Ptplot is available as part of Ptolemy or as a standalone download.

**[89] Ptolemy II 5.0**

*Christopher Brooks, Edward Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao, Haiyang Zheng, Gang Zhou, Ye Zhou. UC Berkeley, 21, July, 2005.*

Ptolemy II is a software framework developed as part of the Ptolemy Project. It is a Java-based component assembly framework with a graphical user interface called Vergil. Vergil itself is a component assembly defined in Ptolemy II.

The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined *models of computation* that govern the interactions between components. A major problem area being addressed is the use of heterogeneous mixtures of models of computation.

The Ptolemy Project web page contains much more information about the project. The work is conducted in the Department of Electrical Engineering and Computer Sciences of the University of California at Berkeley. The project is directed by Prof. Edward Lee. The project is named after Claudius Ptolemaeus, the second century Greek astronomer, mathematician, and geographer.

Ptolemy II 5.0.1 includes

- A Dynamic Dataflow (DDF) domain, in which actors are fired in response to available input data.
- Modeling of Hybrid systems. Hybrid systems are a special case of modal models where finite-state machines (FSMs) are combined with the continuous-time (CT) models to get mixed continuous-time and discrete-event models.
- Stochastic hybrid systems, which add random behavior to continuous-time models mixed with discrete events.
- Heterochronous Dataflow (HDF), which is an extension of synchronous dataflow (SDF) that permits dynamically changing production and consumption patterns without sacrificing static scheduling.

**[90] The Design and Application of Structured Types in Ptolemy II**

*Y. Xiong, E.A. Lee, X. Liu, Y. Zhao, L.C. Zhong. IEEE Int. Conf. on Granular Computing, Grc 2005, July, 2005.*

Ptolemy II is a component-based design and modeling environment. It has a polymorphic type system that supports both the base types and structured types, such as arrays and records. The base type support was reported in [12]. This paper presents the extensions that support structured types. In the base type system, all the types are organized into a type lattice, and type constraints in the form of inequalities can be solved efficiently over the lattice. We take a hierarchical and granular approach to add structured types to the lattice, and extend the format of inequality constraints to allow arbitrary nesting of structured types. We also analyze the convergence of the constraint solving algorithm on an infinite lattice after structured types are added. To show the application of structured types, we present a Ptolemy II model that implements part of the IEEE 802.11 specifications. This model makes extensive use of record types to represent the protocol messages in the system.

**[91] Heterogeneous Concurrent Modeling and Design in Java (Volume 1, Introduction to Ptolemy II)**

*C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Technical report, EECS Dept., UC Berkeley, 21, July, 2005.*

This volume describes how to construct Ptolemy II models for web-based modeling or building applications. The first chapter includes an overview of Ptolemy II software, and a brief description of each of the models of computation that have been implemented. It describes the package structure of the software, and includes as an appendix a brief tutorial on UML notation, which is used throughout the documentation to explain the structure of the software. The second chapter is a tutorial on building models using Vergil, a graphical user interface where models are built pictorially. The third chapter discusses the Ptolemy II expression language, which is used to set parameter values. The next chapter gives an overview of actor libraries. These three chapters, plus one of the domain chapters, will be sufficient for users to start building interesting models in the selected domain. The fifth chapter gives a tutorial on designing actors in Java.The sixth chapter explains MoML, the XML schema used by Vergil to store models. And the seventh chapter, the final one in this part, explains how to construct custom applets. Volume 2 describes the software architecture of Ptolemy II, and volume 3 describes the domains, each of which implements a model of computation.

**[92] Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture)**

*C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Technical report, EECS Dept., UC Berkeley, 22, July, 2005.*

This volume describes the software architecture of Ptolemy II. The first chapter covers the kernel package, which provides a set of Java classes supporting clustered graph topologies for models. Cluster graphs provide a very general abstract syntax for component-based modeling, without assuming or imposing any semantics on the models. The actor package begins to add semantics by providing basic infrastructure for data transport between components. The data package provides classes to encapsulate the data that is transported. It also provides an extensible type system and an interpreted expression language. The graph package provides graph-theoretic algorithms that are used in the type system and by schedulers in the individual domains. The plot package provides a visual data plotting utility that is used in many of the applets and applications. Vergil is the graphical front end to Ptolemy II and Vergil itself uses Ptolemy II to describe its own configuration. Volume 1 gives an introduction to Ptolemy II, including tutorials on the use of the software, and volume 3 describes the domains, each of which implements a model of computation.

**[93] Heterogeneous Concurrent Modling and Design in Java (Volume 3: Ptolemy II Domains)**

*C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Technical report, EECS Dept., UC Berkeley, 23, July, 2005.*

This volume describes Ptolemy II domains. The domains implement models of computation, which are summarized in chapter 1. Most of these models of computation can be viewed as a framework for component-based design, where the framework defines the interaction mechanism between the components. Some of the domains (CSP, DDE, and PN) are thread-oriented, meaning that the components implement Java threads. These can be viewed, therefore, as abstractions upon which to build threaded Java programs. These abstractions are much easier to use (much higher level) than the raw threads and monitors of Java. Others (CT, DE, SDF) of the domains implement their own scheduling between actors, rather than relying on threads. This usually results in much more efficient execution. The Giotto domain, which addresses real-time computation, is not threaded, but has concurrency features similar to threaded domains. The FSM domain is in a category by itself, since in it, the components are not producers and consumers of data, but rather are states. The non-threaded domains are described first, followed by FSM and Giotto, followed by the threaded domains. Within this grouping, the domains are ordered alphabetically (which is an arbitrary choice). Volume 1 is an introduction to Ptolemy II, including tutorials on use of the software, and volume 2 describes the Ptolemy II software architecture.

**[94] VisualSense: Visual Modeling for Wireless and Sensor Network Systems**

*Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao. Technical report, EECS Dept., UC Berkeley, 25, July, 2005.*

VisualSense is a modeling and simulation framework for wireless and sensor networks that builds on and leverages Ptolemy II. Modeling of wireless networks requires sophisticated representation and analysis of communication channels, sensors, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. This modeling framework is designed to support a component-based construction of such models. It supports actor-oriented definition of network nodes, wireless communication channels, physical media such as acoustic channels, and wired subsystems. The software architecture consists of a set of base classes for defining channels and sensor nodes, a library of subclasses that provide certain specific channel models and node models, and an extensible visualization framework. Custom nodes can be defined by subclassing the base classes and defining the behavior in Java or by creating composite models using any of several Ptolemy II modeling environments. Custom channels can be defined by subclassing the WirelessChannel base class and by attaching functionality defined in Ptolemy II models. It is intended to enable the research community to share models of disjoint aspects of the sensor nets problem and to build models that include sophisticated elements from several aspects.

**[95] HyVisual: A Hybrid System Visual Modeler**

*C. Brooks, A. Cataldo, E.A. Lee, J. Liu, X.Liu, S. Neuendorffer, H. Zheng. Technical report, EECS Dept., UC Berkeley, July, 2005.*

The Hybrid System Visual Modeler (HyVisual) is a block-diagram editor and simulator for continuous-time dynamical systems and hybrid systems. Hybrid systems mix continuous-time dynamics, discrete events, and discrete mode changes. This visual modeler supports construction of hierarchical hybrid systems. It uses a block-diagram representation of ordinary differential equations (ODEs) to define continuous dynamics, and allows mixing of continuous-time signals with events that are discrete in time. It uses a bubble-and-arc diagram representation of finite state machines to define discrete behavior driven by mode transitions. In this document, we describe how to graphically construct models and how to interpret the resulting models. HyVisual provides a sophisticated numerical solver that simulates the continuous-time dynamics, and effective use of the system requires at least a rudimentary understanding of the properties of the solver. This document provides a tutorial that will enable the reader to construct elaborate models and to have confidence in the results of a simulation of those models. We begin by explaining how to describe continuous-time models of classical dynamical systems, and then progress to the construction of mixed signal and hybrid systems. The intended audience for this document is an engineer with at least a rudimentary understanding of the theory of continuous-time dynamical systems (ordinary differential equations and Laplace transform representations), who wishes to build models of such systems, and who wishes to learn about hybrid systems and build models of hybrid systems. HyVisual is built on top of Ptolemy II, a framework supporting the construction of such domain-specific tools. See Ptolemy II for more information.

**[96] The Complexity of Stochastic Rabin and Streett Games**

*Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger. ICALP, July, 2005.*

The theory of graph games with \omega-regular winning conditions is the foundation for modeling and synthesizing reactive processes. In the case of stochastic reactive processes, the corresponding stochastic graph games have three players, two of them (System and Environment) behaving adversarially, and the third (Uncertainty) behaving probabilistically. We consider two problems for stochastic graph games: the qualitative problem asks for the set of states from which a player can win with probability 1 (almost-sure winning); the quantitative problem asks for the maximal probability of winning (optimal winning) from each state. We show that for Rabin winning conditions, both problems are in NP. As these problems were known to be NP-hard, it follows that they are NP-complete for Rabin conditions, and dually, coNP-complete for Streett conditions. The proof proceeds by showing that pure memoryless strategies suffice for qualitatively and quantitatively winning stochastic graph games with Rabin conditions. This insight is of interest in its own right, as it implies that controllers for Rabin objectives have simple implementations. We also prove that for every \omega-regular condition, optimal winning strategies are no more complex than almost-sure winning strategies.

**[97] Counterexample-guided Planning**

*Krishnendu Chatterjee, Thomas A. Henzinger, Ranjit Jhala and Rupak Majumdar. UAI, July, 2005.*

Planning in adversarial and uncertain environments can be modeled as the problem of devising strategies in stochastic perfect information games. These games are generalizations of Markov decision processes (MDPs): there are two (adversarial) players, and a source of randomness. The main practical obstacle to computing winning strategies in such games is the size of the state space. In practice therefore, one typically works with abstractions of the model. The difficulty is to come up with an abstraction that is neither too coarse to remove all winning strategies (plans), nor too fine to be intractable. In verification, the paradigm of counterexample-guided abstraction refinement has been successful to construct useful but parsimonious abstractions automatically. We extend this paradigm to probabilistic models (namely, 2\half games and, as a special case, MDPs). This allows us to apply the counterexample-guided abstraction paradigm to the AI planning problem. As special cases, we get planning algorithms for MDPs and deterministic systems that automatically construct system abstractions.

**[98] Dynamic Surface Control of Engine Exhaust Hydrocarbons and Catalyst Temperature for Reduced Coldstart Emissions**

*Pannag R Sanketi, J. Carlos Zavala, J. K. Hedrick. Proc. of International Federation of Automatic Control (IFAC) Conference, July, 2005; Prague, Czech Rep.*

Almost three quarters of the hydrocarbon (HC) emissions emitted by an automobile in a typical drive-cycle are produced during the first three minutes of its operation called the coldstart period. In this paper, we propose a way to decrease cold start emissions. A Model-Based paradigm is used to aid the generation of an efficient controller. The controller is built around a mean value engine model and a simplified catalyst model characterized by thermal dynamics, oxygen storage and static efficiency curves. It is shown that the control of engine-out exhaust gas temperature for faster catalyst light-off could be detrimental to the catalyst. A control scheme comprising engine-out hydrocarbon emissions control and catalyst temperature control through dynamic surface control is developed to reduce the tailpipe emissions. It is shown that reduced tailpipe emissions can be chieved without the risk of damaging the catalyst.

**[99] Implementing and Testing a Nonlinear Model Predictive Tracking Controller for Aerial Pursuit Evasion Games on a Fixed Wing Aircraft**

*J. Mikael Eklund, Jonathan Sprinkle, S. Shankar Sastry. Proceedings of American Control Conference (ACC) 2005, 1509-1514, June, 2005.*

The capability of Unmanned Aerial Vehicles (UAVs) to perform autonomously has not yet been demonstrated, however this is an important step to enable at least limited autonomy in such aircraft to allow then to operate with temporary loss of remote control, or when confronted with an adversary or obstacles for which remote control is insufficient. Such capabilities have been under development through Software Enabled Control (SEC) program and were recently tested in the Capstone Demonstration of that

program. In this paper the final simulation and flight test results are presented for a Non-linear Model Predictive Controller (NMPC) used in evasive maneuvers in three dimensions on a fixed wing UAV for the purposes of pursuit/evasion games with a piloted F-15 aircraft.

**[100] Deciding to Land a UAV Safely in Real Time**

*Jonathan Sprinkle, J. Mikael Eklund, S. Shankar Sastry. Proceedings of American Control Conference (ACC) 2005, 3506-3511, June, 2005.*

The difficulty of autonomous free-flight of a fixed-wing UAV is trivial when compared to that of takeoff and landing. There is an even more marked difference when deciding whether or not a UAV can capture or recapture a certain trajectory, since the answer depends on the operating ranges of the aircraft. A common example of requiring this calculation, from a military perspective, is the determination of whether or not an aircraft can capture a landing trajectory (i.e., glideslope) from a certain initial state (velocity, position, etc.). As state dimensions increase, the time to calculate the decision grows exponentially. This paper describes how we can make this decision at flight time, and guarantee that the decision gives a safe answer before the state changes enough to invalidate the decision. We also describe how the computations should be formulated, and how the partitioning of the state-space can be done to reduce the computation time required. Flight testing was performed with our design, and results are given.

**[101] Mean-Payoff Parity Games**

*Krishnendu Chatterjee, Thomas A. Henzinger and Marcin Jurdzinski. LICS 05, June, 2005.*

Games played on graphs may have qualitative objectives, such as the satisfaction of an \omega-regular property, or quantitative objectives, such as the optimization of a realvalued reward. When games are used to model reactive systems with both fairness assumptions and quantitative (e.g., resource) constraints, then the corresponding objective combines both a qualitative and a quantitative component. In a general case of interest, the qualitative component is a parity condition and the quantitative component is a mean-payoff reward. We study and solve such mean-payoff parity games. We also prove some interesting facts about mean-payoff parity games which distinguish them both from mean-payoff and from parity games. In particular, we show that optimal strategies exist in mean-payoff parity games, but they may require infinite memory.

**[102] Composable Code Generation for Distributed Giotto**

*Thomas Henzinger, Christoph Kirsch, Slobodan Matic. Proceedings of LCTES 2005, 21-30, June, 2005.*

We present a compositional approach to the implementation of hard real-time software running on a distributed platform. We explain how several code suppliers, coordinated by a system integrator, can independently generate different parts of the distributed software. The task structure, interaction, and timing is specified as a Giotto program. Each supplier is given a part of the Giotto program and a timing interface, from which the supplier generates task and scheduling code. The integrator then checks, individually for each supplier, in pseudo-polynomial time, if the supplied code meets its timing specification.

If all checks succeed, then the supplied software parts are guaranteed to work together and implement the original Giotto program. The feasibility of the approach is demonstrated by a prototype implementation.

**[103] Simulation Based Deadlock Analysis for System Level Designs**

*Xi Chen, Abhijit Davare, Harry Hsieh, Alberto Sangiovanni-Vincentelli, Yosinori Watanabe. 42nd Annual Design Automation Conference, 260-265, June, 2005.*

In the design of highly complex, heterogeneous, and concurrent systems, deadlock detection and resolution remains an important issue. In this paper, we systematically analyze the synchronization dependencies in concurrent systems modeled in the Metropolis design environment, where system functions, high level architectures and function-architecture mappings can be modeled and simulated. We propose a data structure called the dynamic synchronization dependency graph, which captures the runtime (blocking) dependencies. A loop-detection algorithm is then used to detect deadlocks and help designers quickly isolate and identify modeling errors that cause the deadlock problems. We demonstrate our approach through a real world design example, which is a complex functional model for video processing and a high level model of function-architecture mapping.

**[104] Implementation of AFR Controller in an Event-driven Real Time Language**

*Arkadeb Ghosal, J. Carlos Zavala J., Marco A. A. Sanvido and J. Karl Hedrick. 2005 American Control Conference, June, 2005.*

The control of emissions has been addressed in the past to comply with environmental regulations. In particular air-to-fuel ratio control is key to reach the allowed pollution levels. The aim of this work is to present an alternative approach which allows for more flexibility to account for the type of signals and requirements of automotive applications, specifically, handling of time and event triggered tasks. An Air-Fuel Ratio nonlinear controller is developed for an automobile engine. The controller is then implemented using the event-driven real-time programming language xGiotto on the OSEK platform provided by WindRiver. Special attention is given to show the advantage that can be gained from using an event driven paradigm for implementing automotive controllers.

**[105] Permissive interfaces**

*Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. Proceedings of the 13th Annual Symposium on Foundations of Software Engineering (FSE), ACM Press, 2005, pp. 31-40.*

A modular program analysis considers components independently and provides a succinct summary for each component, which is used when checking the rest of the system. Consider a system consisting of a library and a client. A temporal summary, or interface, of the library specifies legal sequences of library calls. The interface is safe if no call sequence violates the library's internal invariants; the interface is permissive if it contains every such sequence. Modular program analysis requires full interfaces, which are both safe and permissive: the client does not cause errors in the library if and only if it makes only sequences of library calls that are allowed by the full interface of the library.

Previous interface-based methods have focused on safe interfaces, which may be too restrictive and thus reject good clients. We present an algorithm for automatically synthesizing software interfaces that are both safe and permissive. The algorithm generates interfaces as graphs whose vertices are labeled with predicates over the library's internal state, and whose edges are labeled with library calls. The interface state is refined incrementally until the full interface is constructed. In other words, the algorithm automatically synthesizes a typestate system for the library, against which any client can be checked for compatibility. We present an implementation of the algorithm which is based on the Blast model checker, and we evaluate some case studies.

**[106] Automatic rectangular refinement of affine hybrid systems**

*Laurent Doyen, Thomas A. Henzinger, and Jean-Francois Raskin. Proceedings of the Third International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS), Lecture Notes in Computer Science 3829, Springer, 2005, pp. 144-161.*

We show how to automatically construct and refine rectangular abstractions of systems of linear differential equations. From a hybrid automaton whose dynamics are given by a system of linear differential equations, our method computes automatically a sequence of rectangular hybrid automata that are increasingly precise overapproximations of the original hybrid automaton. We prove an optimality criterion for successive refinements. We also show that this method can take into account a safety property to be verified, refining only relevant parts of the state space. The practicability of the method is illustrated on a benchmark case study.

**[107] Interface-based design**

*Luca de Alfaro and Thomas A. Henzinger. In Engineering Theories of Software-intensive Systems (M. Broy, J. Gruenbauer, D. Harel, and C.A.R. Hoare, eds.), NATO Science Series: Mathematics, Physics, and Chemistry, Vol. 195, Springer, 2005, pp. 83-104.*

We motivate and introduce the theory behind formalizing rich interfaces for software and hardware components. Rich interfaces specify the protocol aspects of component interaction. Their formalization, called interface automata, permits a compiler to check the compatibility of component interaction protocols. Interface automata support incremental design and independent implementability. Incremental design means that the compatibility checking of interfaces can proceed for partial system descriptions, without knowing the interfaces of all components.

Independent implementability means that compatible interfaces can be refined separately, while still maintaining compatibility.

**[108] Model checking discounted temporal properties**

*Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Marielle Stoelinga. Theoretical Computer Science 345:139-170, 2005.*

Temporal logic is two-valued: a property is either true or false. When applied to the analysis of stochastic systems, or systems with imprecise formal models, temporal logic

is therefore fragile: even small changes in the model can lead to opposite truth values for a specification. We present a generalization of the branching-time logic CTL which achieves robustness with respect to model perturbations by giving a quantitative interpretation to predicates and logical operators, and by discounting the importance of events according to how late they occur.

In every state, the value of a formula is a real number in the interval [0,1], where 1 corresponds to truth and 0 to falsehood. The boolean operators and and or are replaced by min and max, the path quantifiers E and A determine sup and inf over all paths from a given state, and the temporal operators F and G specify sup and inf over a given path; a new operator averages all values along a path. Furthermore, all path operators are discounted by a parameter that can be chosen to give more weight to states that are closer to the beginning of the path.

We interpret the resulting logic DCTL over transition systems, Markov chains, and Markov decision processes. We present two semantics for DCTL: a path semantics, inspired by the standard interpretation of state and path formulas in CTL, and a fixpoint semantics, inspired by the mu-calculus evaluation of CTL formulas. We show that, while these semantics coincide for CTL, they differ for DCTL, and we provide model-checking algorithms for both semantics.

**[109] A classification of symbolic transition systems**

*Thomas A. Henzinger, Rupak Majumdar, and Jean-Francois Raskin. ACM Transactions on Computational Logic. 6:1-32, 2005.*

We define five increasingly comprehensive classes of infinite-state systems, called STS1-5, whose state spaces have finitary structure. For four of these classes, we provide examples from hybrid systems.

STS1 These are the systems with finite bisimilarity quotients. They can be analyzed symbolically by iteratively applying predecessor and boolean operations on state sets, starting from a finite number of observable state sets. Any such iteration is guaranteed to terminate in that only a finite number of state sets can be generated. This enables model checking of the mu-calculus.

STS2 These are the systems with finite similarity quotients. They can be analyzed symbolically by iterating the predecessor and positive boolean operations. This enables model checking of the existential and universal fragments of the mu-calculus.

STS3 These are the systems with finite trace-equivalence quotients. They can be analyzed symbolically by iterating the predecessor operation and a restricted form of positive boolean operations (intersection is restricted to intersection with observables). This enables model checking of all omega-regular properties, including linear temporal logic.

STS4 These are the systems with finite distance-equivalence quotients (two states are equivalent if for every distance d, the same observables can be reached in d transitions). The systems in this class can be analyzed symbolically by iterating the predecessor operation and terminating when no new state sets are generated. This enables model checking of the existential conjunction-free and universal disjunction-free fragments of the mu-calculus.

STS5 These are the systems with finite bounded-reachability quotients (two states are equivalent if for every distance d, the same observables can be reached in d or fewer transitions). The systems in this class can be analyzed symbolically by iterating the predecessor operation and terminating when no new states are encountered (this is a weaker termination condition than above). This enables model checking of reachability properties.

**[110] A programmable microkernel for real-time systems.**

*Christoph M. Kirsch, Marco A.A. Sanvido, and Thomas A. Henzinger. Proceedings of the First International Conference on Virtual Execution Environments (VEE), ACM Press, 2005, pp. 35-45.*

We present a new software system architecture for the implementation of hard real-time applications. The core of the system is a microkernel whose reactivity (interrupt handling as in synchronous reactive programs) and proactivity (task scheduling as in traditional RTOSs) are fully programmable. The microkernel, which we implemented on a StrongARM processor, consists of two interacting domain-specific virtual machines, a reactive E (Embedded) machine and a proactive S (Scheduling) machine. The microkernel code (or microcode) that runs on the microkernel is partitioned into E and S code. E code manages the interaction of the system with the physical environment: the execution of E code is triggered by environment interrupts, which signal external events such as the arrival of a message or sensor value, and it releases application tasks to the S machine. S code manages the interaction of the system with the processor: the execution of S code is triggered by hardware interrupts, which signal internal events such as the completion of a task or time slice, and it dispatches application tasks to the CPU, possibly preempting a running task. This partition of the system orthogonalizes the two main concerns of real-time implementations: E code refers to environment time and thus defines the reactivity of the system in a hardware- and scheduler-independent fashion; S code refers to CPU time and defines a system scheduler. If both time lines can be reconciled, then the code is called time safe; violations of time safety are handled again in a programmable way, by run-time exceptions. The separation of E from S code permits the independent programming, verification, optimization, composition, dynamic adaptation, and reuse of both reaction and scheduling mechanisms. Our measurements show that the system overhead is very acceptable even for large sets of task, generally in the 0.2-0.3% range.

**[111] A formal approach to fault tree synthesis for the analysis of distributed fault tolerant systems**

*Mark L. McKelvin Jr, Gabriel Eirea, Claudio Pinello, Sri Kanajan, and Alberto L. Sangiovanni-Vincentelli. In Procs. of EMSOFT, pp. 237-246, 2005.*

Designing cost-sensitive real-time control systems for safety-critical applications requires a careful analysis of both performance versus cost aspects and fault coverage of fault tolerant solutions. This further complicates the difficult task of deploying the embedded software that implements the control algorithms on a possibly distributed execution platform (for instance in automotive applications). In this paper, we present a novel technique for constructing a fault tree that models how component faults may lead to system failure. The fault tree enables us to use existing commercial analysis tools to assess a number of dependability metrics of the system. Our approach is centered on a model of computation, Fault Tolerant Data Flow (FTDF), that enables the integration of formal verification techniques. This new analysis capability is added to an existing design framework, also based on FTDF, that enables a synthesis-based, correct-by-construction, design methodology for the deployment of real-time feedback control systems in safety critical applications.

**[112] Beyond Zeno: Get on with It!**

*Haiyang Zheng, Edward A. Lee and Aaron D. Ames, Joao Hespanha, Ashish Tiwari, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 3927, 2006.*

In this paper we propose a technique to extend the simulation of a Zeno hybrid system beyond its Zeno time point. A Zeno hybrid system model is a hybrid system with an execution that takes an infinite number of discrete transitions during a finite time interval. We argue that the presence of Zeno behavior indicates that the hybrid system model is incomplete by considering some classical Zeno models that incompletely describe the dynamics of the system being modeled. This motivates the systematic development of a method for completing hybrid system models through the introduction of new post-Zeno states, where the completed hybrid system transitions to these post-Zeno states at the Zeno time point. In practice, simulating a Zeno hybrid system is challenging in that simulation effectively halts near the Zeno time point. Moreover, due to unavoidable numerical errors, it is not practical to exactly simulate a Zeno hybrid system. Therefore, we propose a method for constructing approximations of Zeno models by leveraging the completed hybrid system model. Using these approximations, we can simulate a Zeno hybrid system model beyond its Zeno point and reveal the complete dynamics of the system being modeled.

**[113] On the Stability of Zeno Equilibria**

*A. D. Ames, P. Tabuada and S. Sastry, 34-48, Lecture Notes in Com, 3927, Springer-Verlag, 2006.*

Zeno behaviors are one of the (perhaps unintended) features of many hybrid models of physical systems. They have no counterpart in traditional dynamical systems or automata theory and yet they have remained relatively unexplored over the years. In this paper we address the stability properties of a class of Zeno equilibria, and we introduce a necessary

paradigm shift in the study of hybrid stability. Motivated by the peculiarities of Zeno equilibria, we consider a form of asymptotic stability that is global in the continuous state, but local in the discrete state. We provide sufficient conditions for stability of these equilibria, resulting in sufficient conditions for the existence of Zeno behavior.

**[114] CRC Handbook of Dynamic System Modeling**

*Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Aniruddha Gokhale, Sandeep Neema, and Jonathan Sprinkle, Paul A. Fishwick, (in publication), CRC Press, 2006.*

Since the inception of the software industry, modeling tools have been a core product offered by commercial vendors. In this chapter, the essential characteristics of DSM are presented, including a discussion regarding those domains that are most likely to benefit from DSM adoption. The chapter also contains a case study section where two different examples are presented in two different metamodeling tools. An overview of the history of metamodeling tools is also provided, as well as concluding comments.

## 2.3. Project Training and Development

We continue to use the CHESS Software Lab, which is focused on supporting the creation of publication-quality software in support of embedded systems design. The lab is a room with wireless and wired network connections, a large table for collaborative work, a large format printer (used for UML diagrams and poster preparation), comfortable furniture supporting extended hours of collaborative work, a coffee machine, and a library that inherited a collection of software technology books from the Ptolemy Project. This room is used to promote a local version of the Extreme Programming (XP) software design practice, which advocates pair programming, design reviews, code reviews, extensive use of automated regression tests, and a collaboratively maintained body of code (we use CVS). The room began operation in March of 2003 and has been in nearly constant use for collaborative design work. The principal focus of that work has been on advanced tool architectures for hybrid and embedded software systems design.

## 2.4. Outreach Activities

Continuing in our mission to build a modern systems science (MSS) with profound implications on the nature and scope of computer science and engineering research, the structure of computer science and electrical engineering curricula, and future industrial practice. This new systems science must pervade engineering education throughout the undergraduate and graduate levels. Embedded software and systems represent a major departure from the current, separated structure of computer science (CS), computer engineering (CE), and electrical engineering (EE). In fact, the new, emerging systems science reintegrates information and physical sciences. The impact of this change on teaching is profound, and cannot be confined to graduate level.

This year we have continued our work to lay the foundation for a new philosophy of undergraduate teaching at the participating institutions. We also used the summer months to foster appreciation for research in underprivileged and minority students in engineering, by

continuing to sponsor and participate in the established REU programs SUPERB-IT at UCB and SIPHER at VU.

We continue the collaboration with San Jose State University to continue to develop the undergraduate embedded control course jointly between Berkeley, Vanderbilt and San Jose State University. Prof. Ping Hsu from San Jose State University teaches the class at both Berkeley and San Jose State.

### 2.4.1. Curriculum Development for Modern Systems Science (MSS)

Our agenda is to restructure computer science and electrical engineering curricula to adapt to a tighter integration of computational and physical systems. Embedded software and systems represent a major departure from the current, separated structure of computer science (CS), computer engineering (CE), and electrical engineering (EE). In fact, the new, emerging systems science reintegrates information and physical sciences. The impact of this change on teaching is profound, and cannot be confined to graduate level. Based on the ongoing, groundbreaking effort at UCB, we are engaged in retooling undergraduate teaching at the participating institutions, and making the results widely available to encourage critical discussion and facilitate adoption.

We are engaged in an effort at UCB to restructure the undergraduate systems curriculum (which includes courses in signals and systems, communications, signal processing, control systems, image processing, and random processes). The traditional curriculum in these areas is mature and established, so making changes is challenging. We are at the stage of attempting to build faculty consensus for an approach that shortens the pre-requisite chain and allows for introduction of new courses in hybrid systems and embedded software systems.

#### Undergrad Course Insertion and Transfer

At many institutions, introductory courses are quite large. This makes conducting such a course a substantial undertaking. In particular, the newness of the subject means that there are relatively few available homework and lab exercises and exam questions. To facilitate use of this approach by other instructors, we have engaged technical staff to build web infrastructure supporting such courses. We have built an instructor forum that enables submission and selection of problems from the text and from a library of submitted problems and exercises. A server-side infrastructure generates PDF files for problem sets and solution sets.

The tight integration of computational and physical topics offers opportunities for leveraging technology to illustrate fundamental concepts. We have developed a suite of web pages with applets that use sound, images, and graphs interactively. Our staff has extended and upgraded these applets and created a suite of Powerpoint slides for use by instructors.

We have begun to define an upper division course in embedded software (aimed at juniors and seniors). This new course will replace the control course at the upper division level at San Jose State. We also continued to teach at UC Berkeley the integrated course designed by Prof. Lee, which employs techniques discovered in the hybrid and embedded systems research to interpret traditional signals.

A special topics course was taught in 2005-06 which grew out of original collaborations formed in the UC Berkeley SUPERB program. This course, which was taught by Prof. Sastry and Aaron

Ames, a graduate mentor in SUPERB, involved a SUPERB graduate who was a Berkeley student as well as two other undergraduates who were interested in the area. Based on results gained in SUPERB and through the ITR the students were able to produce realistic results that will be presented in conferences on robotics.

***Course: Structure and Interpretation of Signals and Systems (UCB, EECS 20N)***
***http://ptolemy.eecs.berkeley.edu/eecs20/***
Instructor: Prof. Edward A. Lee
Prof. Pravin Varaiya
Prof. Babak Ayazifar

This course is an introduction to mathematical modeling techniques used in the design of electronic systems. Signals are defined as functions on a set. Examples include continuous time signals (audio, radio, voltages), discrete time signals (digital audio, synchronous circuits), images (discrete and continuous), discrete event signals, and sequences. Systems are defined as mappings on signals. The notion of state is discussed in a general way. Feedback systems and automata illustrate alternative approaches to modeling state in systems. Automata theory is studied using Mealy machines with input and output. Notions of equivalence of automata and concurrent composition are introduced. Hybrid systems combine time-based signals with event sequences. Difference and differential equations are considered as models for linear, time-invariant state machines. Frequency domain models for signals and frequency response for systems are investigated. Sampling of continuous signals is discussed to relate continuous time and discrete time signals.

***Course: Bipedal Robotic Walking: From Theory to Practice (UCB, EECS Special Topics)***
***http://chess.eecs.berkeley.edu/bipeds/***
Instructor: Aaron D. Ames
Prof. Shankar Sastry

The goal of this experimental undergraduate research course is to introduce fundamental concepts from the area of hybrid and systems theory in the context of bipedal robotic walkers. This provides undergraduates with a physical system in which to understand abstract and complex concepts. The course begins by introducing the mathematical basics necessary to understand robotic systems undergoing impacts, including Lagrangians and hybrid systems; the students become familiar with these concepts by deriving the hybrid models for simple mechanical systems.

Bipedal walkers are then studied in detail beginning from the mathematical modeling of these systems, followed by the implementation of these mathematical models into software, and concluding with a systematic study of the behavioral properties of these systems, e.g., types of waking gaits, stability of waking gaits. Time permitting, original research topics in the area of bipedal robotic walking are addressed.

## Graduate Courses

Several graduate courses were taught in the area of embedded and hybrid systems, as well as systems modeling. All of these courses are a reflection of the teaching and curriculum goals of the ITR and its affiliated faculty.

***Course: Concurrent Models of Computation for Embedded Software (UCB, EECS 290N)***
***http://embedded.eecs.berkeley.edu/concurrency/***
Instructor: Prof. Edward A. Lee

This experimental research course will study models of computation used for the specification and modeling of concurrent real-time systems, particularly those with relevance to embedded software. Current research and industrial approaches will be considered, including real-time operating systems, synchronous languages (such as used in SCADE, Esterel, and Statecharts), timed models (such as used in Simulink, OPNET, NS-2, VHDL, and Verilog), dataflow models (such as a used in Labview and SPW), process networks (such as used in SDL), and software component models (such as nesC/TinyOS, Click, and CORBA). The course will combine an experimental approach with a study of formal semantics. The objective will be to develop a deep understanding of the wealth of alternative approaches to managing concurrency and time in software.

***Course: Hybrid Systems: Computation and Control (UCB, EECS 291E/ME 290S)***
***http://robotics.eecs.berkeley.edu/~sastry/ee291e/HSCC05.htm***
Instructor: Prof. S. Shankar Sastry
Dr. Jonathan Sprinkle

This course investigates modeling, analysis and verification of various classes of hybrid systems. Special attention is paid to computational and simulation tools for hybrid systems. Applications ranging from networked sensors, power electronics, avionics, and autonomous vehicles will be covered. The course consists of lectures, a handful of homework assignments, and a final project.

***Course: Embedded System Design: Models, Validation, and Synthesis (UCB EE249)***
Instructor: Prof. Alberto Sangiovanni-Vincentelli

This course is about the design of embedded real-time systems. Embedded realtime systems are pervasive in today's world. The methodology used for the design of these devices is still based on principles and tools that are not adequate for the complexity of the applications being developed today. The most important characteristic of these systems is the massive use of programmable components to achieve the design goals. Today, the dominant part of the design effort for embedded system is software. Real-time and power dissipation constraints make embedded software design particularly difficult since traditional abstraction for software do not include physical quantities. The choice of the architecture of the implementation is another essential characteristic of embedded system design. The implementation platform should be selected to support the application of interest optimizing a set of conflicting criteria that include flexibility, scalability, design time, manufacturing cost and reliability. In this course, we will present the principles of a methodology that favors design re-use, formal verification, software design and optimized architecture selection. The basic tenet of the methodology is

orthogonalization of concerns, and, in particular, separation of function and architecture, computation and communication. This methodology called platform-based design will be presented as a paradigm that incorporates these principles and spans the entire design process, from system-level specification to detailed circuit implementation.

***Course: Foundations of Hybrid and Embedded Systems (VU, CS 376)***
Instructor: Prof. Xenofon Koutsoukos
Prof. T. John Koo

Modeling, analysis, and design of hybrid and embedded systems. Heterogeneous modeling and design of embedded systems using formal models of computation, modeling and simulation of hybrid systems, properties of hybrid systems, analysis methods based on abstractions, reachability, and verification of hybrid systems.

***Course: Control Systems I (VU, EECE 257-01)***
Instructor: Prof. T. John Koo

Introduction to the theory and design of feedback control systems, steady-state and transient analyses, stability considerations, model representation, state-variable models. Prerequisite: EECE 213 or consent of instructor. The objective of this course is to develop an understanding and the ability to use basic tools for analyzing and designing linear, time-invariant control systems. Materials are primarily taught in classroom setting. A Matlab-based simulation package package Simulink will be used to design, analyze and simulate the control systems. This course is organized around the concepts of linear, time-invariant feedback control theory. Main emphasis will be on the classical methods of control engineering in the frequency and time domains. Also covered are the fundamental concepts of modern control theory including state variable modeling and solution of state equations.

***Course: Model Integrated Computing (VU, CS 388 / EE 395)***
Instructor: Prof. Janos Sztipanovits

Model-Integrated Computing (MIC) addresses the problems of designing, creating, and evolving information systems by providing rich, domain-specific modeling environments including model analysis and model-based program synthesis tools. MIC is used to create and evolve integrated, multiple-aspect models using concepts, relations, and model composition principles routinely used in the specific field, to facilitate systems/software engineering analysis of the models, and to automatically synthesize applications from the models.

***Course: Real-Time Systems (VU, EECE 353-01)***
Instructor: Prof. Aniruddha Gokhālé
Prof. Douglas Schmidt
Bala Natarajan

This course focuses on the analysis and design of real-time systems. The course covers topics on system modeling using the tagged signal models and timed models of computation, specifications and scheduling techniques for real-time tasks, simulation and verification of real-time systems, software architecture and language for constructing

real-time systems. Special attention is paid to computational and simulation tools for real-time systems. Applications ranging from robotics, embedded control systems, drive-by-wire systems, space missions, telecommunication systems, industrial automation, and middleware software systems will be covered.

***Course: Automated Verification (VU, EECE 315)***
Instructor: Dr. Sherif Abdelwahed

Several notations and methods have been developed to help the designer specify clear and unambiguous system requirements, verify that the requirements are consistent and correct, and verify that the refined design meets its specification. However, these methods are time-consuming and error-prone, and can be applied more effectively if there are tools to check their correctness. The goal of the course is to emphasize formal notations and methods that have tool support. We will cover the basis of underlying theory for the tools.

***Course: Automated Verification (VU, EECE 375)***
Instructor: Dr. Sherif Abdelwahed

This course provides a detailed coverage of the diagnosis and supervisory control problem for discrete, asynchronous, nondeterministic systems like manufacturing, traffic and communication systems. The underlying theory is developed in an elementary framework of automata and formal languages, and is supported by a software package for creating applications.

### 2.4.2. SUPERB-IT Program

The Summer Undergraduate Program in Engineering Research at Berkeley - Information Technology (SUPERB-IT) in the Electrical Engineering and Computer Sciences (EECS) Department offers a group of talented undergraduate engineering students the opportunity to gain research experience. The program's objective is to provide research opportunities in engineering to students who have been historically underrepresented in the field for reasons of social, cultural, educational or economic barriers, by affirming students' motivation for graduate study and strengthening their qualifications.

SUPERB-IT participants spent eight weeks at UC Berkeley during the summer of 2005 working on exciting ongoing research projects in information technology with EECS faculty mentors and graduate students. Students who participate in this research apprenticeship explore options for graduate study, gain exposure to a large research-oriented department, and are motivated to pursue graduate study. Additional information about the program can be obtained at:

http://www.eecs.berkeley.edu/Programs/ugrad/superb/superb.html

This ITR project contributed to the support of six SUPERB-IT students in 2005, and organized projects (described below in the abstracts from their papers) in hybrid systems theory, wireless sensor networks, dynamical systems simulation, and computer vision. The students were hosted by the Chess center at Berkeley (Center for Hybrid and Embedded Software Systems).

SUPERB-IT participants received a $3,500 stipend, room and board on campus in the International House, and up to $600 for travel expenses. In addition, Chess provided these

students with one laptop computer each, configured with appropriate software, plus laboratory facilities for construction of associated hardware.

The students supported at Berkeley in 2005 were Rey Romero, Lana Carnel, Simon Ng, Bobby Gregg, Murphy Gant, and Shams Karimkhan. The students worked on individual projects which were tailored to fit their individuation needs, interests, and background, as well as to contribute to the overall goals of the ITR project. Six graduate student mentors facilitated the process, and the operation was coordinated and directed by Professor Shankar Sastry and Dr. Jonathan Sprinkle. Each student was responsible for an individual project, as described below. Their project posters and reports are available at

http://chess.eecs.berkeley.edu/projects/ITR/2005/superb/index.html

***Project: Visual Target Segmentation and Identification***
Student: Lana Carnel—*University of Tennessee, Knoxville*
Mentor: Parvez Ahammad

> Locating, isolating and tracking the objects of interest are a key step in visual scene analysis in surveillance. Reflection, variations in appearance, clutter, and occlusion create challenges in identifying the object of interest robustly. Cues (or features) such as color, motion, size and dynamics must be used in tandem, to effectively decide what is relevant and discard the unnecessary information based on the goal at hand. In this work, we look at the target identification and tracking problem from the point of view of building surveillance applications and address the aforementioned issues by using multiple layers of segmentation. The approach involves intelligent feature selection and robust combination of these features to locate and track these objects. We demonstrate the results of our implementation on videos and images taken both in controlled and uncontrolled environments.
>
> *Lana Carnel is a rising Junior in Electrical Engineering at the University of Tennesse, Knoxville. She is working this summer with Mentor Parvez Ahammad.*
>
> *Her service activities for the 2005-06 academic year at UTK chapters include: President, Society of Women Engineers; Vice-Chair, IEEE; Secretary, Engineers Without Borders; Treasurer, Eta Kappa Nu.*
>
> *Lana graduated magna cum laude in May 2003 from the University of Tennessee, Knoxville, in the College Scholars Program, concentrating in Film Production and Cinema Studies.*

***Project: Modeling of Distributed Camera Networks***
Student: Murphy Gant—*University of California, Berkeley, via Sacramento City College*
Mentor: Yang Zhao

> Camera sensor networks are attracting for environment monitoring and object tracking; however, the main issue is effectively using the computation power of each camera. This paper investigates cheap cameras, with some compuation ability of basic information processing and some communication abilities. Through the functionalities of VisualSense, a modeling and simulation framework for wireless and sensor networks that builds and leverages on Ptolemy, one is not just confined to existing base classes or

libraries of subclasses that provide specific channel and node models, but is open to create their own composite actors and Java classes for simulation. Algorithms were created to handle such issues of camera management, visibility, and energy consumption and this research focused on the simulation of a camera network that monitored the motion of a single object in a level of Cory Hall. Implementation of reliable camera management techniques through the use of dual-staged state machines and intuitive procedures reinforced proposed solutions; however, there were tradeoff factors such as between communication and power consumption that were associated with trying to minimize or maximize certain elements of the system. Although this research was based on the limited processing capabilities of camera sensors, new insights into developing more formidable camera sensors would provide a springboard towards other advancements. Ultimately, regardless of any stance taken towards enhancing the capabilities of modeling camera sensor networks, the well-anchored framework of VisualSense supports most.

*Murphy Gant is a new addition to UC Berkeley's undergraduate Electrical Engineering and the Computer Sciences department, transferring from Sacramento City College, where he achieved magna cum laude honors.*

*His research interests include wireless communications systems and network securities. In the future, Murphy plans on taking part in research for the Team for Research in Ubiquitous Secure Technology (TRUST).*

*This summer Murphy is working with his mentor Yang Zhao, with intentions of modeling a distributed camera network system.*

***Project: Hybrid Reduction of a Bipedal Walker from Three to Two Dimensions***
Student: Bobby Gregg—*University of California, Berkeley*
Mentor: Aaron D. Ames

Because the complexity of bipedal walking robots doubles when increasing a model's dimensions from two to three, many previously established analytical techniques are computably impractical for three-dimensional models. If bipedal walkers can be analyzed in three dimensions, we can more accurately reproduce the humanoid walking that we observe in our three-dimensional world. This paper offers a systematic approach to reducing a 3D biped model into two dimensions, on which 2D analytical methods can be used, such as numerical analysis to find the limit cycles that result in asymptotically stable walking. The hybrid reduction consists of five stages: hybridization of the robot's motion, Lagrangian formulation of the continuous dynamics, formulation of the discrete impact transition map, dependency simplification, and the Lagrangian reduction. We present the results of this method's application on a simple compass-gait biped using a fixed angle simplification and Routhian reduction. We show that the reduced model is related to the analogous 2D model by a computable augmented potential component. The model is easily brought back into 3D using the Routhian relation and can be implemented in a simulation for analysis. Moreover, we provide supporting evidence for periodicity in the reconstructed 3D model given periodicity in the reduced 2D model. The outcome of this paper is a general framework by which previously established techniques can be applied to three-dimensional biped models.

*Robert D. Gregg is a rising senior at the University of California, Berkeley, majoring in Electrical Engineering and Computer Science. He is working this summer with Mentor Aaron D. Ames.*

*His areas of interest are control of hybrid systems, autonomous software and systems, and communication systems. His SUPERB work is the mathematical modelling and Lagrangian reduction of a passive bipedal walker from three to two dimensions. Robert plans on pursuing a PhD in electrical engineering and will be applying to schools in Fall 2005.*

*He is the Layout Manager of UC Berkeley's very own California Engineer magazine, which publishes undergraduate research from all the UC campuses. When he is not working in the research community, he serves as Chair of the Judicial Council of the Associated Students of the University of California. He is a member of the Chi Phi Fraternity and is a sitting member of the Greek Judicial Committee.*

***Project: A Hybrid Systems Approach to Communication Networks: Zeno Behavior and Guaranteed Simulations***
Student: Shams Karimkhan—*Wright State University*
Mentor: Alessandro Abate

In nature objects are time-dependent and with time they change in shape, position and composition. System engineering describes these phenomena with mathematical models. In this project we will use MATLAB to model and simulate Hybrid Systems (HS). A HS models a discrete program with an analog environment. The analog part is described by an ordinary differential equation (ODE). In order to know how the system works we need to find the solution to the ODE. Due to the fact that it is almost impossible to find the exact solution we need to use numerical approximation. By doing this we introduce some uncertainty, or rather some errors. The next step is controlling the errors, i.e. refining the simulations, in order to make sure that the simulated solution will have the same "shape" (behavior, evolution) of the actual simulation. This idea will be possibly embedded in the currently available simulation tools for HS. A TCP will be modeled as a HS and the Zeno behavior will be studied as well as error handling.

*Shams Karimkhan is a rising senior at Wright State University, majoring in Electrical Engineering. He is working this summer with Mentor Alessandro Abate.*

*Shams was born in San Francisco, in December 1982, and moved to Iran around the age of one year. After getting his diploma in math and physics, he moved to the US to continue his studies.*

*His specific research is on simulation of hybrid embedded systems. He will graduate from Wright State University, in Dayton, OH, in June 2006, and he plans to pursue his MS and PhD degrees.*

***Project: Modeling, Simulation, and Analysis of a Bipedal Walker***
Student: Simon Ng—*Michigan State University*
Mentor: Haiyang Zheng

The goal of this project is to simulate a bipedal walker that walks down a slight slope without any power. We would like to make a bipedal walker that allows continuous dynamics to be interrupted by discrete time events. We will use existing equations from past papers as well as reduplicate dynamics given by Aaron Ames and Bobby Gregg. We will use a software package called HyVisual. An analysis of how this was implemented in HyVisual as well as how the graphics were animated in Ptolemy. The results of using these tools will be a bipedal walker where one can expand the model in the future.

*Simon Ng is a rising senior at Michigan State University, majoring in Computer Science. He is working this summer with Mentor Haiyang Zheng.*

*Simon was born in Hong Kong and grew up in Michigan since he was three years old. His specific reserath with the SUPERB-IT program here at UC Berkeley for the summer is the simulation of a bipedal walker that walks down a slight slope, using the HyVisual modeling environment of the Ptolemy II tool. His future plans are to attend graduate school and head out into industry after completion.*

***Project: Modeling and Analysis of On-Chip Networks***
Student: Reinaldo Romero—*Pennsylvania State University*
Mentor: Alessandro Pinto

The complexity of hardware platforms doubles every eighteen months. The number of processing elements on the same chip is going to be of the order of hundreds in the near future which makes the communication infrastructure very difficult to design. Constraints, especially in terms of power consumption, must be taken into account and the communication infrastructure has to be highly optimized. The optimization of a network gives different results depending on the trade-off between communication and computation. In this work we derive an expression for such trade-off and we predict how future communication topologies will look like by analyzing on-chip networks using simple analytical models.

*Reinaldo Romero, aka Rey, is a rising senior at Penn State University, majoring in Electrical Engineering. He is working this summer with Mentor Alessandro Pinto.*

*Aside from his academics, he likes to play various sports including baseball, basketball, volleyball, and soccer.*

## Plans for 2006

The SUPERB program is dedicated to providing undergraduate students with the opportunities and experiences of research, and will take place between June 11 and August 4, 2006. At CHESS, where our research goals are at the intersection of traditional Electrical Engineering and Computer Science, this includes a wide variety of possible topics including robotics, systems theory, programming, image processing, algorithm development, simulation, and good old mathematics.

More importantly, the CHESSphilosophy is that new developments in traditional research areas will emerge from interdisciplinary cooperation between Electrical Engineering and Computer

Science researchers. To reflect this, and the wide array of possible topics, we have chosen for the 2006 SUPERB program a set of projects that intersect in some areas, are independent in others, and will allow for inter-student cooperation to devise new solutions to unsolved problems.

It is important to note that several projects fit into more than one category. We believe this is typical of research at Berkeley, and will be reflective of research everywhere in the future. Our goal is to use this as a benefit for the student over the summer, where the student will learn how to think in terms of multiple goals at once, and will achieve results that can be interesting to experts in more than one field.

***Project: Highway traffic flow analysis and control***
Student: Dominique Duncan, University of Chicago
Mentor: Alexandr Kurzhanskiy

Significant inspiration of computer network topology and communication comes from an empirical understanding of how road networks function. This project takes foundational work by CHESS researchers in hybrid systems to investigate a macroscopic switching-mode model (SMM) of traffic. The goal is to learn how hybrid systems are used for traffic modeling, experiment with different techniques for reachability analysis, implement a controller for the system, and then study the system behavior in the presence of disturbances. The end result will be MATLAB simulations which show the impact of the work.

*Dominique Duncan is a rising Junior at the University of Chicago, majoring in Math and Polish Literature.*

***Project: Tool for probabilistic safety verification of stochastic hybrid systems***
Student: Nandita Mitra, Rutgers University
Mentors: Saurabh Amin, Alessandro Abate

At the heart of research in the CHESS Center is the area of computational hybrid systems. The purpose of this project is to develop a computational tool to study some simple stochastic hybrid systems and to implement a methodology for stochastic reachability analysis for controlled SHS. For example, safety critical systems like air traffic control involves modeling their behavior as controlled stochastic hybrid systems (SHS), the quantification of the effect of external inputs and disturbances by way of simulation, and designing controllers that guarantee a certain safety criterion can both also be posed as stochastic hybrid systems problems. The end result of this project will be MATLAB simulations which show the impact of the work, as well as possible theoretical understandings in this rich area of research.

*Nandita Mitra is a rising Senior at Rutgers University, majoring in Electrical Engineering.*

***Project: Autopilot for ultra-light flying wing***
Student: Nashlie Sephus, Mississippi State University
Mentor: Todd Templeton

Embedded software is most interesting when it is actually doing something, and CHESS researchers have strong ties to autonomous systems. The purpose of this project is to

develop an auto-pilot for a small, light fixed-wing aircraft named the Zagi. This aircraft is interesting because it is inexpensive, simple and fast to deploy, and is virtually indestructible since it is made of foam. Development of the autopilot will allow for seamless simulation of high-level algorithms such as pursuit-evasion games, waypoint following, and loitering. The final goal is to deploy the autopilot interface into embedded hardware.

*Nashlie Sephus is a rising Senior at Mississippi State University, majoring in Computer Engineering.*

***Project: Viptos: A graphical development and simulation environment for***
Student: Heather Taylor, University of Vermont
Mentor: Elaine Cheong

TinyOS-based wireless sensor networks Wireless Sensor Networks are a burgeoning area of research and application in embedded systems. The purpose of this project is to understand and further develop Viptos (Visual Ptolemy and TinyOS), an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. TinyOS is the operating systems for the Berkeley Motes, which are small embedded systems capable of collecting audio, temperature, radio, and other kinds of sensor data. A key piece of the TinyOS simulator is the ability to simulate a network topology rapidly once it seems to behave appropriately. Viptos extends the capabilities of the TinyOS Simulator to allow simulation of heterogeneous networks. The final goal is to produce complex simulations and enable programming support that has not been previously possible.

*Heather Taylor is a rising Senior at the University of Vermont, majoring in Electrical Engineering.*

Six graduate student mentors have been identified to facilitate the process, and the operation is being coordinated and directed by Dr. Jonathan Sprinkle (the CHESS Executive Director). Although the students are working together and interacting extensively, each will be responsible for a single project's full completion.

In addition to the science of research, we will also expose students to the reporting aspects of research. These include the techniques used for writing papers, technical skills such as the use of LaTeX, and the importance of having a stock version of what exactly it is you are working on**.** Specific cross-cutting tasks include LaTeX template design, poster design and best practices, using the Concurrent Versioning System (CVS), participation in a literature reading group, and research reporting through the web.

### 2.4.3. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program

The SIPHER program (Summer Internship Program in Hybrid and Embedded Software Research) is a program similar to SUPERB-IT, but located at Vanderbilt. More information about the program can be found at:

http://fountain.isis.vanderbilt.edu/teaching

The Institute for Software-Integrated Systems (ISIS) at Vanderbilt University's School of Engineering in cooperation with UC Berkeley and University of Memphis has recently been awarded a grant by the National Science Foundation to conduct research in the field of Hybrid and Embedded Systems (HES). The research aims at laying the scientific and technological foundations of embedded system design. Embedded computing systems are present in all traits of modern society: in cars and airplanes, in cell phones, in household devices, in medical devices, just to name a few. This is a multi-year research project that builds the science: the principles and the math, and the technology: the tools that the next generation of engineers will use to build these systems in the future, better than ever before.

The objective of the SIPHER program is that undergraduates from underrepresented groups participate in the research program: receive training in the science and technology developed by the researchers, and work on specific research problems. The program will be coordinated with UC Berkeley's SUPERB-IT, and joint teleconferences are expected.

The undergraduate students who apply to this program are expected to be rising juniors and seniors in an Electrical or Computer Engineering or Computer Science program leading towards a BSc or BE degree. The students must have a background in elementary systems courses: signals and systems for EE, digital systems in CE/CS, and have skills in programming using a high-level language. Engineering students from other programs, like Mechanical Engineering and Chemical Engineering are also encouraged to apply, provided they have similar backgrounds (e.g. in controls).

The SIPHER program runs for 10 weeks each summer, and there are 7 (seven) positions available, with a $6,000 stipend for the period. This year, the participants will be partly funded by the Tennessee Louis Stokes Alliance for Minority Participation (TLSAMP) project (supported by NSF). Students are expected to pay for their accommodations. Limited housing opportunities may be available on the Vanderbilt campus. Applicants are competitively selected to the program.

In the SIPHER activities, we organized a summer internship in 2005 for eight participants from underrepresented groups. The students are organized into groups who solved different embedded software development problems. The students used Vanderbilt-developed modeling tool to create models of the embedded applications, develop code for the components, and then use the model transformation tools to create the final application.

The students worked on small, team-oriented projects related to development of embedded software. In this work they used software tools available at ISIS, and they were supervised by professors and senior graduate students. During first few weeks they underwent rigorous training to learn how to use the design tools. The training was provided by lecturers who deliver our Model-Integrated Computing classes. All of the students had backgrounds in programming, and thus were able to solve the project problems. Similarly to UCB, graduate student mentors assisted and guided the student projects. Descriptions for the projects and the students who worked on them are included below.

In 2005 we had 7 undergraduates supported by the ITR SIPHER program, and one additional undergraduate was supported by an NSF REU grant. All these students worked on small research projects related to the goals of the ITR.

***Project: Process Control Systems with Simulink/Stateflow***
Students: Mr Karlston Martin
Ms Shantell Hinton

Karlston and Shantell have worked on a process control system. First they studied a laboratory setup consisting of three thanks and interconnecting pipes, pumps, and valves. This is a classical system for modeling and analyzing hybrid system behavior, often called as the 'three-tank' (or 3T) system. They have developed models in Simulink/Stateflow, and they executed a system identification procedure to determine the plant's parameters. They have also experimented with various control algorithms for moving the fluid among the tanks and maintaining fluid levels.

***Project: Smart Structures***
Students: Ms Alicia Varden

Alicia has worked on prototype system for smart structures: a piezo-electrically activated bending plate. She has developed first a model of the plate in Simulink, then she designed a compensating controller for the system (that was responsible for damping out the vibrations of the plate), and finally she demonstrated the control software on the real physical device.

***Project: Wireless Sensor Networks***
Students: Ms Chanel Mitchell
Mr Omar Abdul-Ali

Chanel and Omar have worked on a wireless sensor network application. They have used a WiFi device as the wireless transmitter, and they were using Ptolemy II to model a setup with wireless devices placed into the different rooms in a building. They validated the models from actual measurements from the physical devices and studied the impact of the geometrical configuration on the quality of communication between devices.

***Project: Autonomous Robot Path Planning and Mapping***
Students: Ms Lauren Mitchell
Ms Sarah Francis

Lauren and Sarah have worked on small autonomous robots. They were using Bluetooth-equipped Lego robots, and they programmed the robots to perform small autonomous tasks, like mapping a maze and reporting the results to a master computer via the communication links. They have also experimented with various exploration and path finding algorithms that were controlling the robots.

***Project: Embedded real-time operating systems***
Students: Ryan Thibodeaux

Ryan has worked a small real-time operating system. He has ported the uCOS-II RTOS to a 68HC12 processor, such that the paging hardware of the processor was used. He has learned about paging hardware issues, interrupt systems, and the implementation of real-time kernels in general. His project has results are being used in our embedded system undergraduate courses. He has also been successfully recruited to our EE PhD Graduate Program

## Plans for 2006

SIPHER is dedicated to providing undergraduate students with the opportunities and experiences of research. As the main research goals of the project are set, the selected research topics are at the intersection of traditional Electrical Engineering and Computer Science. For this reason, we will implement a set of research projects that for the 2006 SIPHER program that reflect this goal.

In addition to the science of research, we will also expose students to the other aspects of research. During the summer programs, there will be two formal reviews, with presentations and formal reports. The students will also participate in field trips to nearby industrial and research sites.

# 3. Publications and Products

In this section, we list published papers only. Submitted papers and in press papers are described in Section 2.2.

## 3.1. Journal Publications

- Online Safety Calculations for Glideslope Recapture, Jonathan Sprinkle, Aaron D. Ames, J. Mikael Eklund, Ian Mitchell, S. Shankar Sastry. Online Safety Calculations for Glideslope Recapture. *Innovations in Systems and Software Engineering*, 1(2):157-175, September 2005; This was an invited paper, and was published without peer review.

- Games with Secure Equilibria, Krishnendu Chatterjee, Thomas A. Henzinger and Marcin Jurdzinski. Games with Secure Equilibria. *Theoretica Computer Science*, January 2006.

- Automotive engine hybrid modelling and control for reduction of hydrocarbon emissions, P.R. Sanketi, J.C. Zavala and J.K. Hedrick. Automotive engine hybrid modelling and control for reduction of hydrocarbon emissions. *International Journal of Control*, 79(5):449-464, May 2006.

- Languages and Tools for Hybrid Systems Design, Luca P. Carloni, Roberto Passerore, Alessandro Pinto and Alberto Sangiovanni-Vincentelli. Languages and Tools for Hybrid Systems Design. *Foundations and Trends in Design Automation*, 1(1):1-204, January 2006.

- The Problem with Threads, Edward A. Lee. The Problem with Threads. *IEEE Computer*, 39(5):33-42, May 2006.

- S. Abdelwahed, J. Wu, G. Biswas, J. Ramirez, E.J. Manders, "Online Fault-Adaptive Control for Efficient Resource Management in Advanced Life Support Systems," *Habitation: International Journal of Human Support Research*, vol. 10, no. 2, pp. 105-115, 2005.

- G. Biswas, E.J. Manders, J.W. Ramirez, N. Mahadevan, S. Abdelwahed, "Online Model-Based Diagnosis to Support Autonomous Operation of an Advanced Life Support System," *Habitation: International Journal of Human Support Research*, vol. 10, no. 1, pp. 21-38, 2004.

- M. Emerson, J. Sztipanovits, T. Bapty, "A MOF-Based Metamodeling Environment," *Journal of Universal Computer Science*, vol. 10, No. 10, pp. 1357-1382, October, 2004.

- K. D. Frampton, "Distributed Group-Based Vibration Control with a Networked Embedded System," *Journal of Intelligent Materials Systems and Structures*, Vol. 14, pp. 307--314, 2005.

- G. Karsai, A. Lang, S. Neema, "Design Patterns for Open Tool Integration," *Journal of Software and System Modeling*, vol 4., no. 1, DOI: 10.1007/s10270-004-0073-y, 2004.

- E. A. Lee, E.H. Abed (Ed.), "Engineering Education: A Focus on System, in Advances in Control, Communication Networks, and Transportation Systems: In Honor of Pravin Varaiya," *Systems and Control: Foundations and Applications Series*, Birkhauser, Boston, 2005.

- E. A. Lee, "What are the Key Challenges in Embedded Software?" *Guest Editorial in System Design Frontier*, Shanghai Hometown Microsystems Inc., Volume 2, Number 1, January 2005.

- M. Maroti, G. Simon, A. Ledeczi, J. Sztipanovits, "Shooter Localization in Urban Terrain," *IEEE Computer*, pp. 60-61, August, 2004.

- G. C. Necula, J. Condit, M. Harren, S. McPeak, W. Weimer, "CCured: Type-Safe Retrofitting of Legacy Software," *ACM Transactions on Programming Languages and Systems*, vol. 27, no. 3, May, 2005.

- P. Schmidt, I. Amundson, K.D. Frampton, "A Distributed Algorithm for Acoustic Localization Using a Distributed Sensor Network," *Journal of the Acoustical Society of America*, Vol. 115, No. 5, Pt. 2, pp. 2578, 2004.

- J. Sprinkle, "Generative Components for Hybrid Systems Tools," *Journal of Object Technology*, vol. 4, no. 3, April 2005, *Special issue: 6th GPCE Young Researchers Workshop 2004*, pp. 35-39, Available at http://www.jot.fm/issues/issue_2005_04/article5

- J. Sprinkle, G. Karsai, "A Domain-Specific Visual Language for Domain Model Evolution," *J. Vis. Lang. and Comp.*, vol. 15, no. 3-4, pp. 291-307, Jun., 2004.

- T. Szemethy, G. Karsai, "Platform Modeling and Model Transformations for Analysis," *Journal of Universal Computer Science*, vol. 10, no. 10, pp 1383-1406, 2004.


## 3.2. Conference Papers


- Jonathan Sprinkle, Aaron D. Ames, Alessandro Pinto, Haiyang Zheng, S. Shankar Sastry. On the Partitioning of Syntax and Semantics For Hybrid Systems Tools. 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), IEEE Controls Society, 4694-4699, December, 2005.

- Matthew Harren and George C. Necula. Using Dependent Types to Certify the Safety of Assembly Code. Static Analysis Symposium (SAS), Springer-Verlag LNCS, 155-170, September, 2005; Available at http://www.cs.berkeley.edu/~matth/papers/sas05.pdf.

- HyVisual: A Hybrid System Modeling Framework Based on Ptolemy II, Edward A. Lee and Haiyang Zheng. HyVisual: A Hybrid System Modeling Framework Based on Ptolemy II. IFAC Conference on Analysis and Design of Hybrid Systems, January, 2006.

- Elaine Cheong, Edward A. Lee, and Yang Zhao. Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks. Proceedings of the Third ACM Conference on Embedded Networked Sensory Systems, ACM, November, 2005.

- Jie Liu, Elaine Cheong, and Feng Zhao. Semantics-Based Optimization Across Uncoordinated Tasks in Networked Embedded Systems. 5th ACM Conference on Embedded Software (EMSOFT 2005), EMSOFT '05, September, 2005.

- Edward A. Lee, Haiyang Zheng, Ye Zhou. Causality Interfaces and Compositional Causality Analysis. Foundations of Interface Technologies (FIT), CONCUR 2005, ENTCS TBD, August, 2005.

- E. Wandeler, J.W. Janneck, E.A. Lee, L. Thiele. Counting Interface Automata and their Application in Static Analysis of Actor Models. 3rd International Conference on Software Engineering and Formal Methods - SEFB 2005, SEFM 2005, September, 2005.

- Y. Xiong, E.A. Lee, X. Liu, Y. Zhao, L.C. Zhong. The Design and Application of Structured Types in Ptolemy II. IEEE Int. Conf. on Granular Computing, Grc 2005, July, 2005.

- J. Mikael Eklund, Thomas Risgaard Hansen, Jonathan Sprinkle, S. Shankar Sastry. Information Technology for Assisted Living at Home: Building a Wireless Infrastructure for Assisted Living. 27th Annual International Conference of the IEEE Engineering In Medicine and Biology Society (EMBS), 3931-3934, September, 2005.

- J. Mikael Eklund, Jonathan Sprinkle, S. Shankar Sastry. Implementing and Testing a Nonlinear Model Predictive Tracking Controller for Aerial Pursuit Evasion Games on a Fixed Wing Aircraft. Proceedings of American Control Conference (ACC) 2005, 1509-1514, June, 2005.

- Jonathan Sprinkle, J. Mikael Eklund, S. Shankar Sastry. Deciding to Land a UAV Safely in Real Time. Proceedings of American Control Conference (ACC) 2005, 3506-3511, June, 2005.

- J. Mikael Eklund, Ruzena Bajcsy, Jonathan Sprinkle, Gregory V. Simpson. Computing Inverse MEG Signals in the Brain. 2005 IEEE Computational Systems Bioinformatics Conference, Controlling Complexity, 332-335, August, 2005.

- Krishnendu Chatterjee, Thomas A. Henzinger and Marcin Jurdzinski. Mean-Payoff Parity Games. LICS 05, June, 2005.

- Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger. The Complexity of Stochastic Rabin and Streett Games. ICALP, July, 2005.

- Krishnendu Chatterjee, Thomas A. Henzinger, Ranjit Jhala and Rupak Majumdar. Counterexample-guided Planning. UAI, July, 2005.

- Krishnendu Chatterjee. Two-player Nonzero-sum \omega-Regular Games. CONCUR, August, 2005.

- Arindam Chakrabarti, Krishnendu Chatterjee, Thomas A. Henzinger, Orna Kupferman and Rupak Majumdar. Verifying Quantitative Properties Using Bound Functions. CHARME, 50--64, October, 2005.

- Krishnendu Chatterjee and Thomas A. Henzinger. Semi-perfect Information Games. FSTTCS, December, 2005.

- Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger. The Complexity of Quantitative Concurrent Parity Games. SODA, January, 2006.

- Krishnendu Chatterjee and Thomas A. Henzinger. Strategy Improvement and Randomized Subexponential Algorithms for Stochastic Parity Games. STACS, February, 2006.

- Krishnendu Chatterjee, Rupak Majumdar and Thomas A. Henzinger. Markov Decision Processes with Multiple Objectives. STACS, February, 2006.

- Krishnendu Chatterjee and Thomas A. Henzinger. Finitary Winning in \omega-Regular Games. TACAS, March, 2006.

- Pannag R Sanketi, J. Carlos Zavala, J. K. Hedrick. Dynamic Surface Control of Engine Exhaust Hydrocarbons and Catalyst Temperature for Reduced Coldstart Emissions. Proc. of International Federation of Automatic Control (IFAC) Conference, July, 2005; Prague, Czech Rep.

- Pannag R Sanketi, J. K. Hedrick, Tomoyuki Kaga. A Simplified Catalytic Converter Model for Automotive Coldstart Control Applications. Proceedings of 2005 ASME International Mechanical Engineering Congress and Exposition (IMECE2005), November, 2005; Orlando, Florida USA.

- Thomas Henzinger, Christoph Kirsch, Slobodan Matic. Composable Code Generation for Distributed Giotto. Proceedings of LCTES 2005, 21-30, June, 2005.

- Slobodan Matic, Thomas Henzinger. Trading End-to-End Latency for Composability. Proceedings of RTSS 2005, 99-110, December, 2005.

- Thomas Henzinger, Slobodan Matic. An Interface Algebra for Real-time Components. Proceedings of RTAS 2006, 253-263, April, 2006.

- Thomas A. Henzinger, Rupak Majumdar, and Vinayak Prabhu. Quantifying similarities between timed systems.. Proceedings of the Third International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS), Lecture Notes in Computer Science 3829, Springer, 2005, 226-241, September, 2005.

- Abhijit Davare, Qi Zhu, John Moondanos, Alberto Sangiovanni-VIncentelli. JPEG Encoding on the Intel MXP5800: A Platform-Based Design Case Study. ESTIMedia 2005: 3rd Workshop on Embedded Systems for Real-time Multimedia, September, 2005.

- Dirk Beyer, Arindam Chakrabarti, Thomas A. Henzinger. An Interface Formalism for Web Services. Foundations of Interface Technologies (FIT), 2005, August, 2005.

- S. Amin, A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Reachability Analysis of Controlled Discrete-Time Stochastic Hybrid Systems. Hybrid Systems: Computation and Control, Proceedings of the 9th International Workshop, Santa Barbara, CA, vol. 3927 of Lecture Notes in Computer Science, J. Hespanha and A. Tiwari, Springer-Verlag, pp. 49-63, March, 2006.

- A. Abate, A. D. Ames, and Shankar S. Sastry. A-Priori Detection of Zeno Behavior in Communication Networks Modeled as Hybrid Systems. Proc. 25th IEEE American Control Conference, Minneapolis, MN, Jun. 2006., January, 2006.

- Alessandro Pinto, Luca P. Carloni, Roberto Passerone and Alberto Sangiovanni-Vincentelli. Interchange Formats for Hybrid Systems: Abstract Semantics. Hybrid Systems: Computation and Control, Joao Hespanha and Ashish Tiwari, 491-506, March, 2006.

- A. Abate, A. D. Ames, and S. Sastry. Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems. Proc. 25th IEEE American Control Conference, Minneapolis, MN, Jun. 2006., January, 2006.

- A. Abate, A. D. Ames, and S. S. Sastry. Stochastic Approximations of Hybrid Systems. Proc. 24th IEEE American Control Conference, Portland, OR, 2005, January, 2006.

- Xi Chen, Abhijit Davare, Harry Hsieh, Alberto Sangiovanni-Vincentelli, Yosinori Watanabe. Simulation Based Deadlock Analysis for System Level Designs. 42nd Annual Design Automation Conference, 260-265, June, 2005.

- Haibo Zeng, Abhijit Davare, Alberto Sangiovanni-Vincentelli, Sampada Sonalkar, Sri Kanajan, Claudio Pinello. Design Space Exploration of Automotive Platforms in Metropolis. Society of Automotive Engineers Congress, April, 2006.

- Yujia Jin, Nadathur Satish, Kaushik Ravindran, Kurt Keutzer. An automated exploration framework for FPGA-based soft multiprocessor systems. Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis CODES+ISSS '05, ACM Press, 273 - 278, September, 2005.

- Kaushik Ravindran, Nadathur Satish, Yujia Jin, Kurt Keutzer. An FPGA-Based Soft Multiprocessor System for IPv4 Packet Forwarding. Proceedings of the 15th International Conference on Field Programmable Logic and Applications (FPL-05), 487-492, August, 2005.

- A. D. Ames, A. Sangiovanni-Vincentelli and S. Sastry. Homogeneous Semantics Preserving Deployments of Heterogeneous Networks of Embedded Systems. Workshop on Networked Embedded Sensing and Control, October, 2005.

- A. D. Ames, A. Sangiovanni-Vincentelli and S. Sastry. Homogenous Semantic Preserving Deployments of Heterogenous Networks of Embedded Systems. Workshop on Networked Embedded Sensing and Control, October, 2005.

- Arkadeb Ghosal, J. Carlos Zavala J., Marco A. A. Sanvido and J. Karl Hedrick. Implementation of AFR Controller in an Event-driven Real Time Language. 2005 American Control Conference, June, 2005.

- A. D. Ames, A. Abate and S. Sastry. Sufficient Conditions for the Existence of Zeno Behavior. IEEE Conference on Decision and Control, December, 2005.

- Guang Yang, Xi Chen, Felice Balarin, Harry Hsieh, Alberto Sangiovanni-Vincentelli. Communication and Co-Simulation Infrastructure for Heterogeneous System Integration. Design Automation and Test in Europe, March, 2006.

- 5th OOPSLA Workshop on Domain-Specific Modeling (DSM'05). Juha-Pekka Tolvanen, Jonathan Sprinkle, Matti Rossi, Computer Science and Information System Reports, Technical Reports, TR-36, University of Jyväskylä, Finland, October, 2005.

- A. Abate, L. El Ghaoui, "Robust Convex Optimization through Adjustable Robust Variables: an application to Model Predictive Control," In Proc. International Conference on Decision and Control, Atlantis, Bahamas, December 2004.

- A. Abate, L. Shi, S. Simic, S. Sastry, "A Stability Criterion for Stochastic Hybrid Systems," *In Proc. International Symposium of Mathematical Theory of Networks and Systems*, Leuven, July 2004.

- A. Agrawal, Gy. Simon, G. Karsai, "Semantic Translation of Simulink/Stateflow models to Hybrid Automata using Graph Transformations," Electronic Notes in Theoretical Computer Science, In Proc. Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2004), Volume 109, pp. 43-56.

- A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, G. Karsai, "Reusable Idioms and Patterns in Graph Transformation Languages," International Workshop on Graph-Based Tools, In Proc. 2004 International Conference on Graph Transformations, Rome, Italy, October, 2004.

- A. D. Ames, S. Sastry, "Blowing Up Affine Hybrid Systems," *43rd IEEE Conference on Decision and Control 2004 (CDC'04)*, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 473-478.

- A. D. Ames, S. Sastry, "A Homology Theory for Hybrid Systems: Hybrid Homology," *In Proc. Hybrid Systems: Computation and Control, 8th International Workshop*, Zurich, Switzerland, March 9-11, M. Morari and L. Thiele, eds., vol. 3414 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 86-102, 2005.

- I. Amundson, P. Schmidt, K. D. Frampton, "A Decentralized Approach to Sound Source Localization with Sensor Networks," *Presented at the 2004 ASME International Mechanical Engineering Conference and Exposition*, Anaheim CA, November 2004.

- D. Beyer, A. Chakrabarti, T. A. Henzinger, "Web Service Interfaces," *Proc. 14th International World Wide Web Conference (WWW 2005)*, Chiba, Japan, May 10—14 2005.

- K. Chatterjee, L. de Alfaro, T. A. Henzinger. "Trading Memory for Randomness," *In Proc. 1st International Conference on Quantitative Evaluation of Systems (QEST 04)*, University of Twente, Enschede, The Netherlands, September 27 --30, 2004.

- E. Cheong, J. Liu, "galsC: A Language for Event-Driven Embedded Systems," *Presented at Design, Automation and Test in Europe (DATE)*, Munich, Germany, March 7--11, 2005.

- A. Davare, K. Lwin, A. Kondratyev, A. Sangiovanni-Vincentelli. "The Best of Both Worlds: The Efficient Asynchronous Implementation of Synchronous Specifications," *Presented at ACM/IEEE Design Automation Conference*, San Diego, CA, June 7 --11, 2004.

- M. Emerson, J. Sztipanovits, "Implementing a MOF-Based Metamodeling Environment Using Graph Transformations" *In Proc. 4th Workshop on Domain-Specific Modeling*, *20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA),* pp. 83-92, Vancouver, Canada, October 2004.

- K. D. Frampton, "Vibroacoustic Control with a Distributed Sensor Network," *Presented at Applications of Graph Transformations with Industrial Relevance (AGTIVE04*), Williamsburg, VA, September, 2004.

- R. D. Harvey, D. G. Walker, K. D. Frampton, "Distributed Control to Improve Performance of Thermoelectric Coolers," *Presented at 2004 ASME International Mechanical Engineering Conference and Exposition*, Anaheim CA, November 2004.

- G. Karsai, A. Agrawal, "Graph Transformations in OMG's Model-Driven Architecture," *In Proc. Applications of Graph Transformations with Industrial Relevance (AGTIVE 2003)*, LNCS 2062. pp. 243-259, Charlottesville, VA, September 29—October 1, 2003.

- E. A. Lee, S. Neuendorffer, "Classes and Subclasses in Actor-Oriented Design," invited paper, *Conference on Formal Methods and Models for Codesign/ (MEMOCODE)*, San Diego, CA, USA, June 22-25, 2004.

- E. A. Lee, H. Zheng, "Operational Semantics of Hybrid Systems" invited paper, *In Proc. Hybrid Systems: Computation and Control (HSCC),* LNCS TBD, Zurich, Switzerland, March 9-11, 2005.

- G. Madl, S. Abdelwahed, G. Karsai, "Automatic Verification of Component-based Real-time CORBA Applications," *In Proc. 25th IEEE International Real-Time Systems Symposium (RTSS'04)*, Lisbon, Portugal, Dec. 2004, pp. 231—240.

- M. L. McKelvin, Jr, J. Sprinkle, C. Pinello, A. Sangiovanni-Vincentelli, "Fault Tolerant Data Flow Modeling Using the Generic Modeling Environment," *12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, Greenbelt, Maryland, Apr. 4--5, 2005, pp. 229–235.

- T. Meyerowitz, J. Sprinkle, A. Sangiovanni-Vincentelli, "A Visual Language for Describing Instruction Sets and Generating Decoders," *20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, Vancouver, BC, Oct., 25, 2004, pp. 23–32.

- S. Neema, A. Dixon, T. Bapty, J. Sztipanovits, "Model-Integrated Computing for Heterogeneous Systems," *In Proc. International Conference on Computing, Communications and Control Technologies (CCCT'04)*, Austin, TX, August 14-17, 2004.

- S. Neuendorffer, "Modeling Real-World Control Systems: Beyond Hybrid Systems," *In Proc. Winter Simulation Conference (WSC)*, Washington, DC, USA, December 5--8, 2004.

- W. Plishker, K. Ravindran, N. Shah, K. Keutzer. "Automated Task Allocation for Network Processors," *In Proc. Network System Design Conference*, October, 2004, pp. 235-245.

- L. Shi, A. Abate, S. Sastry, "Optimal Control for a class of Stochastic Hybrid Systems," *In Proc. International Conference on Decision and Control*, Atlantis, Bahamas, December 2004.

- J. Sprinkle, J. Davis, G. Nordstrom, "A Paradigm for Teaching Modeling Environment Design," *20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Educators Symposium (Poster Session)*, ACM, Vancouver, BC, Oct., 24--28, 2004.

- J. Sprinkle, J. M. Eklund, H. J. Kim, S. S. Sastry, "Encoding Aerial Pursuit/Evasion Games with Fixed Wing Aircraft into a Nonlinear Model Predictive Tracking Controller," *In Proc. IEEE Conference on Decision and Control*, pp. 2609--2614 , Dec., 2004.

- J. Sprinkle, J. M. Eklund, S. S. Sastry, "Toward Design Parameterization Support for Model Predictive Control," *IEEE 4th International Conference on Intelligent Systems Design and Application*, IEEE, IEEE Press, Budapest, Hungary, Aug., 26--28, 2004.

- J. Sprinkle, O. Shakernia, R. Miller, S. S. Sastry, "Using the Hybrid Systems Interchange Format to Input Design Models to Verification & Validation Tools," *IEEE Aerospace Conference*, Big Sky, MT, Mar., 2005.

- T. Tao, K.D. Frampton, "Experiments on the Decentralized Vibration Control with Networked Embedded Systems," *Presented at the 2004 ASME International Mechanical Engineering Conference and Exposition*, Anaheim CA, November 2004.

- J. Tolvanen, J. Sprinkle, M. Rossi, eds., "Proceedings of the 4th OOPSLA Workshop on Domain-Specific Modeling (DSM'04)," Jyvavaskyla, Finland, *20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, University of Jyvavaskyla, Oct., 2004.

- A. Vizhanyo, A. Agrawal, F. Shi, "Towards Generation of Efficient Transformations," *In Proc. Generative Programming and Component Engineering, 3rd International Conference*, October, 2004, LNCS 3286, pp 298-316, 2004.

- D. G. Walker, K. D. Frampton, R.D. Harvey, "Distributed Control of Thermoelectric Coolers," *In Proc. 9th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, paper no. 151, Las Vegas, NV, June 1-4, 2004, pp. 361--366.

## 3.3. Books, Reports, and Other One-Time Publications

- Elaine Cheong, Prof. Edward A. Lee, Yang Zhao. Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks. Technical report, EECS Dept. UC Berkeley, 15, February, 2006; Presented in conjunction with BEARS 2006.

- Huining Thomas Feng and Edward A. Lee. Incremental Checkpointing with Application to Distributed Discrete Event Simulation. Technical report, EECS Dept., University of California Berkeley, 37, April, 2006.

- James Adam Cataldo and Edward A. Lee. Composition Languages. Technical report, EECS Dept., University of California, Berkeley, 24, March, 2006; Found at: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-24.pdf.

- James Adam Cataldo, Edward A. Lee, Xiaojun Liu, Eleftherios Dimitrios Matsikoudis and Haiyang Zheng. A Constructive Fixed-Point Theorem and the Feedback Semantics of Timed Systems. Technical report, EECS Dept. University of California, Berkeley, 4, January, 2006.

- Edward A. Lee. The Problem with Threads. Technical report, EECS Dept., University of California, Berkeley, 1, January, 2006.

- Edward A. Lee. Building Unreliable Systems out of Reliable Components: The Real Time Story. Technical report, EECS Dept., University of California, Berkeley, 5, October, 2005.

- C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Heterogeneous Concurrent Modeling and Design in Java, (Volume 1, Introduction to Ptolemy II). Technical report, EECS Dept., UC Berkeley, 21, July, 2005.

- C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture). Technical report, EECS Dept., UC Berkeley, 22, July, 2005.

- C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.). Heterogeneous Concurrent Modling and Design in Java (Volume 3: Ptolemy II Domains). Technical report, EECS Dept., UC Berkeley, 23, July, 2005.

- Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao. VisualSense: Visual Modeling for Wireless and Sensor Network Systems. Technical report, EECS Dept., UC Berkeley, 25, July, 2005.

- HyVisual: A Hybrid System Visual Modeler. Technical report, EECS Dept., UC Berkeley, July, 2005.

- Adam Cataldo, Elaine Cheong, Thomas Huining Feng, Edward A. Lee and Andrew Mihal. A Formalism for Higher-Order Composition Languages that Satisfies the Church-Rosser Property. Technical report, EECS Dept., University of California, Berkeley, 48, May, 2006.

- Alex A. Kurzhanskiy, Pravin Varaiya. Ellipsoidal Toolbox. Technical report, EECS, UC Berkeley, January, 2006.

- Abhijit Davare, Qi Zhu, Alberto Sangiovanni-Vincentelli. A Platform-based Design Flow for Kahn Process Networks. Technical report, UC Berkeley, 2006-30, March, 2006.

- Jike Chong, Abhijit Davare, Kelvin Lwin. Concurrent Embedded Design for Multimedia: JPEG encoding on Xilinx FPGA Case Study. Technical report, UC Berkeley, April, 2006.

- Xiaojun Liu, Eleftherios Matsikoudis and Edward A. Lee. Modeling Timed Concurrent Systems using Generalized Ultrametrics. Technical report, EECS Department, UC Berkeley, May, 2006.

- Xiaojun Liu and Edward A. Lee. COP Semantics of Timed Interactive Actor Networks. Technical report, EECS Department, UC Berkeley, 67, May, 2006.

- Arkadeb Ghosal, Thomas A. Henzinger, Daniel Iercan, Christoph Kirsch and Alberto L. Sangiovanni-Vincentelli. Hierarchical Timing Language. Technical report, EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-20, May, 2006.

- Xiaojun Liu. Semantic Foundation of the Tagged Signal Model. PhD thesis, University of California, Berkeley, December, 2005.

- Jongho Lee. New real-time embedded software for an autonomous helicopter system using Giotto. Master's thesis, UC Berkeley, May, 2006.

- A. D. Ames, S. Sastry, "A Homology Theory for Hybrid Systems: Hybrid Homology," *In Proc. Hybrid Systems: Computation and Control, 8th International Workshop*, Zurich, Switzerland, March 9-11, M. Morari and L. Thiele, eds., vol. 3414 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 86-102, 2005.

- C. Brooks, A. Cataldo, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, H. Zheng, "HyVisual: A Hybrid System Visual Modeler," *Technical Memorandum UCB/ERL M04/18/*, University of California, Berkeley, June 28, 2004.

- C. Brooks, E.A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II)," *Technical Memorandum UCB/ERL M04/27/*, University of California, Berkeley, July 29, 2004.

- C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture)," *Technical Memorandum UCB/ERL M04/16/*, University of California, Berkeley, June 24, 2004.

- C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)," *Technical Memorandum UCB/ERL M04/17/*, University of California, Berkeley, June 24, 2004.

- M.Chen, A. Abate, A. Zakhor, S. Sastry, "Stability and Delay Considerations for Flow Control Over Wireless Networks," *UCB ERL Tech Report No M05/14*, Berkeley, CA, 2005.

- J. Davis, C. Hylands., J. Janneck, E.A. Lee, J. Liu, X.Liu, S. Neuendorffer, S. Sachs, M. Stewart, K. Vissers, P. Whitaker, Y. Xiong, "Overview of the Ptolemy Project," *Technical Memorandum UCB/ERL M01/11*, EECS, University of California, Berkeley, March 6, 2001.

- V. Krishnan, Master's Report, "Real-Time Systems Design in Ptolemy II: A Time-Triggered Approach," *Technical Memorandum UCB/ERL M04/22/*, University of California, Berkeley, July 12, 2004.

- E. A. Lee, Editorial, February 19, 2005, "Absolutely Positively On Time: What Would It Take?" Available at http://ptolemy.eecs.berkeley.edu/publications/papers/05/EmbeddedSoftwareColumn/

- E. A. Lee, "Balance between Formal and Informal Methods, Engineering and Artistry, Evolution and Rebuild," *Technical Memorandum UCB/ERL M04/19/*, University of California, Berkeley, July 4, 2004.

- E. A. Lee, "Concurrent Models of Computation for Embedded Software," *Technical Memorandum, University of California, Berkeley, UCB/ERL M05/2/*, January 4, 2005.

- E. A. Lee, S. Neuendorffer, "Concurrent Models of Computation for Embedded Software," *Technical Memorandum UCB/ERL M04/26/*, University of California, Berkeley, July 22, 2004.

- S. Neuendorffer, "Actor-Oriented Metaprogramming," *PhD Thesis*, University of California, Berkeley, December 21, 2004.

- G. Zhou, "Dynamic Dataflow Modeling in Ptolemy II," *Master's Report, Technical Memorandum No. UCB/ERL M05/2/*, University of California, Berkeley, December 21, 2004.

## 3.4. Dissemination

Although this is a long term project focused on foundations, we are actively working to set up effective technology transfer mechanisms for dissemination of the research results. A major part of this is expected to occur through the open dissemination of software tools.

### 3.4.1. Software Maturation

Making these software tools useful and usable outside the research community is a significant issue. Towards this end, we have cooperated with the formation of the Escher consortium, which has begun operating (www.escherinstitute.org). Escher has negotiated with both Berkeley and Vanderbilt specific priorities for "industrial hardening" of research tools from this project. In particular, at Berkeley, top priority will be placed on Giotto, xGiotto, and Ptolemy II in the near term. At Vanderbilt, top priority will be placed on GME, Desert, and GReAT. General Motors, Raytheon, and Boeing are signed up as charter industrial partners in Escher, and more companies are expected.

### Industry Technology Transition

1. The MIC tool suite is included in the ESCHER maturation program funded by GM, Boeing and Raytheon. The tool suite is now fully integrated and accessible through the quality controlled ESCHER Repository.
2. The MIC tool suite has been adopted by the Boeing Company, Lead System Integrator for the Future Combat Systems (FCS) program, for architecture modeling, architecture

exploration and model-based system integration. These applications has opened up new dimensions for the industrial applicability of model-based design approaches.

3. GM Research has continued working with the MIC tool suite in a new experimental vehicle program: Smart Adaptive Vehicle (SAV-2). This effort is a significant driver for our semantic foundations work and provides for us a continuous stream of "industrial strength" challenges.
4. The Raytheon Company has completed the development of the Signal Processing Platform tool suite based on MIC and tests its application in various programs.
5. We continue interaction with Microsoft Visual Studio Product Division on using our experience in the Software Factory product line, and started working with Microsoft Research on expanding our collaboration on Abstract State Machines, the Abstract State Machine Language (AsmL) and its application in semantic anchoring.

### 3.4.2. Working Groups and Standards

An important, emerging forum for dissemination of information and influencing industrial practice is the recently form Model-Integrated Computing Special Interest Group (MIC PSIG) of OMG. (http://mic.omg.org/) This forum is run by industry and its primary goal is the preparation and management of standardization activities related to various aspects model-based design in embedded systems. The ISIS and CHESS teams are very much involved these activities. The White Paper for a standard Open Tool Integration Framework (OTIF) is based on the work of researcher at ISIS.

### 3.4.3. "After Theory?" The 2005-2006 Chess seminar series

The Chess seminar series provides a weekly forum for the problems and solutions found and solved by Chess members, as well as ongoing research updates. This forum works best when the audience is diverse in background, because the goal is to aid researchers in seeing how the other sub-disciplines are approaching similar problems, or to encourage them to work on problems they had not yet considered.

A common thread for this year's seminar series is "After Theory?" which explored how to best provide support for existing and emerging problems in embedded and hybrid systems. Many difficult problems can be solved as proof of concept; however, once scaled up (or down), the limits of the supporting tools and analysis may be exceeded. This year we will explore the tools and analysis methods (existing or proposed) which an engineer needs in order to do these "scaled problems." Also of interest is how to make these tools work best together, and problems to exercise these tools or drive the development of new ones. We also encourage appropriate ideas outside of this scope, since we advocate this diverse seminar series to stay abreast of current research trends.

A full listing of this project-year's speakers is below. Most talks can be downloaded from the seminar website, at

http://chess.eecs.berkeley.edu/seminar.htm

- "A Categorical Theory of Hybrid Systems"
  Aaron D. Ames, UC Berkeley, May 9, 2006

- "The Role of Control in Design: from Fixing Problems to the Design of Dynamics"
  Andrzej Banaszuk, United Technologies Research Center, April 25, 2006
- "Autonomous Rotorcraft Landing Using Computer Vision"
  Todd Templeton, UC Berkeley, April 18, 2006
- "Modular Performance Analysis and Interface-Based Design for Real-Time Systems"
  Ernesto Wandeler, ETH Zürich, April 11, 2006.
- "Semantic Interpretation of Causal Timed Systems"
  Eleftherios Matsikoudis, UC Berkeley, March 21, 2006
- "Causality Interfaces"
  Rachel Zhou, UC Berkeley, March 7, 2006.
- "Semantic Foundation of the Tagged Signal Model"
  Xiaojun Liu, UC Berkeley/Xilinx, February 22, 2006.
- "HOPES: Embedded Software Development for MPSoC"
  Soonhoi Ha, Seoul National University, February 14, 2006.
- "The JAviator Project"
  Christoph Kirsch, University of Salzburg, February 7, 2006.
- "Theoretical and Practical Challenges of LXI and IEEE 1588 Measurement Systems"
  John C. Eidson, Agilent Technologies, January 24, 2006.
- "Observability of Hybrid Systems and applications to ATM"
  Alessandro D'Innocenzo, University of L'Aquila, December 6, 2005.
- "Achievable Bisimilar Behaviours of Abstract State Systems"
  Giordano Pola, University of L'Aquila, December 6, 2005.
- "Approximation Metrics for Discrete, Continuous and Hybrid Systems"
  Antoine Girard, University of Pennsylvania, November 29, 2005.
- "Semantics-Based Optimization Across Uncoordinated Tasks in Networked Embedded Systems"
  Elaine Cheong, UC Berkeley, November 15, 2005.
- "Programmable Internet Environment"
  Sinisa Srbljic, University of Zagreb, November 8, 2005.
- "The Teja Model of Computation"
  Akash Deshpande, Teja Technologies, November 1, 2005.
- "Semantic Anchoring"
  Ethan Jackson, Vanderbilt University, October 18, 2005.
- "Reasoning about Timed Systems Using Boolean Methods"
  Sanjit A. Seshia, UC Berkeley, October 11, 2005.
- "The Ultraconcurrency Revolution in Hardware and Software"
  Carl Hewitt, MIT, October 4, 2005.

- "Causality Interfaces and Compositional Causality Analysis"
  Haiyang Zheng, UC Berkeley, September 20, 2005.
- "A Structural Approach to Quasi-Static Schedulability Analysis of Communicating Concurrent Programs"
  Cong Liu, UC Berkeley, September 13, 2005.
- "Tools for the Simulation, Verification and Synthesis of Hybrid Systems"
  Alessandro Pinto, UC Berkeley, September 6, 2005.
- "Optimal Scheduling of Acyclic Branching Programs on Parallel Machines"
  Oded Maler, CNRS-Verimag, Grenoble, France, August 30, 2005.
- "Coordinated Component Composition"
  Farhad Arbab, CWI, Amsterdam and Leiden University, August 25, 2005.
- "Mode Transition Behavior in Hybrid Dynamic Systems"
  Pieter J. Mosterman, The Mathworks, August 16, 2005.
- "On the Complexity of Multi-Modal Control Procedures"
  Magnus Egerstedt, Georgia Institute of Technology, August 1, 2005.
- "Promoting reuse and repurposing on the Semantic Grid"
  Antoon Goderis, University of Manchester, July 19, 2005.

### 3.4.4. Workshops and Invited Talks

In addition to the below invited and workshop organizational activities, Chess faculty have delivered numerous plenary talks, invited talks, as well as informal dissemination of Chess goals and research.

***Hybrid and Embedded Systems: Technologies and Applications***
February 21-22, 2006: CHESS researchers, Profs. Gabor Karsai, T. John Koo, Shankar Sastry and Janos Sztipanovits, were invited to give lectures at the workshop on Hybrid and Embedded Systems: Technologies and Applications, which was jointly organized by the Hong Kong Science and Technology Parks Corporation and the Department of Automation and Computer-aided Engineering, the Chinese University of Hong Kong, held at the Hong Kong Science and Technology Parks.

The Workshop was proposed and chaired by Prof. T. John Koo of Vanderbilt University and Prof. Yeung Yam of CUHK for promoting HEMS related technologies and applications in Hong Kong. The welcome speeches of the workshop were given by Ir S. W. Cheung. Vice President of HKSTP and Prof. Billy K.-L. So of CUHK.

Chess researchers also participated in an Open Forum moderated by the Dr. OnChing Yue, Science Advisor of the Innovation and Technology Commission, the Government of the Hong Kong Special Administrative Region, on many important issues ranging from the scientific research to application development of Hybrid and Embedded Systems in the Greater China Region.

The workshop website:

http://www.acae.cuhk.edu.hk/HEMS2006/

The news appears in

http://chess.eecs.berkeley.edu/

and

Faculty and Staff Notes, Vanderbilt Register, March 13, 2006
http://www.vanderbilt.edu/register/articles?id=25215

On February 22, Profs. Gabor Karsai, T. John Koo and Janos Sztipanovits also visited the Hong Kong Applied Science and Technology Research Institute (ASTRI) and had a meeting with, Dr. Shen-Chang Chao, Vice President of Enterprise and Consumer Electronics Group at ASTRI, who is responsible for research and development in innovative technologies and applications for digital home, multimedia communications and pervasive services.

***International Embedded Systems Forum***

On 23-24 February, 2006, Profs. Gabor Karsai, T. John Koo, and Janos Sztipanovits of Vanderbilt University were invited by the Shantou University (a university heavily subsidised by the Li Ka Shing Foundation) to give lectures at the International Embedded Systems Forum at the Shantou University. They had meetings with the Vice President (Research and Academic) Prof. Peihua Gu, the Assistant to Director of Li Ka Shing Foundation, Eric Chow, and other faculties. They were briefed on various university reforms initiated by the Li Ka Shing Foundation and the Shantou University, and exchanged ideas on possible collaboration with both the Foundation and the University in embedded systems research.

***5th OOPSLA Workshop on Domain-Specific Modeling***

During the OOPSLA Conference in October of 2006, Dr. Jonathan Sprinkle participated in, and helped to organize, the 5th OOPSLA Workshop on Domain-Specific Modeling. Domain-Specific Modeling aims at raising the level of abstraction beyond programming by specifying the solution directly using domain concepts. In a number of cases the final products can be generated from these high-level specifications. This automation is possible because of domain-specificity: both the modeling language and code generators fit to the requirements of a narrow domain only, often in a single company. This is the fifth workshop on Domain-Specific Modeling, following the encouraging experiences from the earlier workshops at past OOPSLA conferences (Tampa 2001, Seattle 2002, Anaheim 2003 and Vancouver 2004). During the time the DSM workshops have been organized, interest in domain-specific modeling languages, metamodeling and supporting tools has seen a revival. The electronic version of the proceedings, presentation slides and group work results is available at

http://www.dsmforum.org/events/

***ARTEMIS 2006 Annual Conference***

Edward A. Lee was invited to present a talk on the future of embedded software at the ARTEMIS 2006 Annual Conference that occurred May 22-24, 2006 in Graz, Austria, for its 3rd annual conference, the European Technology Platform

'ARTEMIS'- Advanced Research and Technology for Embedded Intelligence and Systems - welcomed all stakeholders in the domain of embedded systems. Other participants from Daimler-Chrysler, Esterel Technologies, STMicroelectronics, and various universities presented talks on economic impact of embedded systems, design methods and tools, reference designs and architectures as well as a panel discussion on the future of embedded computing, the subject of Prof. Lee's presentation.

***OSD Workshops on the Software Producibility Initiative***
In Rosslyn, VA, in May 2006, Prof. Shankar Sastry briefed executives from Boeing, Raytheon, Lockheed Martin, Sikorsky, and other major defense acquisition companies, in addition to the Navy, Air Force, and Army procurement offices, on the state of affairs in research problems in software-intensive systems, and how interaction with the research community is necessary to avert potential problems in systems complexity. This briefing was preceded by a workshop detailing the current research and application gap, and was a continuation of a workshop held in Berkeley in August 2006.

### 3.4.5. General Dissemination

The Chess website, http://chess.eecs.berkeley.edu, includes publications and software distributions. In addition, as part of the outreach effort, the UC Berkeley introductory signals systems course, which introduces hybrid systems, is available at http://ptolemy.eecs.berkeley.edu/eecs20/ and Ptolemy II software is available at http://ptolemy.eecs.berkeley.edu.

The ISIS website, http://www.isis.vanderbilt.edu, makes publications and software available.

## 3.5. Other Specific Product

The following software packages have been made available during this review period on the Chess website, http://chess.eecs.berkeley.edu:

- The Generic Modeling Environment (GME 5) is a configurable toolkit for creating domain-specific modeling and program synthesis environments. The configuration is accomplished through metamodels specifying the modeling paradigm (modeling language) of the application domain. The modeling paradigm contains all the syntactic, semantic, and presentation information regarding the domain; which concepts will be used to construct models, what relationships may exist among those concepts, how the concepts may be organized and viewed by the modeler, and rules governing the construction of models. The modeling paradigm defines the family of models that can be created using the resultant modeling environment. The latest release (11/18/05) can be found at: http://www.isis.vanderbilt.edu/projects/gme/.
- GReAT (Graph Rewriting And Transformation) is a component technology of GME compriseed of a metamodel based graph transformation language useful for the specification and implementation of model-to-model transformations. The most recent release can be downloaded from: http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#GREAT
- Universal Data Model (UDM) Generates C++ API from UML class diagrams. The API can be used to read/write XML files, GME databases, etc. and is component technology for Graph Rewriting And Transformation (GReAT). The most recent release can be downloaded from http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#UDM
- Ptplot is a 2D signal plotter implemented in Java. Ptplot can be used in a standalone applet or application or used in your own applet or application. PtPlot 5.5. UC Berkeley,

28 July, 2005 is available as part of Ptolemy or as a standalone download at http://ptolemy.eecs.berkeley.edu/java/ptplot

- HyVisual 5.0.1 is a hybrid system visual modeler, was released in October 2005. This visual modeler supports construction of hierarchical hybrid systems. It uses a block-diagram representation of ordinary differential equations (ODEs) to define continuous dynamics. It uses a bubble-and-arc diagram representation of finite state machines to define discrete behavior. HyVisual UC Berkeley 5.0.1 October, 2005 is available at http://ptolemy.eecs.berkeley.edu/hyvisual/index.htm

- Ptolemy II 5.0.1 beta is a set of Java packages supporting heterogeneous, concurrent modeling and design. Its kernel package supports clustered hierarchical graphs, which are collections of entities and relations between those entities. Its actor package extends the kernel so that entities have functionality and can communicate via the relations. Its domains extend the actor package by imposing models of computation on the interaction between entities. Examples of models of computation include discrete-event systems, data flow, process networks, synchronous/reactive systems, and communicating sequential processes. Ptolemy II includes a number of support packages, such as data, providing a type system, data encapsulation and an expression parser, plot, providing visual display data, math, providing matrix and vector math and signal processing functions, and graph, providing graph-theoretic manipulations. Ptolemy II 5.0.1. UC Berkeley, October, 2005 release is available at http://ptolemy.eecs.berkeley.edu/ptolemyII/

- CIL is a front-end for the C programming language that facilitates program analysis and transformation. CIL will parse and typecheck a program, and compile it into a simplified subset of C. For example, in CIL all looping constructs are given a single form and expressions have no side-effects. This reduces the number of cases that must be considered when manipulating a C program. CIL has been used for a variety of projects, including CCured, a tool that makes C programs memory safe. CIL supports ANSI C as well as most of the extensions of the GNU C and Microsoft C compilers. A Perl script acts as a drop in replacement for either gcc or Microsoft's cl, and allows merging of the source files in your project. Other features include support for control-flow and points-to analyses. CIL Version 1.3.4, UC Berkeley May 2005 release is available at http://cil.sourceforge.net.

- The Ellipsoidal Toolbox is a standalone set of easy-to-use configurable MATLAB routines to perform operations with ellipsoids and hyperplanes of arbitrary dimensions. It computes the external and internal ellipsoidal approximations of geometric (Minkowski) sums and differences of ellipsoids, intersections of ellipsoids and intersections of ellipsoids with halfspaces and polytopes; distances between ellipsoids, between ellipsoids and hyperplanes, between ellipsoids and polytopes; and projections onto given subspaces. Ellipsoidal methods are used to compute forward and backward reach sets of continuous- and discrete-time piecewise affine systems. Forward and backward reach sets can be also computed for continuous-time piece-wise linear systems with disturbances. It can be verified if computed reach sets intersect with given ellipsoids, hyperplanes, or polytopes. The toolbox provides efficient plotting routines for ellipsoids, hyperplanes and reach sets

ET version 1.03 released January 2006, available at www.eecs.berkeley.edu/~akurzhan/ellipsoids

- Viptos (Visual Ptolemy and TinyOS) is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. Viptos allows developers to create block and arrow diagrams to construct TinyOS programs from any standard library of nesC/TinyOS components. The tool automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target hardware. In particular, Viptos includes the full capabilities of VisualSense, which can model communication channels, networks, and non-TinyOS nodes. Viptos is compatible with nesC 1.2 and includes tools to harvest existing TinyOS components and applications and convert them into a format that can be displayed as block (and arrow) diagrams and simulated Viptos 5.1-alpha was released November 2005 and is available at http://ptolemy.eecs.berkeley.edu/viptos/

- Prospector: a programmer's search engine based on jungloid mining: Software developers try to reuse existing code when possible, but reuse is often difficult because APIs are complex and the required client code to use the API can be hard to write. We identify a common type of problematic client code, jungloids, which are unary expressions. We describe a simple query language programmers can use to find jungloids, and search techniques that use both type signatures and examples to answer queries. We implemented a prototype jungloid search tool, Prospector, based on these techniques. In a test of query processing accuracy, Prospector found the desired jungloid for 17 of 20 queries. We also evaluated Prospector in a pilot user study and found evidence that it helps programmers reuse libraries faster and more reliably. Prospector is available at http://www.cs.berkeley.edu/~bodik/research/prospector.html

- SKETCH is a sketching system based on combinatorial search, as opposed to transformations. In this system, the sketch is given in the form of a partial program--a program with holes--and the sketch resolution synthesizes code to fill in the holes. The holes may stand for index expressions, lookup tables or bitmasks, and the programmer can easily define new kinds of holes with the synthesis operators provided. SKETCH completes sketches by means of a combinatorial search based on generalized boolean satisfiability. SKETCH is available at http://www.cs.berkeley.edu/~asolar/SketchProjects.htm

- We have designed and implemented a new programming language for hard real-time systems. Critical timing constraints are specified within the language, and ensured by the compiler. The main novel feature of the language is that programs are extensible in two dimensions without changing their timing behavior: new program modules can be added, and individual program tasks can be refined. The mechanism that supports time invariance under parallel composition is that different program modules communicate at specified instances of time. Time invariance under refinement is achieved by conservative scheduling of the top level. The language, which assembles real-time tasks within a hierarchical module structure with timing constraints, is called Hierarchical Timing Language (HTL). It is a coordination language, in that individual tasks can be implemented in other languages. We present a distributed HTL implementation of an

automotive steer-by-wire controller as a case study. HTL is available at http://htl.cs.uni-salzburg.at/

# 4. Contributions

This section summarizes the major contributions during this reporting period.

## 4.1. Within Discipline

### 4.1.1. Hybrid Systems Theory

- We have worked with our definition of an operational semantics for hybrid systems in the current and next generation of toolsets to reflect these semantics.
- We have developed algorithms for computing the real value of discounted properties, and continued investigation of their application.
- We have matured a theory of a homology theory of hybrid systems which enables elegant characterization of Zeno and other qualitative properties of hybrid systems.
- We have improved on the best known algorithms for finding strategies for the control of stochastic hybrid systems.
- We have continued development of a toolbox using ellipsoidal methods to calculate reach sets for linear dynamic systems, and begun to apply those to hybrid systems.
- We have developed an extensive theory of two and multi person stochastic games with extensions of notions of safety and almost safety in a number of important directions.
- We have continued to apply and study stochastic hybrid systems within the domain of biological systems.
- We are developing a static analysis mechanism that infers the common causality properties of a modal model from those of its modes. The result of the static analysis is conservative, but provides safety guarantees.
- We have continued in our broad initiative to support toolchains in hybrid systems under semantic anchoring and model transformations.
- We derived verifiable necessary and sufficient conditions on when composition preserves semantics for a heterogeneous network of embedded systems.
- We have formally proved the benefits of the logical execution time (LET) model in terms of composability over traditional real-time models.
- We have developed a technique to extend the simulation of a hybrid system past its Zeno point, reducing the computational burden past that point and revealing the complete behavior of the system.

### 4.1.2. Model-Based Design

- We have developed the first release of a semantic anchoring tool suite, and have demonstrated the use of the tool infrastructure in specifying the semantics of hierarchical state automata.

- Using various specifications of timed automata, we have examined approaches for defining semantic units. We demonstrated the concepts with developing a semantic unit for timed automata and showed the anchoring of UPAAL and IF to this common semantic unit.

- We started investigating the problems of defining semantics for heterogeneous modeling languages, and began establishing a composition theory for semantic units.

- Applying our ongoing work on metamodeling, we have continued development on semantic anchoring for model-based development. Specifically, we have extended the semantic anchoring framework to heterogeneous behaviors.

- We have continued to demonstrate our defined agent algebras as a formal framework for uniformly representing and reasoning about models of computation used in the design of hybrid and embedded software systems.

- We have continued to demonstrate our theoretical and compositional framework for reasoning about causality in components which are composed under concurrent models of computation.

- We have extended our previously developed tagged-signal model for concurrent models of computation to represent the semantics of globally asynchronous, locally synchronous systems built upon loosely time-triggered architectures.

- We have continued to maintain a language and a suite of supporting tools for the specification of model transformations based on graph rewriting.

- We have continued to use our approach to model synthesis based on patterns specified formally as metamodels.

- We have developed an interface theory based approach to static analysis of actor models through composition. It results in an automaton which will contain information used for further static analysis of a composed actor model.

- We have developed a new component model for timed models of computation such as discrete event, continuous time, hybrid systems, and synchronous/reactive models.

- We have built a scalable and formal specification language for embedded systems which can use constraint checking to auto-generate parts of a specification and to approximate the correctness of the specification without invoking verification tools

### 4.1.3. Advanced Tool Architectures

- We have further developed the code generation approach based on component specialization by developing a formal framework for reasoning about reconfiguration in embedded software.

- We have continued to improve the performance and feature set of the Metropolis framework.

- We have further developed our notion of interface theories to support reasoning about heterogeneous component composition and about the dynamics of models of computation.

- We formulated and solved the task allocation problem for a popular multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor.

- We have continued to investigate interests in fault-tolerant systems by developing new modeling languages which simulate and trace faults in a system.

- We have continued development of the Ptolemy II toolsuite, including Hyvisual, VisualSense, and Viptos tools for hybrid systems, sensor networks, and NesC-based wireless sensor programming.

- We have shown how to guarantee type-safety in legacy C programs and verify memory safety in the assembly code.

- We have strengthened our understanding of discounted reward objectives to yield real-numbered quantities (e.g., power consumption) that can be expressed during verification.

### 4.1.4. Experimental Research

- We have extended model predictive control for hybrid systems with a finite control set to develop air and water recovery systems for the NASA Advanced Life Support (ALS) system for long-duration missions.

- We have begun to apply our previous work on safe set calculations to the Autonomous Aerial Refueling (AAR) while in formation problem.

- We have deployed the Metropolis platform-based design methodology for use on various veitronics problems of interest to Toyota, GM, and BMW.

- We have continued development, and deployed a modeling environment for wireless sensor networks. These have been used to simulate detection of a dirty bomb.

- We have developed new programming models for sensor networks that build on the popular TinyOS models.

- We have shown how compositional technologies can be used to produce an autonomous helicopter in the loop with a camera to choose a landing zone, and physically land the vehicle.

- We have used reachability to perform analysis of the cold start problem and shown anticipated reduction in raw hydrocarbon emissions during warm-up using a hybrid systems model.

- We have shown how fault tolerant data flow can be used to synthesize real-time feedback controllers for safety critical applications.

- We have shown that hybrid systems theory can be coupled with Lagrangian methods to produce reduced state-space expressions of computationally difficult problems, such as the motion of a bipedal walker.

## 4.2. Other Disciplines

- We developed new efficient algorithms for solving stochastic games, which have applications in other fields such as economics and biology.

- We contributed to scientific interdisciplinary information sharing through collaboration and major contribution to the framework of the Kepler Scientific Workflow project.

- We have shown that hybrid systems theory can be coupled with Lagrangian methods to produce reduced state-space expressions of computationally difficult problems, such as the motion of a bipedal walker.

## 4.3. Human Resource Development

Several panels in important conferences and workshops pertinent to embedded systems (e.g., DAC, ICCAD, HSCC, EMSOFT, CASES, and RTSS) have pointed out the necessity of upgrading the talents of the engineering community to cope with the challenges posed by the next generation embedded system technology. Our research program has touched many graduate students in our institutions and several visiting researchers from industry and other Universities so that they now have a deep understanding of embedded system software issues and techniques to address them.

Specifically, our directors played a major role in the development of workshops and briefings to executives and researchers in the avionics industry to motivate increased research spending due to an anticipated drop in research funds available to train graduates in embedded software and embedded systems. One particular intersection with our efforts is the Software Producibility Initiative out of the Office of the Secretary of Defense.

The industrial affiliates to our research program are increasing and we hope to be able to export in their environments a modern view of system design. Preliminary feedback from our partners has underlined the importance of this process to develop the professional talent pool.

## 4.4. Integration of Research and Education

In this report, we have touched multiple times on research and education especially in the outreach section. In addition, there has been a strong activity in the continued update of the undergraduate course taught at Berkeley on the foundations of embedded system design. The graduate program at Berkeley and at Vanderbilt has greatly benefited from the research work in the ITR. EE249 at Berkeley has incorporated the most important results thus far obtained in the research program. EE 290 A and C, advanced courses for PhD students, have featured hybrid system and the interface theories developed under this project. EE219C, a course on formal verification, has used results from the hybrid theory verification work in the program. Finally, many final projects in these graduate courses have resulted in papers and reports listed in this document. The course EE291E on Hybrid Systems: Computation and Control is jointly taught at Berkeley and Vanderbilt and is benefiting a great deal from comments of students as far as the development of new text book material.

In addition to the influence on graduate students, we have endeavored to show hybrid and embedded systems as emerging research opportunities to undergraduates. We have also demonstrated that for advanced undergraduates these topics are not out of place as senior design courses, or advanced topics courses, which may in the future lead to the integration of these as disciplines in engineering across a broader reach of universities.

Due to the benefit of a large center, we were able to use the model CHESS espoused in developing the summer program for SUPERB in influencing how the remainder of the program would be run this year, paying special attention to defining an undergraduate research project which could then be matured by a graduate student. This year the SUPERB program produced a two-semester undergraduate course which attracted electrical and mechanical engineering undergraduates to use hybrid systems theory for application in bipedal walking. In addition, there were several conference papers written which are in publication at this time; these included the undergraduates as authors.

## 4.5. Beyond Science and Engineering

Embedded systems are part of our everyday life and will be much more so in the future. In particular, wireless sensor networks will provide a framework for much better environmental monitoring, energy conservation programs, defense and health care. Already in the application chapter, we can see the impact of our work on these themes. In the domain of transportation systems, our research is improving safety in cars, and foundationally improving control of energy conserving aspects such as hydrocarbon emissions. Future applications of hybrid system technology will involve biological systems to a much larger extent showing that our approach can be exported to other field of knowledge ranging from economics to biology and medicine. At Berkeley, the Center for Information Technology Research in the Interest of Society is demonstrating the potential of our research in fields that touch all aspects of our life. Some key societal grand challenge problems where our ITR research is making a difference includes health care delivery, high confidence medical devices and systems, avionics, cybersecurity, and transportation.