FINAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-0225610

UNIVERSITY OF CALIFORNIA, BERKELEY

November 26, 2010

PERIOD OF PERFORMANCE COVERED: November 14, 2002 – August 31, 2010

Contents

1 Participants **3**
1.1 People . . . 3
1.2 Partner Organizations: . . . 9
1.3 Collaborators: . . . 9

2 Activities and Findings **14**
2.1 Project Activities . . . 14
2.1.1 ITR Events . . . 16
2.1.2 Hybrid Systems Theory . . . 16
2.1.3 Deep Compositionality . . . 16
2.1.4 Robust Hybrid Systems . . . 16
2.1.5 Hybrid Systems and Systems Biology . . . 16
2.2 ProjectFindings . . . 16

3 Outreach **16**
3.1 Project Training and Development . . . 16
3.2 Outreach Activities . . . 16
3.2.1 Curriculum Development for Modern Systems Science (MSS) . . . 16
3.2.2 Undergrad Course Insertion and Transfer . . . 16
3.2.3 Graduate Courses . . . 16

4 Publications and Products **16**
4.1 Technical reports . . . 16
4.2 PhD theses . . . 16
4.3 PhD theses . . . 16
4.4 Conference papers . . . 16
4.5 Books . . . 16
4.6 Journal articles . . . 16
4.6.1 The 2009-2010 Chess seminar series . . . 16
4.6.2 Workshops and Invited Talks . . . 16
4.6.3 General Dissemination . . . 16
4.7 Other Specific Products . . . 16

5 Contributions **16**
5.1 Within Discipline . . . 16
5.1.1 Hybrid Systems Theory . . . 16
5.1.2 Model-Based Design . . . 16
5.1.3 Advanced Tool Architectures . . . 16
5.1.4 Experimental Research . . . 16
5.2 Other Disciplines . . . 16
5.3 Human Resource Development . . . 16

5.4 Integration of Research and Education 16
5.5 Beyond Science and Engineering 16

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)
CLAIRE TOMLIN (UC BERKELEY, EECS)
FACULTY INVESTIGATORS
ALEX AIKEN (UC BERKELEY, CS)
AHMAD BAHAI (UC BERKELEY, EECS)
RUZENA BAJCSY (UC BERKELEY, EECS)
ALEXANDRE BAYEN (UC BERKELEY, CIVIL ENGINEERING)
GAUTAM BISWAS (VANDERBILT, CS)
RASTISLAV BODIK (UC BERKELEY, EECS)
BELLA BOLLOBAS (MEMPHIS, MATHEMATICS)
JEROME A. FELDMAN (UC BERKELEY)
KENNETH FRAMPTON (VANDERBILT, ME)
WAGDY H. MAHMOUD (TENNESSEE TECH. UNIVERSITY)
J. KARL HEDRICK (UC BERKELEY, ME)
GABOR KARSAI (VANDERBILT, ECE)
KURT KEUTZER (UC BERKELEY, EECS)
T. JOHN KOO (VANDERBILT)
GEORGE NECULA (UC BERKELEY, EECS)
SRINI RAMASWAMY (TENNESSEE TECH. UNIVERSITY)
CARLO SEQUIN (UC BERKELEY, EECS)
MASAYOSHI TOMIZUKA (UC BERKELEY, ME)
POST DOCTORAL RESEARCHERS:
ALVARO CARDENAS (UC BERKELEY)
MASSIMO FRANCESHETTI (UC BERKELEY)
CHRISTOPH KIRSH (UC BERKELEY)
AKOS LEDECZI (VANDERBILT)
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)
ANIL ASWANI (UC BERKELEY)
DANIEL BALASUBRAMANIAN (UC BERKELEY, VANDERBILT)
CHRIS BEERS (VANDERBILT)
JAN BIERMEYER (UC BERKELEY)
LUCA CARLONI (UC BERKELEY)
JOSE CARLOS ZAVALA (UC BERKELEY)
ADAM CATALDO (UC BERKELEY)
ARINDAM CHAKRABARTI (UC BERKELEY)
YOUNG-HWAN CHANG (UC BERKELEY)
DENNIS CHANG (UC BERKELEY)
KRISHNENDU CHATTERJEE (UC BERKELEY)
KAI CHEN (VANDEREBILT)
ELAINE CHEONG (UC BERKELEY)
ANDREW D. DIXON (VANDERBILT)
ABHIJIT DAVARE (UC BERKELEY)
DOUGLAS DENSMORE (UC BERKELEY)
ANDREW DIXON (VANDERBILT)
MILOS DREZGIC (UC BERKELEY)
BRANDON EAMES (VANDERBILT)
MATTHEW EMERSON (VANDERBILT)
HUINING FENG (UC BERKELEY)
AARONG FLEETWOOD (VANDERBILT)
JOYTI GANDHE (VANDERBILT)
SUMITRA GANESH (UC BERKELEY)
CSABA GARAY (VANDERBILT)
ARKEDEB GHOSAL (UC BERKELEY)
MATTHEW HARREN (UC BERKELEY)
GRAHAM HEMMINGWAY (VANDERBILT)
THOMAS HUINING FENG (UC BERKELEY)
SLEIMAN ITANI (UC BERKELEY)
MATTHEW J. EMERSON (VANDERBILT)
ETHAN JACKSON (VANDERBILT)
FARINAZ KOUSHANFAR (UC BERKELEY)
NARAYANAN KRISHNAN (UC BERKELEY)
ALEXANDER KURZHANSKIY (UC BERKELEY)
MANISH KUSHWAHA (VANDERBILT)
BRANISLAV KUSHY (VANDERBILT)
JONGHO LEE (UC BERKELEY)
CHRISTINA LEE (VANDERBILT)
XIAOJUN LIU (UC BERKELEY)

GABOR MADL (VANDERBILT)
SLOBODAN MATIC (UC BERKELEY)
ELEFTERIOUS 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)
DAVID P. MANDELIN (UC BERKELEY)
ALESSANDRO PINTO (UC BERKELEY)
WILLIAM PLISHKER (UC BERKELEY)
VINAYAK PRABHU (UC BERKELEY)
BENJAMIN R. LIBLIT (UC BERKELEY)
KAUSHIK RAVINDRAN (UC BERKELEY)
JANOS SALLAI (VANDERBILT)
PANNAG SANKETI (UC BERKELEY)
TRIPTI SAXENA (VANDERBILT)
PETER SCHMIDT (VANDERBILT)
TIVADAR SZEMETHY (VANDERBILT)
RYAN T THIBODEAUX (VANDERBILT)
TAO TAO (VANDERBILT)
TODD TEMPLETON (UC BERKELEY)
AMBUJ TEWARI (UC BERKELEY)
GUOGIANG WANG (UC BERKELEY)
GUANG YANG (UC BERKELEY)
YANG YANG (UC BERKELEY)
INSOON YANG (UC BERKELEY)
JAMES YEH (UC BERKELEY)
HAIBO ZENG (UC BERKELEY)
YANG ZHAO (UC BERKELEY)
HAIYANG ZHENG (UC BERKELEY)
WEI ZHENG (UC BERKELEY)
GANG ZHOU (UC BERKELEY)
QI ZHOU (UC BERKELEY)
YE ZHOU (UC BERKELEY)

UNDERGRADS:

DANIEL BALASUBRAMANIAN
PHILIP BALDWIN (UC BERKELEY)
COLIN COCHRAN (UC BERKELEY)
NICKOLIA COOMBS (VANDERBILT)
RACHAEL DENNISON (VANDERBILT)

DAVID GARCIA (VANDERBILT)
MARCUS GRALY (UC BERKELEY)
SHANTEL HIGGINS (VANDERBILT)
JOHN KILBY (VANDERBILT)
EFOSA OJOMO (VANDERBILT)
MIKE OKYERE (UC BERKELEY)
RAKESH REDDY (UC BERKELEY)
DEVON REED (UC BERKELEY)
MICHAEL RIVERA-JACKSON (VANDERBILT)
ISMAEL SARMIENTO (UC BERKELEY)
BINA SHAH (VANDERBILT)
EDWIN VARGAS (VANDERBILT)
TRIONE VINCENT (VANDERBILT)
ANTONIO YORDAN-NONES (UC BERKELEY)
DANIEL BALASUBRAMANIAN
NICKOLIA COOMBS (VANDERBILT)
RACHAEL DENNISON (VANDERBILT)
DAVID GARCIA (VANDERBILT)
SHANTEL HIGGINS (VANDERBILT)
JOHN KILBY (VANDERBILT)
EFOSA OJOMO (VANDERBILT)
MICHAEL RIVERA-JACKSON (VANDERBILT)
BINA SHAH (VANDERBILT)
EDWIN VARGAS (VANDERBILT)
TRIONE VINCENT (VANDERBILT)
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)
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)
ZULHIMI AHMAD (VANDERBILT)
ANGELINE BROWN (VANDERBILT)
LOUISE COLLINS (VANDERBILT)
DOMINIQUE DUNCAN (UC BERKELEY/UNIVERSITY OF CHICAGO)
TARRELL EZELL (VANDERBILT)
DANIELLE JONES (VANDERBILT)
CHARLES LEFONT (VANDERBILT)
KENNETH MCPHERSON (VANDERBILT)
NANDITA MITRA (UC BERKELEY/RUTGERS)
NASHLIE SEPHUS (UC BERKELEY/MISSISSIPPI STATE)
HEATHER TAYLOR (UC BERKELEY/UNIVERSITY OF VERMONT)
STEPHANIE WRIGHT (VANDERBILT)

STAFF:

GYORGY BALOGH (VANDERBILT)
CHRISTOPHER BROOKS (UC BERKELEY)
JESSICA GAMBLE (UC BERKELEY)
ALLEN HOPKINS (UC BERKELEY)
CHRISTOPHER HYLANDS (UC BERKELEY)
NATHAN JEW (UC BERKELEY)
PHILLIP LOARIE (UC BERKELEY)
BRADLEY A. KREBS (UC BERKELEY)
NEAL MASTER (UC BERKELEY)
ZOLTAN MOLNAR (VANDERBILT)
MARVIN MOTLEY (UC BERKELEY)
GUNNAR PROPPE (UC BERKELEY)
DAN STEWARD (VANDERBILT)
MARY STEWART (UC BERKELEY)
NEIL TURNER (UC BERKELEY)
AARON WALBURG (UC BERKELEY)
BRIAN WILLIAMS (UC BERKELEY)
BRIAN WILLIAMS (VANDERBILT)
ZHAOQIAN XIA (UC BERKELEY)

BUSINESS ADMINISTRATOR:

ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)
MICHELE M. CODD (VANDERBILT)
GLADYS KHOURY (UC BERKELEY)
SUSAN GARDNER (UC BERKELEY)
TRACEY RICHARDS (UC BERKELEY)

EXECUTIVE DIRECTORS:

CHRISTOPHER BROOKS (UC BERKELEY)
JONATHAN SPRINKLE (UC BERKELEY)

1.2 Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY
VANDERBILT UNIVERSITY
UNIVERSITY OF MEMPHIS

1.3 Collaborators:

PIETER ABBEEL (UC BERKELEY)
NORIYASU ADACHI (TOYOTA TECHNICAL CENTER)
ILKAY ALTINTAS (SDSC)
AARON AMES (CALTECH)
SAURABH AMIN (UC BERKELEY)
HUGO A. ANDRADE (NATIONAL INSTRUMENTS)
ANIL ASWANI (STANFORD UNIVERSITY)
JEFF AXELROD (STANFORD UNIVERSITY)
FELICE BALARIN (CADENCE BERKELEY LABORATORIES)
PHILIP BALDWIN (2 MITCHELL, ECE)
SHAMIK BANDYOPADHYAY (UC BERKELEY)
ALBERT BENVENISTE (INRIA, IRISA)
CHRISTIAN BERGER (RWTH AACHEN, GERMANY)
CHAD BERKLEY (UC SANTA BARBARA)
DIRK BEYER (EPFL, SIMON FRASER UNIVERSITY)
PETER BICKEL (UC BERKELEY)
MARK BIGGIN (LBNL)
THOMAS BRIHAYE (UNIVERSITE DE MONS-HAINAUT, LSV-CNRS & ENS DE CACHAN, FRANCE)
JAMES BROWN (UC BERKELEY)
JEFF BURCH (AGILENT)
BENOIT CAILLAUD (INRIA, IRISA)
LUCA CARLONI (COLUMBIA UNIVERSITY)
PAUL CASPI (VERIMAG)
MINGHUA CHEN (MICROSOFT RESEARCH)
XI CHEN (NOVAS SOFTWARE)
XI CHEN (UC RIVERSIDE)
HENRIK CHRISTENSEN (ROYAL INSTITUTE OF TECHNOLOGY, STOCKHOLM, SWEDEN)
JASON CONG (UNIVERSITY OF CALIFORNIA, LOS ANGELES)
DAVID E. CORMAN (BOEING PHANTOM WORKS)

MASSIMILIANO D'ANGELO (UNIVERSITY OF L'AQUILA AND PARADES GEIE)
ALESSANDRO D'INNOCENZO (UNIVERSITA' DELL'AQUILA, UNIVERSITY OF PENNSYLVANIA)
LUCA DE ALFARO (UC SANTA CRUZ)
DOUGLAS DENSMORE (UC BERKELEY)
MARIKA DI BENEDETTO (UNIVERSITY OF L'AQUILA)
MARCO DI NATALE (GENERAL MOTORS RESEARCH, SCUOLA SUPERIORE SANT'ANNA)
MARIA DOMENICA DI BENEDETTO (UNIVERSITA' DELL'AQUILA)
ADAM DONLIN (XILINX RESEARCH)
LAURENT DOYEN (EPFL)
KEMAL EBCIOUGLU (IBM)
STEPHEN EDWARDS (COLUMBIA UNIVERSITY)
J. MIKAEL EKLUND (UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY)
JOHN EIDSEN (AGILENT)
MIKE EISEN (LBNL)
LAURENT EL GHAOUI (UC BERKELEY)
ROB ENNALS (INTEL RESEARCH, CAMBRIDGE, UK)
MARCO FAELLA (UC SANTA CRUZ)
CARLO FISCHIONE (UC BERKELEY)
DAVID GAY (INTEL RESEARCH BERKELEY)
ANNARITA GIANI (UC BERKELEY)
ANOUCK GIRARD (COLUMBIA UNIVERSITY, ME)
PAOLO GIUSTO (GM RESEARCH)
ANTOON GODERIS (UNIVERSITY OF MANCHESTER)
ANIRUDDA GOKHALE (VANDERBILT UNIVERSITY)
DENIS GOPAN (UNIVERSITY OF WISCONSIN)
JEFF GRAY (UNIVERSITY OF ALABAMA (BIRMINGHAM))
JEFF GRAY (VANDERBILT UNIVERSITY)
JOEL GRAY (LAWRENCE BERKELEY NATIONAL LABORATORY)
ROBERT D. GREGG (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
ESTEN GROTLI (UNIVERSITY OF TRONDHEIM)
BRUCE HAMILTON (AGILENT)
FALK HANTE (UNIVERSITY OF ERLANG)
JENS HARNISCH (INFINEON)
DAN HIGGINS (UC SANTA BARBARA)
MARK HILL (UNIVERSITY OF WISCONSIN)
ANDREW HINES (UNIVERSITY OF BRITISH COLUMBIA)
HARRY HSIEH (UC RIVERSIDE)
CHI-YEN HUANG (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)
YU-LUN HUANG (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)

INSEOK HWANG (PURDUE UNIVERSITY)
DANIEL IERCAN (POLITECHNICA U. OF TIMISOARA, UNIVERSITY OF SALZBURG)
ETHAN JACKSON (VANDERBILT)
EFRET JAEGER (SDSC)
STAN JEFFERSON (AGILENT)
RANJIT JHALA (UC SAN DIEGO)
MATTHEW JONES (UC SANTA BARBARA)
WOOYOUNG JUNG (DGIST)
MARCIN JURDZINSKI (UC BERKELEY)
MARCIN JURDZINSKI (UNIVERSITY OF WARWICK)
TOMOYUKI KAGA (TOYOTA MOTOR ENGINEERING & MANUFACTURING)
SHINJIRO KAKITA (SONY CORPORATION)
SRI KANAJAN (GM RESEARCH)
STEVEN KELLY (VANDERBILT)
SOILE VE KERANEN (LBNL)
EUNJOO KIM (DGIST)
DOUG KIMELMAN (IBM)
CHRISTOPH KIRSCH (UNIVERSITY OF SALZBURG)
DAVID W. KNOWLES (LBNL)
HERMANN KOPETZ (TECHNICAL UNIVERSITY OF VIENNA, AUSTRIA)
JAMES KORKOLA (LAWRENCE BERKELEY NATIONAL LABORATORY)
ORNA KUPFERMAN (HEBREW UNIVERSITY)
KLARA NAHRSTEDT (UIUC URBANA-CHAMPAIGN)
ANDREW B. KAHNG (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
DOMINIK LANGEN (INFINEON TECHNOLOGIES)
KIN LARSEN (UNIVERSITY OF AALBORG, AALBORG, DENMARK)
ELIZABETH LATRONICO (BOSCH)
MAN-KIT LEUNG (UC BERKELEY)
ZONG-SYUN LIN (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)
JIE LIU (MICROSOFT RESEARCH)
XIAOJUN LIU (SUN MICROSYSTEMS)
BERTRAM LUDASCHER (UC DAVIS)
JOHN LYGEROS (ETH ZURICH, UNIVERSITY OF PATRAS)
IAN M. MITCHELL (UNIVERSITY OF BRITISH COLUMBIA)
YI MA (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
RUPAK MAJUMDAR (UC LOS ANGELES)
JOSEPH MAKIN (UC BERKELEY)
THOMAS MANDL (BOSCH)
FREDDY MANG (UC BERKELEY)
RADU MARCULESCU (CMU)
RICK MCGEER (HP LABS)
KATHERINE MEZURE (MILLS COLLEGE)

ANDREW MIHAL (UC BERKELEY)
MARK MILLER (HP LABS)
JOHN MOONDANOS (INTEL)
MANFRED MORARI (ETH, ZURICH, SWITZERLAND)
SWAMY MUDDU (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
STEPHEN NEUENDORFFER (XILINX)
GARRY P. NOLAN (STANFORD)
CHRISTOS H. PAPADIMITRIOU (UC BERKELEY)
JAMES L. PAUNICKA (BOEING PHANTOM WORKS)
MARIA PARANDINI (POLITECNICO DI MILANO)
ROBERTO PASSERONE (UNIVERSITY OF TRENTO, ITALY)
GABOR PECELI (TECHNICAL UNIVERSITY OF BUDAPEST, HUNGARY)
ADRIAN PERRIG (CARNEGIE MELLON UNIVERSITY)
CLAUDIO PINELLO (CADENCE BERKELEY LABS, GM)
NIR PITERMAN (EPFL)
GIORDANO POLA (UNIVERSITA' DELL'AQUILA)
G. POLA (UNIVERSITY OF L'AQUILA)
ANUPAM PRAKSH (UC BERKELEY)
MARIA PRANDINI (POLITECNICO DI MILANO)
RODRIC RABBAH (MIT)
JEAN-FRANCOIS RASKIN (UNIVERSITE LIBRE DE BRUXELLES)
SANJAY REKHI (CYPRESS SEMICONDUCTOR)
THOMAS RISGAARD HANSEN (UNIVERSITY OF AARHUS, CS)
TANYA ROOSTA (UC BERKELEY)
MATTI ROSSI (UNIVERSITY OF HELSINKI, ECONOMICS)
KAUSHIK ROY (STANFORD UNIVERSITY)
BERNHARD RUMPE (SOFTWARE ENGINEERING, RWTH AACHEN)
KAREN SACHS (STANFORD)
KAMBIZ SAMADI (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
MOHAN SAROVAR (UC BERKELEY)
S. SASTRY (UNIVERSITY OF WISCONSIN)
NADATHUR SATISH (UC BERKELEY)
MIRKO SAUERMANN (INFINEON TECHNOLOGIES)
EELCO SCHOLTE (UNITED TECHNOLOGIES RESEARCH CENTER)
KOUSHIK SEN (UC BERKELEY)
PUNEET SHARMA (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
JOSEPH SIFAKIS (CNRS VERIMAG, GRENOBLE, FRANCE)
BRUNO SINOPOLI (CARNEGIE MELLON UNIVERSITY)
JIM SMITH (UNIVERSITY OF WISCONSIN)
MYOUNGKYU SOHN (DGIST)
SAMPADA SONALKAR (GENERAL MOTORS, INDIA)
PAUL SPELLMAN (LAWRENCE BERKELEY NATIONAL LABORATORY)

JONATHAN SPRINKLE (UNIVERSITY OF ARIZONA)
MARIELLE STOELINGA (UNIVERSITY OF TWENTE)
PAULO TABUADA (UNIVERSITY OF NOTRE DAME)
CAROLYN TALCOTT (SRI INTERNATIONAL)
ASHISH TIWARI (SRI INTERNATIONAL)
JUHA-PEKKA TOLVANEN (METACASE CONSULTING, VANDERBILT UNIVERSITY)
STAVROS TRIPAKIS (CADENCE BERKELEY LABS)
HSINYI TSAI (NATIONAL CHIAO TUNG UNIVERSITY)
BEN UPCROFT (UNIVERSITY OF SYDNEY)
RANDALL URBANCE (GM RESEARCH)
GUANQIANG WANG (EECS (UC BERKELEY)
LYNN WANG (UC BERKELEY)
YOSINORI WATANABE (CADENCE)
ERIC D.B. WENDEL (SENSIS CORPORATION)
MARK WILCUTTS (TOYOTA MOTOR ENGINEERING & MANUFACTURING NORTH AMERICA)
JOHN WRIGHT (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
LISA WYMORE (UCB DANCE DEPARTMENT)
JOSEPH WYSOCKI (HRL, INDEPENDENT CONSULTANT (MALIBU (CALIFORNIA))
YUHONG XIONG (HP LABORATORIES)
MIN XU (UNIVERSITY OF WISCONSIN)
YEON-MO YANG (DGIST)
GUANG YANG (UC BERKELEY)
HAKAN YAZAREL (TOYOTA TECHNICAL CENTER)
AVIDEH ZAKOR (UC BERKELEY)
FENG ZHAO (MICROSOFT RESEARCH)
LIZHI ZHONG (STMICROELECTRONICS)
QI ZHU (UC BERKELEY)

2 Activities and Findings

2.1 Project Activities

This is the final report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software." hosted at the (Center for Hybrid and Embedded Systems and Software (CHESS), http://chess.eecs.berkeley.edu.
Research at the other CHESS partners: 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 ended before the period covered by this report.
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.

This is a short summary of the overall ITR Project on Foundations of Hybrid and Embedded Systems and Software. Attached to this summary is the compendium of the annual reports for each of the years from 03-10. The format of each of the reports includes the Events each year, the Project Findings in the four key areas, outreach activities and course work development. The number of undergraduate, graduate students and post doctoral scholars educated on this ITR constitutes a tremendous legacy for this program. We will review some of the most recent outcomes in general terms.

2.2 Research

The project made major progress on all sections of the proposed research as detailed below. In addition all of the principals involved in this project worked with national and EU groups to lay out the framework for the research agenda for Cyber Physical Systems, which is a logical successor to Hybrid and Embedded Systems Theory.

2.2.1 Hybrid Systems Theory

Here we made a great deal of progress in the synthesis of controller for hybrid systems, Zeno behavior and the existence of solutions to hybrid dynamical systems, games and stochastic hybrid systems. Additionally, we began a rapproachment between the new results in game theory in theoretical computer science and in control systems design.

It is, however, fair to say that existence to solutions of multi-player games, computation of Nash equilibria for hybrid systems, and the design of high confidence hybrid and embedded systems containing fault tolerant, intrustion tolerant attributes is still in its infancy.

2.2.2 Deep Compositionality

Deep compositionality of hybrid systems was important to establish the interconnections of these systems and platform based design of these systems. This work was encapsulated under

the heading of interface theories for hybrid systems. Model based tools for the simulation and design of systems incorporating both deterministic and stochastic elements is critical especially for network embedded systems.

2.2.3 Robust Hybrid Systems

A large amount of effort was spent in the ITR at approximate and conservative approximations to the design of hybrid systems with uncertainties. This included ways of approximating the synthesis of controllers to hybrid systems and safe set synthesis.

Under the auspices of this ITR, we created the new field of stochastic hybrid systems and came up with some new approaches for synthesizing probabilitic solutions to the synthesis of controllers for hybrid systems.

2.2.4 Hybrid Systems and Systems Biology

The methods of hybrid and embedded systems and software are very relevant to the study of systems biology, with gene-protein and protein-protein networks playing the role of (rather noisy) stochastic hybrid systems. In some exciting new developments towards the end of the project, Tomlin has shown the applicability of the methods of this ITR to the study of the propagation (and eventually therapeutics) for certain kinds of cancers.

2.3 Outreach Activities

Substantive outreach activities included outreach to a number of HBCUs and HSIs for participation in the programs of the ITR, but a significant component of the outreach was to San Jose State University for the development of course work and courses for hybrid and embedded systems. Additionally, each year we sponsored approximately 8 undergraduate students from HBCU/HSI for a summer REU experience.

2.4 Knowledge Transfer

During the course of this ITR, we had very extensive industrial contacts with a number of companies. The majoirty of these relations continue to this day and include: Boeing, Lockheed Martin, Northrup Grumann, United Technologies, Honeywell, GM, Ford, Toyota, Bosch, Siemens, Telecom Italia, Agilent, HP, Microsoft, IBM.

In terms of conferences, we continued to support the conferences and workshops that we helped start: Hybrid Systems: Computation and Control (HSCC), EMSOFT, and most recently CPS Week and the CPS conferences. We worked on US-EU workshops on a diverse set of topics including SCADA, medical devices, air transportation, etc. Most recently, we have been working on a new workshop/conference on High Confidence Embedded Systems Design (offered already once in Stockholm in 2010).

2.5 Curriculum Development for Modern Systems Science (MSS)

Modules were developed to teach hybrid systems to departments other than EECS and CS departments such as Civil, Mechanical, and Chemical Engineering. Additionally, a new undergraduate course (and textbook under preparation by Edward Lee) on Hybrid and Embedded Systems was developed and piloted at Berkeley. A large number of graduate courses were developed and three are offered on a regular basis at Berkeley and Vanderbilt (and also in other universities). A textbook on hybrid control systems is under development by Lygeros, Tomlin and Sastry.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-00225610

UNIVERSITY OF CALIFORNIA AT BERKELEY
VANDERBILT UNIVERSITY
UNIVERSITY OF MEMPHIS

AUGUST 6, 2003

Contents

1. Participants ... 3
1.1. People ... 3
1.2. Partner Organizations ... 5
1.3. Collaborators ... 5
2. Activities and Findings ... 6
2.1. Project Activities ... 6
2.1.1. Hybrid Systems Theory ... 6
2.1.2. Model-Based Design ... 10
2.1.3. Advanced Tool Architectures ... 14
2.1.4. Experimental Research ... 17
2.2. Project Findings ... 21
2.3. Project Training and Development ... 35
2.4. Outreach Activities ... 35
2.4.1. Curriculum Development for Modern Systems Science (MSS) ... 35
2.4.2. Undergraduate Curriculum Insertion and Transfer ... 36
2.4.3. SUPERB-IT Program ... 38
2.4.4. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program ... 41
3. Publications and Products ... 45
3.1. Journal Publications ... 45
3.2. Books and Other Non-Periodical, One-Time Publications ... 48
3.3. Internet Dissemination ... 48
3.4. Other Specific Product ... 48
4. Contributions ... 49
4.1. Within Discipline ... 49
4.1.1. Hybrid Systems Theory ... 49
4.1.2. Model-Based Design ... 50
4.1.3. Advanced Tool Architectures ... 50
4.1.4. Experimental Research ... 51
4.2. Other Disciplines ... 51
4.3. Human Resource Development ... 51
4.4. Research and Education ... 52
4.5. Beyond Science and Engineering ... 52

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, ELECTRICAL AND COMPUTER ENGINEERING)

FACULTY INVESTIGATORS:

ALEX AIKEN, (UC BERKELEY, CS)
RUZENA BAJCSY, (UC BERKELEY, EECS)
GAUTAM BISWAS, (VANDERBILT, COMPUTER SCIENCES)
BELLA BOLLOBAS, (UNIVERSITY OF MEMPHIS, MATHEMATICS)
JEROME A. FELDMAN (UC BERKELEY, EECS)
KENNETH FRAMPTON, (VANDERBILT, MECHANICAL ENGINEERING)
GABOR KARSAI, (VANDERBILT, ELECTRICAL AND COMPUTER ENGINEERING)
KURT KEUTZER, (UC BERKELEY, EECS)
WAGDY H. MAHMOUD (TENNESSEE TECH. UNIVERSITY)
GEORGE NECULA, (UC BERKELEY, EECS)
SRINI RAMASWAMY (TENNESSEE TECH. UNIVERSITY)
PRAVIN VARAIYA, (UC BERKELEY, EECS)

GRADUATE STUDENTS:

LUCA CARLONI (UC BERKELEY)
ABHIJIT DAVARE (UC BERKELEY)
DOUG DENSMORE (UC BERKELEY)
JOYTI GANDHE (VANDERBILT)
MATTHEW HARREN (UC BERKELEY)
FARINAZ KOUSHANFAR (UC BERKELEY)
BENJAMIN R. LIBLIT (UC BERKELEY)
GABOR MADL (VANDERBILT)
DAVID P. MANDELIN (UC BERKELEY)
ELEFTERIOUS MATSIKOUDIS (UC BERKELEY)
WILLIAM PLISHKER (UC BERKELEY)
KAUSHIK RAVINDRAN (UC BERKELEY)
PETER SCHMIDT (VANDERBILT)

TIVADAR SZEMETHY (VANDERBILT)
AMBUJ TEWARI (UC BERKELEY)
GUANG YANG (UC BERKELEY)
JAMES YEH (UC BERKELEY)
YANG ZHAO (UC BERKELEY)

UNDERGRADUATE STUDENTS:

DANIEL BALASUBRAMANIAN
PHILIP BALDWIN (UC BERKELEY)
COLIN COCHRAN (UC BERKELEY)
NICKOLIA COOMBS (VANDERBILT)
RACHAEL DENNISON (VANDERBILT)
DAVID GARCIA (VANDERBILT)
MARCUS GRALY (UC BERKELEY)
SHANTEL HIGGINS (VANDERBILT)
JOHN KILBY (VANDERBILT)
EFOSA OJOMO (VANDERBILT)
MIKE OKYERE (UC BERKELEY)
RAKESH REDDY (UC BERKELEY)
DEVON REED (UC BERKELEY)
MICHAEL RIVERA-JACKSON (VANDERBILT)
ISMAEL SARMIENTO (UC BERKELEY)
BINA SHAH (VANDERBILT)
EDWIN VARGAS (VANDERBILT)
TRIONE VINCENT (VANDERBILT)
ANTONIO YORDAN-NONES (UC BERKELEY)

TECHNICAL STAFF, PROGRAMMERS:

ALLEN HOPKINS (UC BERKELEY)
CHRISTOPHER HYLANDS (UC BERKELEY)
NATHAN JEW (UC BERKELEY)
BRADLEY A. KREBS (UC BERKELEY)
MARVIN MOTLEY (UC BERKELEY)
GUNNAR PROPPE (UC BERKELEY)
MARY STEWART (UC BERKELEY)
NEIL TURNER (UC BERKELEY)
BRIAN WILLIAMS (VANDERBILT)

BUSINESS ADMINISTRATORS:

SUSAN B. GARDNER (UC BERKELEY)

ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)

1.2. Partner Organizations

- University of California at Berkeley
- Vanderbilt University
- University of Memphis

1.3. Collaborators

- Albert Benveniste (IRISA INRIA, Rennes,
- Hermann Kopetz (Technical University of Vienna, Austria)
- Manfred Morari (ETH, Zurich, Switzerland)
- Joseph Sifakis (CNRS VERIMAG, Grenoble, France)
- Kim 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 first Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software". This research activity is primarily organized through a Center at Berkeley CHESS (the Center for Hybrid and Embedded Systems and Software, http://chess.eecs.berkeley.edu), the Vanderbilt ISIS (Institute for Software Integrated Systems, http://www.isis.vanderbilt.edu), and the Department of Mathematical Sciences, (http://msci.memphis.edu) at 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 first year were:

1. The kickoff of the project on November 14th, 2002. The program for the kickoff and the presentations are available at http://chess.eecs.berkeley.edu/conferences/02/program.htm

2. The annual review meeting of the ITR Project on May 8th 2003. The program for the kickoff and presentations are available at http://chess.eecs.berkeley.edu/conferences/03/review03Program.htm

3. The Curriculum Council Kickoff Meeting held in Berkeley on March 1, 2003. The presentation at the kickoff is available on the Chess presentations page http://chess.eecs.berkeley.edu/presentations.htm.

4. A weekly Chess workshop was held at Berkeley. Presentations for the workshop are available at http://chess.eecs.berkeley.edu/workshop.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 are pursuing four directions.

- First, we need to design models of computation that permit the composition of non-functional properties. We have done so for real-time, interrupt-driven, and resource-constrained systems.

- Second, we need to design robust models of computation, where small perturbations of the system description cause only small changes in the system behavior. We have started to design such a model based on discounted games. We are also pursuing robustness and error handling from a programming language perspective.

- Third, we are developing and evaluating algorithms for the efficient analysis of hybrid systems. In particular, we have found improved algorithms for solving stochastic games as well as translations between various models, such as hybrid automata and hybrid bond graphs.

- Fourth, we are studying phase transitions in combinatorial structures. Our main result so far is a directed graph model of the World Wide Web.

2.1.1.a. Deep Compositionality

Real-Time Assumptions and Guarantees

We have studied how systems with real-time assumptions and guarantees can be composed in a deeply compositional, symmetric way, which preserves the timing properties. The results are reported in the paper "The element of surprise in timed games" at CONCUR 2003 [17].

Real-Time Programs with Interrupts

We studied how real-time programs that communicate through interrupts can be composed without exceeding interrupt stack size limitations. The results are reported in the paper "Stack size analysis for interrupt-driven programs" at SAS 2003 [16].

Resource Assumptions and Guarantees

We studied how systems with general resource assumptions and guarantees can be composed under constraints about the available resources. The results are reported in the paper "Resource interfaces" at EMSOFT 2003 [14].

2.1.1.b. Robust Hybrid Systems

A Continuous Theory of Discrete Systems

We have introduced a new paradigm for obtaining a continuous theory of discrete systems that is robust under small perturbations of the system. The results are reported in the paper "Discounting the future in systems theory" at ICALP 2003 [18].

Error Handling and Recovery

We are also analyzing the effectiveness of current programming language mechanisms for specifying error handling and recovery. Specifically, we are studying the exception handling mechanisms present in the Java programming language. We analyze statically Java programs and test whether they detect all run-time exceptions that might be raised by the library functions. We have discovered that in many cases exceptions are not detected by the program itself, meaning that if such an exception arises at run time the execution of the program will be stopped by the runtime system. A more detailed analysis looks at how the program manages resources in the presence of errors. Certain resources, such as files, sockets, and database connections, are limited and should be released properly after the program is done using them. We have observed that even in programs that attempt to do this correctly in the presence of exceptions, there are situations when it is possible that a resource is not released because exceptions are either not detected in time or their handling code fails to release resources. We attribute these errors to the difficulty of writing good error handling code using today's exception handling mechanisms. To address this problem, we are developing a higher-lever mechanism for handling resources, one that should allow the programmer to specify easily that certain resources must be released, even in the presence of errors. This mechanism allows the compiler to insert the necessary error handling code to ensure the correct use of resources.

2.1.1.c. Computational Hybrid Systems

Algorithms for Games on Probabilistic State Spaces

We have developed new, more efficient algorithms for solving games on probabilistic state spaces. These algorithms can be used for the control of stochastic systems. The results are reported in the paper "Simple stochastic parity games" at CSL 2003 [15].

Refining Abstractions

We developed a new algorithm for solving games (e.g. for the control of systems) which is based on abstract interpretation, and automatically refines the abstraction to the necessary degree of precision. The results are reported in the paper "Counterexample-guided control" at ICALP 2003 [23].

Hybrid Bond Graphs

We have also been looking into at systematic methods for modeling complex hybrid systems. When modeling complex physical systems using hybrid models, pure hybrid automata-based formalisms are not well suited for modeling tasks, as they are rarely component-oriented, causality is not explicitly represented, and common engineering concepts, like flow of energy are rather implicit. We have developed a modeling language that is based on the physical phenomena of energy flows between components and addresses these shortcomings. The modeling language is based on the concept of Hybrid Bond Graphs (HBG) [9]. A Bond Graph (BG) model of a physical system is similar to a circuit diagram where ensembles of elementary building blocks correspond to physical components. HBGs introduce discrete behaviors into

BGs, by allowing "junctions" (the interconnection points) to be switched *on* and *off*, based on information carrying signals that are external to the system (*controlled*), or by signals that are internally derived from specific physical variables (*autonomous*). The HBG captures discrete dynamics (via changes in the model topology implemented as switched junctions), and continuous dynamics (specified by the current model topology, i.e., the specific set of elements connected by a particular setting of the switches). Each dissipative, energy storage, transformational, and source element in the graph has a parameter (resistance, capacitance, inductivity, conversion ratio, flow, effort), which corresponds to a physical parameter in the real system (e.g., pipe resistance, tank capacity, flywheel inertia, pump efficiency, power source voltage, etc.). The modeling language for HBGs is visual with textual attributes. The visual notation follows the traditional bond graph symbols wherever feasible, but it extends it with new symbols to account for new concepts (e.g., signals, ports on components, etc.). The extended language has been implemented using the Generic Modeling Environment (GME) framework from Vanderbilt.

We have developed a technique for converting HBG models into equivalent hybrid automata. For every configuration of the switched junctions, which corresponds to a discrete mode of system behavior, one can generate the continuous state space model that captures the continuous dynamics of the system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}); \boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$. Thus, a particular configuration of the switches leads to a one (discrete) mode of the hybrid automata with the flow defined by the state space equations. Conditions that define junction switches are enabling conditions for transitions (and/or as invariants for the modes) and thus formulate a hybrid automaton that is formally equivalent to the HBG. Instead of generating a complete hybrid automaton, we have developed efficient mechanisms for computing the flow models in a new mode, once a mode transition occurs in the system. The technique has been applied to derive hybrid observers for tracking complex system behaviors.

We have developed and tested a technique for deriving an equivalent Matlab Simulink/Stateflow (MSS) model from the HBG. It turns out that HBG model elements can be easily mapped into specific, well-defined MSS blocks that implement the elements of the graph: resistive and energy storage elements, transformational elements, switches, junctions, etc. The connectivity between these blocks represents the effort and flow variables in the system, as well as how the elements set and use these variable values. We have prototyped a systematic procedure that can be used to convert an HBG model into an (executable) MSS, which implements a simulation of the dynamic system, incorporating both continuous and discrete dynamics.

Modeling of Cellular Processes Using Stochastic Hybrid Systems

In this research we study the modeling of biological systems using stochastic hybrid systems. As a generalization of the conventional hybrid systems, stochastic hybrid systems contain both deterministic continuous dynamics and random discrete mode switching, and are finding increasing applications in fields outside control engineering. The model we propose has a discrete state evolving according to a birth-and-death Markov chain. The transition probabilities of this Markov chain are functions of the continuous state. On the other hand, the continuous state has linear deterministic dynamics, whose rates of change are different for different discrete modes. This model is then applied to two biological systems: stochastic gating of ion channels in

cellular membrane, and the phase variation of an E. coli population in environment with limited nutrients. We show that each of the two systems can be modeled in the proposed stochastic hybrid system framework. We then study some analytical properties of the theoretical model, for example, the equilibrium probability distributions of the continuous and discrete states, the expected value of the continuous state, particularly when the number of possible values of the discrete state is small. We also obtain through numerical simulation these probabilistic quantities for the general case. The availability of theoretical analysis and numerical simulating tools makes our model a powerful tool for the modeling of complicated biological systems.

Phase Transitions in Combinatorial Structures

Recently there has been much interest in studying large-scale real-world networks and attempting to model their properties using random graphs. Although the study of real-world networks as graphs goes back several decades, recent activity perhaps started in 1998 with the paper of Watts and Strogatz about the `small-world phenomenon'. Since then the main focus of attention has shifted to the `scale-free' nature of the networks concerned, evidenced by, for example, power-law degree distributions. It was quickly observed that the classical models of random graphs introduced by Erdős and Rényi and Gilbert over forty years ago are not appropriate for studying these networks, so many new models were introduced. The work in this field falls very roughly into the following categories.

1) Direct studies of real-world networks themselves, measuring various properties such as degree-distribution, diameter, clustering etc.

2) Suggestions for new random graph models motivated by this study; first the `small-worlds' model, then many `scale-free' models.

3) Computer simulations of the new models, measuring their properties.

4) Heuristic analysis of the models to predict their properties.

5) Rigorous mathematical study of the new models, to prove theorems about their properties.

Although many hundreds of interesting papers have been written in this area, so far almost all of this work falls into the first four categories; to date there has been very little rigorous mathematical work in the field.

One of our aims in the project is to construct precise mathematical models for various real-life large-scale networks and analyze them rigorously by proving mathematical theorems about them that go way beyond computer experiments and heuristic arguments. In this enterprise we have joined forces with Oliver Riordan of the University of Cambridge, England, and also with the two co-managers of the Theory Group at Microsoft Research, Christian Borgs and Jennifer Chayes. The main result so far is a directed graph model of the World Wide Web.

2.1.2. Model-Based Design

While hybrid systems theory provides a semantic, mathematical foundation for the integrated modeling of physical and information systems, model-based design focuses on the formal representation, composition, and manipulation of models during the design process. It addresses system specification, model transformation, synthesis of implementations, model analysis and

validation, execution, and design evolution. The semantic frameworks in which these models are applied may be domain-specific, offering embedded system designers methods and syntaxes that are closer to their application domain. To do this well, they must emphasize concurrency, communication abstractions, and temporal properties, rather than procedural interfaces. For example, domain-specific semantic frameworks 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..

2.1.2.a. Composition of Domain Specific Modeling Languages

Domain Specific Modeling Languages

In model-based design, systems are described by models expressed in domain specific modeling languages (DSML). The short summary of our approach is the following:

1. DSML is a five-tuple of concrete syntax (*C)*, abstract syntax *(A)*, semantic domain *(S)* and semantic and syntactic mappings (M_S, and M_C):
$$L = < C, A, S, M_S, M_C >$$
The *C concrete syntax* defines the specific (textual or graphical) notation used to express models, which may be graphical, textual or mixed. The *A abstract syntax* defines the *concepts, relationships,* and *integrity constraints* available in the language. Thus, the abstract syntax determines all the (syntactically) correct "sentences" (in our case: models) that can be built. (It is important to note that the abstract syntax includes semantic elements as well. The integrity constraints, which define well-formedness rules for the models, are frequently called "static semantics".) The *S semantic domain* is usually defined by means of some mathematical formalism in terms of which the meaning of the models is explained. The $M_C : A \rightarrow C$ mapping assigns syntactic constructs (graphical, textual or both) to the elements of the abstract syntax. The M_S: $A \rightarrow S$ semantic mapping relates syntactic concepts to those of the semantic domain.
2. The languages, which are used for defining components of DSMLs are called *meta-languages* and the concrete, 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. In our work in Model-Integrated Computing (MIC), we adopted UML class diagrams and the Object Constraint Language (OCL) as meta-language. This selection is 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. The semantic domain and semantic mapping defines semantics for a DSML. The role of semantics is to give a precise interpretation for the meaning of models that we can create using the modeling language. Naturally, models might have different interesting properties; therefore a DSML might have a multitude of semantic domains and semantic mappings associated with it. For example, *structural* and *behavioral* semantics are frequently associated with DSMLs. 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 (structural semantics is frequently called *instance semantics*).

Accordingly, the semantic domain for structural semantics is defined by some form of set-relational mathematics. The *behavioral semantics* describes the evolution of the state of the modeled artifact along some time model.

We have continued our work on developing DSML-s and using meta-modeling for the compositional specification of complex, multiple view modeling languages.

Hybrid System Simulation

In collaboration with the DARPA-sponsored Mobies program, we have developed a hybrid system visual modeling tool based on Ptolemy II and posted the first version, designated HyVisual 2.2-beta, on the CHESS website (http://chess.eecs.berkeley.edu) [25]. HyVisual enables the graphical construction of hybrid systems that combine continuous-time dynamics (given as ODEs) with finite-state automata that represent modes of operation of continuous-time components. It illustrates a more hierarchical approach to hybrid system modeling that is prevalent in commercially available tools (such a Simulink) and in most hybrid systems formalisms. HyVisual has been used at Berkeley in a graduate course on hybrid systems. We are leveraging the system to explore the operational semantics of hybrid systems, with the objective of reconciling simulation and denotational semantics. HyVisual includes a facility to translate Hybrid System Interchange Format (HSIF) files into MoML, the XML format used to represent Ptolemy II Models.

Programming Model for Network Processors

We are also working on a programming model for network processors. Network processors, like many embedded systems, have employed a large variety of hardware techniques to accelerate applications in their application domain. These techniques include parallel processing, special-purpose hardware, memory architectures, on-chip communication mechanisms, and the use of peripherals. However, despite this architectural innovation, relatively little effort has been made to make these architectures easily programmable. The current practice of programming network processors is to use assembly language or a subset of C. This low-level programming language places a large burden on the programmer to understand fine details of the architecture just to implement a packet processing application, let alone optimize it. We believe the programmer should be presented with an abstraction of the underlying hardware, or a programming model, which exposes just enough detail to write efficient code for that platform.

Our approach starts with an abstraction based on Click, a domain-specific language for packet processing, and augments it with features of the architecture that allow for successive performance improvement. Our initial work indicated that thread boundaries, the layout of shared data, and the arbitration of shared resources were those features which were most important to creating an efficient implementation. To demonstrate the promise of our programming model, we implemented a representative mix of networking applications on the Intel IXP1200, a common network processor. To date, these applications include IPv4 forwarding, network address translation, and diffserv, each of which had a design time of only a few days: a fraction of the time needed for a hand-coded implementation. For IPv4 forwarding, we found the performance of this approach falls within 10% of a hand-coded design. Future

directions of this work include finding optimal strategies and heuristics to automate traversing this potentially enormous software design space and also finding optimizations based on the high-level description that are not easily found on the generated assembler or C.

2.1.2.b. Extensions to Distributed Models of Embedded systems

Extensions of HSIF for Faulty Behavior

We have been working on developing a modeling language for describing both normal and faulty distributed embedded systems by extending the Hybrid Systems Interchange Format (HSIF). The steer-by-wire system of automobiles, for example, can be described by two interacting subsystems, one for the hand-wheel system and one for the road-wheel system. Each subsystem captures the continuous dynamics of the motors and the wheels as well as the discrete dynamics of the embedded controller. In HSIF, a network of hybrid automata is a tuple *(HA, V, P, C)* consisting of a set of hybrid automata *HA*, a set of variables *V* that include local and shared variables as well as input and output signals that defined the interactions between automata, a set of parameters *P*, and a constraint *C* over the input variables of the network. The network semantics is defined by synchronous continuous steps in time by all automata, and discrete steps that correspond to valid sequences of discrete steps for each of the automata. Automata communicate with each other by synchronous signals and shared variables. This hybrid system model is general enough to capture many classes of dynamic systems including: discrete event systems (no continuous evolution within locations), timed discrete event systems (with only continuously increasing time variables), and linear and nonlinear continuous systems (just one location without transitions except the default self-loop transition to synchronize with other components).

To capture dynamic behavior of the interactions between the network of automata for tracking nominal and faulty behavior of systems, we have been looking at methodologies for systematically defining the coupling between subsystems under nominal and faulty conditions. We have considered two classes of interaction between subsystems: ***physical*** and ***logical***. Such interactions have been modeled in the HSIF framework using input signals, output signals, and shared variables. The semantics of the interactions are being exploited to define the propagation of dynamic effects of faults, and define fault signatures that are exploited for fault isolation tasks.

2.1.2.c. Model Transformation

Graph Transformations

In the past period of the project, the research focused on the foundations and basic algorithms for model transformations. We have developed an approach based on graph transformations that allows a very high-level specification of transformational programs. The approach shows a number of novel features: (1) it is based on UML: an industry standard, (2) the approach uses a subset UML with well-defined defined semantics, (3) the transformation specifications are expressed in terms of UML entities (classes and associations), (4) the semantics of the transformation rules has been formally defined (using Z), and (5) for efficiency, explicit sequencing and control flow of transformations is supported. We have defined a prototype

implementation of a Graph Rewriting Engine (GRE) that implements the core transformation semantics and acts as a "virtual machine" for graph transformations. We have also defined two high-level visual languages for representing the model transformations. One language is suitable for the specification of arbitrary model transformations, while the other language has been tailored for solving model migration problems that arise when the (metamodel of the) modeling language changes. We have developed several prototype model transformers that illustrate how the language and the engine can be used in realistic problems of embedded system development. We have also investigated two additional aspects of model transformations: debugging and code generation. We have done research on how to support the development of graph transformation programs using a debugger tool, and created research prototype for it. We have also studied how directly executable code can be generated from the specification of transformations. We have developed algorithms and implemented them to facilitate this code generation, and the initial results are very encouraging.

Verifying Functional Behavior and its Mapping to Architectural Resources

In collaboration with the Gigascale System Research Center and industry, we have further developed Metropolis. Metropolis provides an infrastructure based on a model with precise semantics that remain general enough to support existing computation models and accommodate new ones. This *metamodel* can support not only functionality capture and analysis, but also architecture description and the mapping of functionality to architectural elements (this is the differentiating factor from other system-level design tools). Metropolis uses a logic language to capture nonfunctional and declarative constraints. Because the model has a precise semantics, it can support several synthesis and formal analysis tools in addition to simulation. The first design activity Metropolis supports, communication of design intent and results, focuses on the interactions between people working at different abstraction levels and between people working concurrently at the same abstraction level. The metamodel includes constraints that represent in abstract form requirements not yet implemented or assumed to be satisfied by the rest of the system and its environment. Under Chess support, G. Yang has worked with the Metropolis MetaModel Simulator. The simulator is the native tool for verification. It was able to concurrently verify functional behavior and its mapping to architectural resources. Checking of simulation results for formal constraint satisfaction, constraints resolution in concurrent simulation, and code generation for debugging at the metamodel level were performed previously by other tools in the environment. In the funding period, the simulator has been updated so that all the verification tasks are unified under its umbrella. Metropolis has been used in the experimental research part of the proposal to select appropriate implementation architectures including distributed systems.

2.1.3. Advanced Tool Architectures

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. Concretely, the key products of this work will be a set of toolkits, frameworks, and other software modules. We still develop tools, but only as reference applications of the toolkits and frameworks.

2.1.3.a. Syntax and Semantics

Hybrid System Simulation

We are investigating the operational semantics of hybrid systems simulators using the HyVisual system, built using Ptolemy II, as an experimental laboratory. Particular issues that are being investigated include the fixed-point semantics of zero-delay operations. We have identified interesting relationships between the semantics of networks of hybrid systems and that of synchronous languages and discrete-event languages. We are working on a unifying theory and common software infrastructure that can be used across these models of computation.

Causality

Hierarchical hybrid systems specifications, such as those in HyVisual, require propagation of certain interface properties across levels of the hierarchy. In particular, when a suite of continuous-time models are hidden hierarchically within a modal model, it becomes necessary to merge certain properties of each of the continuous-time models to infer key properties of the modal model. A key example is causality. For execution of hybrid systems models, it is necessary to determine for each component in the system whether its input-output relationships are strictly causal. If every directed loop has at least one strictly causal component, then the solution to be calculated is assured to be unique. We are working on mechanisms for merging such interface properties and propagating them up the hierarchy.

Choosing Synchronization Policies

In a project on heterogeneous reactive systems modeling and correct-by-construction deployment, we are developing a mathematical framework for the heterogeneous modeling of reactive and real-time systems to allow freedom of choice between different synchronization policies at different stages of the design process. The focus of our framework is on handling communication and coordination among heterogeneous processes in a mathematically sound way, and its interest rests on the theorems it can provide. We have derived a set of theorems that support effective techniques to generate automatically correct-by-construction adaptors between designs formulated using different coordination paradigms.

We have applied these concepts to two scenarios that are of particular relevance for the design of embedded systems: the deployment of a synchronous design over a globally-asynchronous locally-synchronous (GALS) architecture and over a loosely time-triggered architecture (LTTA).

The idea followed in these applications is to abstract away from the synchronous specifications the constraints among events of different signals due to the synchronous paradigm and, then, to map the unconstrained design into a different architecture characterized by a novel set of

requirements among events. In doing so, we must make sure that, when we remap the design, the intended behavior of the system is retained. This is achieved by relying on a formal notion of semantics-preserving transformations that we developed based on the idea of morphisms over tag sets.

We wrote a paper on this work in collaboration with A. Benveniste (INRIA) and P. Caspi (VERIMAG). The paper has been accepted for publication and will be presented at the Third International Conference on Embedded Software (EMSOFT) in October 2003 [7].

2.1.3.b. Interface Theories

Checking Interface Compatibility

We have developed algorithms and built a tool, called Chic (Checking Interface Compatibility), for checking if a set of interfaces that specify synchronous, asynchronous, pushdown (recursive), and resource (time, space, power) constraints fit together, and for deriving the composite interface. We have posted on the CHESS website a first version of Chic, a modular verifier for behavioral compatibility of software and hardware component interfaces. Chic is a modular verifier for behavioral compatibility of software and hardware components. The goal of Chic is to be able to check that the interfaces for software or hardware components provide guarantees that satisfy the assumptions they make about each other. Chic supports a variety of interface property specification formalisms.

We have created a preliminary framework for integrating Chic with Ptolemy II. In this framework, Chic is represented by an attribute that is attached to a design. Interfaces for components are attached to the individual components, and an interface theory is specified by configuring the Chic attribute. A preliminary user interface has been constructed that enables checking interface compatibility. We are using this framework to investigate the use of various interface theories with practical designs.

Debugging Temporal Specifications

We have also worked on the problem of debugging temporal specifications with concept analysis. Here we have developed a novel method for debugging formal temporal specifications. A straightforward way to debug a specification is based on manually examining the short program execution traces that program verification tools generate from specification violations and that specification miners extract from programs. This method is tedious and error-prone because there may be hundreds or thousands of traces to inspect. Our method uses concept analysis to automatically group traces into highly similar clusters. By examining clusters instead of individual traces, a person can debug a specification with less work.

To test our method, we implemented a tool, Cable, for debugging specifications. We have used Cable to debug specifications produced by Strauss, our specification miner. We found that using Cable to debug these specifications requires, on average, less than one third as many user

decisions as debugging by examining all traces requires. In one case, using Cable required only 28 decisions, while debugging by examining all traces required 224.

2.1.3.c. Virtual Machine Architectures

A Virtual Machine for Scheduling

We have developed and implemented a virtual machine for scheduling, which can be used to replace the real-time operating system (or its scheduler) in Giotto implementations. The results are reported in the paper "Schedule-carrying code" at EMSOFT 2003[24] and the S(cheduling) machine implementation is available on the CHESS website.

2.1.3.d. Components for Embedded Systems

Component Libraries

We are building a component library for Ptolemy II that includes capabilities that will support the prototyping of interesting and compelling hybrid and embedded systems applications. This component library includes interfacing to various hardware devices, including X-10, robotic arms, sensors, and actuators, plus components for secure networking, communication, signal processing, and graphics.

Mappings Between Functionality and Architecture

We have been exploring the use of Metropolis for experimenting with mappings between functionality and architecture. In particular, we have been working this summer on the Picture-in-Picture design driver. Currently, we are working on the mapping portion of the design, trying different mappings between functionality and architecture and evaluating their performance. During this process, we are also identifying modifications that can be made to the functional and architectural APIs to facilitate mapping.

In addition, we have worked on platform-based reconfigurable hardware exploration. This project looks at taking a high level description of an application's requirements and transforming it via a series of constraints into an abstraction of the possible configurations of a reconfigurable hardware device. Taking this abstraction (a platform), it then estimates what the performance of various instances of this platform would be on the device (Cypress Semiconductor's PSOC). This methodology is both top down and bottom up in its use of constraints and performance estimation. It frames the construction of platform instances as Boolean constraint formulations and solves them using the principles of Boolean Satisfiability.

2.1.4. Experimental Research

Experimental research is intended to carry forward the application of the principles developed in the other activities of the research program and to validate them. In addition, our approach to experimental research is to extract the fundamental aspects of design to incorporate them in the theory part of our work. Our attention has been focused on important classes of applications that

are relevant to the US public and private sectors. This aspect is a qualifying part of our research program.

2.1.4.a. Embedded Control Systems

Safety Control

We have developed an online approach to the safety control of a general class of hybrid systems [9]. The proposed approach does not require the existence of a finite quotient structure. Moreover, the approach can be adapted to accommodate possible changes in the modeling parameters that may occur, as a result of faults or the time-varying nature of the system. In the current stage, we consider a special model for hybrid systems that can capture the switching dynamics associated with many practical real-life systems. Such class of systems is referred to as switching hybrid systems. The main distinguishing characteristic of switching hybrid system is that the set of inputs is assumed finite. The proposed 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 typical in many practical computer-controlled systems, where the input is usually discrete and restricted to a finite set.

The problem of safety control is stated as follows. *Given a system with a state space X, a set of safe states X_s and a set of initial states $X_o \subseteq X$ where $X_s \subseteq X_o$, design a supervisor that can drive the system from any state in X_o to X_s in finite number of time steps using a finite set of switching events.* In addition, the supervisor is required to keep the system stable within the set X_s. In this setting, the supervisor is simply considered an agent that applies a given sequence of events (possibly changing the discrete input) in order to achieve a certain objective.

For the safety control problem, the selection of the next step is based on a distance map that defines how close the current state is to the safe region. The online supervision algorithm starts by constructing the tree of all possible future states from the current state up to a specified depth (i.e., a finite time horizon). To avoid Zeno effects, in which the controller may try to preempt time indefinitely through continuous switching, we require that a switching event must be followed by at least one time event. The exploration procedure identifies the set of states with the minimal distance from the safe set based on the given distance map. A state x_m is then chosen from this set based on certain optimality criterion (for instance minimal time from the current state), or simply picked at random. The chosen state is then traced back to the current state and the (input) event leading to x_m is used for the next step.

Worst-Case Execution Time Bounds

We have also recently begun looking at the problem of computing worst-case execution time bounds for embedded software. Many embedded systems rely on hard real-time constraints, and we would like to determine statically that these constraints are met. To reduce the overestimation that is common in worst-case execution time bounds, one idea we are exploring is to use optimistic assumptions about cache behavior, and provide mechanisms for the real-time system to recover gracefully if our not-quite-worst-case bound is violated. This research area is

the complement to projects like Giotto, which schedule embedded tasks given the worst-case execution time bounds for these tasks.

Novel Microarchitectures

Using platform-based design as the principle according to which embedded controllers are developed, we have the opportunity of developing novel micro-architectures that support the control code better in terms of computing time and power consumption while minimizing cost. In the application to automotive safety-critical systems we have been able to design novel micro-architectures that are now being manufactured as chip sets. The development of Metropolis is allowing us to analyze and optimize micro-architectures for embedded controllers an order of magnitude better. In particular, we have studied how to develop novel micro-architectures with the Metropolis successive refinement support. Micro-architecture development is becoming more challenging due largely to increased application complexity and the introduction of many heterogeneous hardware components. This requires that system-level-design modeling methodologies incorporate increasingly abstract models in order to simulate the behavior of the design. As the design progresses, these abstract models will be replaced by refined models. The refined models will more accurately reflect the proposed implementation. However, in order to ensure correctness and save on design time, a methodology should be in place to ensure that these refined models exhibit behavior within that of the abstract model. This is the process of refinement verification. The combination of abstraction level exploration and refinement verification can be termed successive platform refinement. We introduce the Metropolis design environment as a means to provide high level, abstract modeling performed on a complex data communication application case study. This modeling and development process proceeds in a platform based design methodology and demonstrates the usefulness of successive platform refinement.

In a more general framework, we are looking at possible ways to model architecture platforms and their refinements. Issues being investigated entail how quantities such as time are measured, how the partitioning of architecture into scheduling and scheduled components should be done, and how tasks make use of services and what services should be provided. This is being examined in a generic context but also with a "picture-in-picture" application in mind.

2.1.4.b. Embedded Software for National and Homeland Security

Software for UAVs

We have been working on embedded software for unmanned aerial vehicles UAVs for national and homeland security. Coordination of multiple unmanned aerial vehicles (UAVs) poses significant theoretical and technical challenges. Recent advances in sensing, communication, and computation enable the conduct of cooperative multiple-UAV missions deemed impossible in the recent past. Through the course of a single mission a group of UAVs organized in close formation may need to reconfigure between formations as different formations are suited to different tasks such as obstacle avoidance and pursuit and evade tactics. Given an initial configuration, final configuration, time with witch to complete the reconfiguration and intra and inter vehicle constraints such as vehicle dynamics and separation constraints, a centralized

optimization algorithm may be used to calculate optimal nominal trajectories for each vehicle to reconfigure from the initial to the final configuration in the allotted time while satisfying the constraints. These trajectories may then be parameterized and stored on line in a hybrid structure of formation maneuvers in which a transition from one formation to another is governed by a finite automaton. The nodes of the resulting hybrid graph consist of formation keeping modes for every desired formation and reconfiguration modes which specify the optimal trajectory solutions between each pair of formations. Graph edges exist between alternating formation keeping and reconfigurations modes. Thus, any reachable formation may be effectively searched through the graph and achieved by performing a sequence of formation reconfigurations.

Soft Walls

We have continued to develop Soft Walls as a Chess application. In brief, modern aircraft all have electronics on board that is involved with the control and navigation of the aircraft. Many of the newer planes have computers on board that mediate the commands issued by the pilot and translate those commands into action, for example to bank and turn to the right. It is possible to modify the software in the computers in such a way that an airplane will refuse to enter pre-specified regions. We call these regions "no-fly zones" and we call the boundaries of these regions "Soft Walls." If an aircraft is equipped with the Soft Walls system, then if the pilot attempts to enter a no-fly zone, the airplane will be diverted. This happens gently at first, but if the pilot does not cooperate, then the system becomes more assertive. The key principle is to give the pilot as much control over the aircraft as is consistent with the constraint that the airplane does not enter the no-fly zone.

In collaboration with a NASA-sponsored project, we have developed a control algorithm that blends pilot commands with a bias that is introduced when an aircraft approaches a no-fly zone. The algorithm keeps the aircraft out of a backward reachable set from the no-fly zone, ensuring that at all times there no sequence of pilot commands that will take the aircraft into the no-fly zone.

2.1.4.c. Networks of Distributed Sensors

Low Energy Coordination in Wireless Ad-Hoc Sensor Networks

We have been working on low energy coordination in wireless ad-hoc sensor networks. Energy consumption is one of the main constraints in wireless ad-hoc networks. In modern wireless technologies, the energy budget is dominated by the communication cost. In the state-of-the-art wireless communication systems, transmission, reception, and listening have similar power requirements that are at least order of magnitude higher than the power consumption in sleep state. Therefore, the most effective way for prolonging the lifetime of the network is to place a majority of the nodes in the sleep state.

We have devised a low power coordination scheme that leverages on the redundancies between the nodes to power off the redundant nodes to save the energy consumption. The technique works by measuring and calculating the relevant properties of the network in such a way that the

targeted functionality is globally preserved. The procedures use new theoretical results, analytic and statistical analysis, modeling of several energy-consumption related factors, and a set of protocols and optimal localized algorithms for selecting the power states of the nodes in the network. We have so far simulated and verified the efficiency of the technique on three different tasks: connectivity, multicasting and exposure.

2.2. Project Findings

ABSTRACTS FOR KEY PUBLICATIONS, WHICH REPRESENT PROJECT FINDINGS, ARE PROVIDED HERE. THESE ARE LISTED ALPHABETICALLY BY FIRST AUTHOR.

[1] Online Safety Control of a Class of Hybrid Systems

Abdelwahed, S., G. Karsai, and G. Biswas,
Proc. 41st IEEE Conference on Decision and Control, Las Vegas, NV, 2002.

Abstract: In this paper we outline a supervisor synthesis procedure for safety control of a class of hybrid systems. The procedure is conducted online based on a limited exploration of the state space. We establish feasibility conditions for online controllability with respect to the safety specifications, and provide an upper limit for the accuracy error of the online controller.

[2] Interpreter Writing using Graph Transformations

Agrawal A., Karsai G., Shi F., Technical Report ISIS-03-401, 2003.

Abstract: This paper introduces a UML-based approach for specifying model transformations. The technique is based on graph transformations, where UML class diagrams are used to represent the graph grammars of the input and the output of the transformations, and the transformations are represented as explicitly sequenced elementary rewriting operations. The paper discusses the visual language designed for the representation of transformation programs and the graph transformation execution engine which implements the semantics of the language.

[3] An End-to-End Domain-Driven Development Framework

Agrawal A., Karsai G., Ledeczi A.
8th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, (under revision), Anaheim, California, October 26, 2003.

Abstract: This paper presents a comprehensive, domain-driven framework for software development. It consists of a meta-programmable domain-specific modeling environment and a model transformation generator toolset based on graph transformations. The framework allows the creation of custom, domain-oriented programming environments that support end-user programmability. In addition, the framework could be considered an early, end-to-end implementation of the concepts advocated by the OMG's Model Driven Architecture initiative.

[4] Generative Programming via Graph Transformations in the Model-Driven Architecture

Agrawal A., Levendovszky T., Sprinkle J., Shi F., Karsai G., OOPSLA, Workshop on Generative Techniques in the Context of Model Driven Architecture, Seattle, WA, November 5, 2002.

Abstract: The Model-Driven Architecture of OMG envisions a development paradigm where designers create a Platform-Independent Model (PIM) of the design, which is then refined into a Platform-Specific Model (PSM). This paper argues that this approach lends itself well to generative programming techniques, and that tools are needed to support this transformation. The paper shows how a technique based on graph transformations could be applied to automate the process, as well as make it user-extendible

[5] Continuous Percolation

Balister, Paul, Béla Bollobás, Mark Walters
Submitted to Annals of Applied Probability, 2003

Abstract: We prove some bounds on the critical probability for continuous percolation in the plane. The proof is in two parts. The first is a rigorous reduction of the problem to a finite problem. We then solve this finite problem using Monte-Carlo methods.

[6] Sharp thresholds in Bootstrap Percolation

Balogh ,Jozsef, and Béla Bollobás
To appear in Physica, 2003.

Abstract: In the standard bootstrap percolation on the *d*-dimensional grid G_n^d, in the initial position each of the n^d sites is *occupied* with probability *p* and *empty* with probability 1 - *p*, independently of the state of every other site. Once a site is occupied, it remains occupied for ever, while an empty site becomes occupied if at least two of its neighbours are occupied. If at the end of the process every site is occupied, we say that the (initial) configuration *percolates*. By makinguse of a theorem of Friedgut and Kalai (Proc. Amer. Math. Soc. 124 (1996) 2993), we shall show that the threshold function of the percolation is sharp. We shall prove similar results for three other models of bootstrap percolation as well.

[7] Heterogeneous Reactive Systems Modeling and Correct-by-Construction Deployment

Benveniste, Albert, Luca P. Carloni, Paul Caspi, and Alberto L. Sangiovanni-Vincentelli
To appear, EMSOFT 2003

Abstract: We propose a mathematical framework to deal with the composition of heterogeneous reactive systems. Our theory allows to establish theorems, from which design techniques can be derived. We illustrate this by two cases: the deployment of synchronous designs over GALS architectures, and the deployment of synchronous designs over the so-called Loosely Time-Triggered Architectures.

[8] Degree Distribution of the FKP Network Model

Berger, Noam, Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan
Proc. ICALP 2003, Lecture Notes in Computer Science, LNCS 2719, Springer-Verlag, 2003.

Abstract: Recently, Fabrikant, Koutsoupias and Papadimitriou in their paper "Heuristically optimized trade-offs: a new paradigm for power laws in the internet" introduced a natural and beautifully simple model of network growth involving a trade-off between geometric and network objectives, with relative strength characterized by a single parameter which scales as a power of the number of nodes. In addition to giving experimental results, they proved a power-law lower bound on *part* of the degree sequence, for a wide range of scalings of the parameter. Here we prove that, despite the FKP results, the overall degree distribution is very far from satisfying a power law.

First, we establish that for almost all scalings of the parameter, either all but a vanishingly small fraction of the nodes have degree 1, or there is exponential decay of node degrees. In the former case, a power law can hold for only a vanishingly small fraction of the nodes. Furthermore, we show that in this case there is a large number of nodes with almost maximum degree. So a power law fails to hold even approximately at either end of the degree sequence range. Thus the power laws found in "Heuristically optimized trade-offs: a new paradigm for power laws in the internet" are very different from those given by other internet models or found experimentally Faloutsos, Faloutsos and Faloutsos, "On power-law relationships of the internet topology".

[9] Self Adaptive Software for Fault Adaptive Control

Biswas, G., G. Simon, G. Karsai, S. Abdelwahed, N. Mahadevan, T. Szemethy, J. Ramirez, G. Péceli and T. Kovácsházy
Proc. International Workshop on Self Adaptive Software, Washington D.C., June 2003.

Abstract: Self-adaptive software is a technology that allows building fault-adaptive control systems: control software that can survive faults in the system under control, and in the control software itself. This form of self-adaptive software requires capabilities for the detection and isolation of faults when the system is in operation, and then taking appropriate control actions to mitigate the fault effects and maintain system operation in the best way possible. This paper discusses heterogeneous model-based approach for building fault-adaptive control software, with special emphasis on the modeling schemes that describe different aspects of the system functionality and behavior at different levels of granularity. The computational architecture is applied to design and run experiments on a fault-adaptive control software of an airplane fuel system.

[10] Directed Scale-free Graphs

Bollobás, Béla, Christian Borgs, Jennifer Chayes, and Oliver Riordan
To appear in Proc. 14th ACM-SIAM Symposium on Discrete Algorithms, 2003.

Abstract: We introduce a model for directed scale-free graphs that grow with preferential attachment depending in a natural way on the in- and out-degrees. We show that the resulting in- and out-degree distributions are power laws with different exponents, reproducing observed properties of the world-wide web. We also derive

exponents for the distribution of in- (out-) degrees among vertices with fixed out- (in-) degree. We conclude by suggesting a corresponding model with hidden variables.

[11] Multicolored Extremal Problems

Bollobás, Béla, Peter Keevash and Benny Sudakov.
Submitted

Abstract: Many problems in extremal set theory can be formulated as finding the largest set system (or r -uniform set system) on a fixed ground set X that doesn't contain some forbidden configuration of sets. We will consider multicoloured versions of such problems, defined as follows. Given a list of set systems, which we think of as colours, we call another set system multicoloured if for each of its sets we can choose one of the colours it belongs to in such a way that each set gets a different colour. Given an integer k and some forbidden configurations, the multicoloured extremal problem is to choose k colours with total size as large as possible subject to containing no multicoloured forbidden configuration.

Let f be the number of sets in the largest forbidden configuration. For $k \leq f - 1$ we can take all colours to consist of all subsets of X (or all r -subsets in the uniform case), and this is trivially the best possible construction. Even for $k \geq f$ - 1, one possible construction is to take $f - 1$ colours to consist of all subsets, and the other colours empty. Another construction is to take all k colours to be equal to a fixed family that is as large as possible subject to not containing a forbidden configuration. We will consider a variety of problems in extremal set theory, for which we show that one of these two constructions is always optimal. This was shown for the multicoloured version of Sperner's theorem by Daykin, Frankl, Greene and Hilton. We will extend their result to some other Sperner problems, and also prove multicoloured versions of the generalised Erdős-Ko-Rado theorem and the Sauer-Shelah theorem.

[12] Coupling Scale-free and Classical Random Graphs

Bollobás, Béla, and Oliver Riordan
To appear in Internet Mathematics, 2003.

Abstract: Recently many new `scale-free' random graph models have been introduced, motivated by the power-law degree sequences observed in many large-scale real-world networks. The most studied of these is the Barabási-Albert model, made precise as the LCD model by the present authors.

Here we use coupling techniques to show that in certain ways the LCD model is not too far from a standard random graph; in particular, the fractions of vertices that must be retained under an optimal attack in order to keep a giant component are within a constant factor for the scale-free and classical models.

[13] Max Cut for random graphs with a planted partition

Bollobás Béla., and A.D. Scott
Submitted to Combinatorics, Probability and Computing, 2003.

Abstract: We give an algorithm that, with high probability, recovers a planted *k*-partition in a random graph, where edges within vertex classes occur with probability *p* and edges between vertex classes occur with probability $r \geq p + c(k)\sqrt{p \log n / n}$. The algorithm can handle vertex classes of different sizes and, for fixed *k*, runs in linear time. We also give variants of the algorithm for partitioning matrices and hypergraphs.

[14] Resource Interfaces

Chakrabarti, Arindam, Luca de Alfaro, Thomas A. Henzinger, and Marielle Stoelinga
To appear, EMSOFT 2003.

Abstract: We present a formalism for specifying component interfaces that expose component requirements on limited resources. The formalism permits an algorithmic check if two or more components, when put together, exceed the available resources. Moreover, the formalism can be used to compute the quantity of resources necessary for satisfying the requirements of a collection of components.

The formalism can be instantiated in several ways. For example, several components may draw power from the same source. Then, the formalism supports compatibility checks such as: can two components, when put together, achieve their tasks without ever exceeding the available amount of peak power? or, can they achieve their tasks by using no more than the available amount of energy (i.e., power accumulated over time)? The corresponding quantitative questions that our algorithms answer are the following: what is the amount of peak power necessary for two components to be put together? what is the corresponding amount of energy? To solve these questions, we model interfaces with resource requirements as games with quantitative objectives, where each state is labeled by a number representing, for example, power consumption. We present solutions for several finite and infinite games not found in the literature. We illustrate the methodology by modeling compatibility questions for networks of embedded motes, and for software modules controlling Lego robots.

[15] Simple Stochastic Parity Games

Chatterjee, Krishnendu, Marcin Jurdzinski, and Thomas A. Henzinger
In Proceedings of the International Conference for Computer Science Logic (CSL), 2003.

Abstract: Many verification, planning, and control problems can be modeled as games played on state-transition graphs by one or two players whose conflicting goals are to form a path in the graph which satisfies a given objective. The focus here is on simple stochastic parity games, that is, two-player games with turn-based probabilistic transitions and omega-regular objectives formalized as parity (Rabin chain) winning conditions. An efficient translation from simple stochastic parity games to nonstochastic parity games is given. As many algorithms are known for solving the latter, the translation yields efficient algorithms for computing the states of a simple stochastic parity game from which a player can win with probability 1.

An important special case of simple stochastic parity games are the Markov decision processes with Buchi objectives. For this special case a first provably subquadratic algorithm is given for computing the states from which the single player has a strategy to achieve a Buchi objective with probability 1. For game graphs with *m* edges the algorithm works in time *O(m^1.5)*. Interestingly, a similar technique sheds light on the question of the computational complexity of solving simple Buchi games and yields the first provably subquadratic algorithm, with a running time of *O(n^2/log n)* for game graphs with *n* vertices and *O(n)* edges.

[16] Stack Size Analysis for Interrupt-driven Programs

Chatterjee, Krishnendu, Di Ma, Rupak Majumdar, Tian Zhao, Thomas A. Henzinger, and Jens Palsberg
Proceedings of the Tenth International Static Analysis Symposium (SAS), 2003.

Abstract: We study the problem of determining stack boundedness and the exact maximum stack size for three classes of interrupt-driven programs. Interrupt-driven programs are used in many real-time applications that require responsive interrupt handling. In order to ensure responsiveness, programmers often enable interrupt processing in the body of lower-priority interrupt handlers. In such programs a programming error can allow interrupt handlers to be interrupted in cyclic fashion to lead to an unbounded stack, causing the system to crash. For a restricted class of interrupt-driven programs, we show that there is a polynomial-time procedure to check stack boundedness, while determining the exact maximum stack size is PSPACE-complete. For a larger class of programs, the two problems are both PSPACE-complete, and for the largest class of programs we consider, the two problems are PSPACE-hard and can be solved in exponential time.

[17] The Element of Surprise in Timed Games

de Alfaro, Luca, Marco Faella, Thomas A. Henzinger*, Rupak Majumdar, and Marielle Stoelinga*
Proceedings of the 14th International Conference on Concurrency Theory (CONCUR), 2003.

Abstract: We consider concurrent two-person games played in real time, in which the players decide both which action to play, and when to play it. Such timed games differ from untimed games in two essential ways. First, players can take each other by surprise, because actions are played with delays that cannot be anticipated by the opponent. Second, a player should not be able to win the game by preventing time from diverging. We present a model of timed games that preserves the element of surprise and accounts for time divergence in a way that treats both players symmetrically and applies to all omega-regular winning conditions. We prove that the ability to take each other by surprise adds extra power to the players. For the case that the games are specified in the style of timed automata, we provide symbolic algorithms for their solution with respect to all omega-regular winning conditions. We also show that for these timed games, memory strategies are more powerful than memorylless strategies already in the case of reachability objectives.

[18] Discounting the Future in Systems Theory

de Alfaro, Luca, Thomas A. Henzinger and, Rupak Majumdar
Proc. of the 30th International Colloquium on Automata, Languages, and Programming (ICALP), 2003.

Abstract: Discounting the future means that the value, today, of a unit payoff is 1 if the payoff occurs today, a if it occurs tomorrow, a^2 if it occurs the day after tomorrow, and so on, for some real-valued discount factor $0 < a < 1$. Discounting (or inflation) is a key paradigm in economics and has been studied in Markov decision processes as well as game theory. We submit that discounting also has a natural place in systems engineering: for nonterminating systems, a potential bug in the far-away future is less troubling than a potential bug today. We therefore develop a systems theory with discounting. Our theory includes several basic elements: discounted versions of system properties that correspond to the omega-regular properties, fixpoint-based algorithms for checking discounted properties, and a quantitative notion of bisimilarity for capturing the difference between two states with respect to discounted properties. We present the theory in a general form that applies to probabilistic systems as well as multicomponent systems (games), but it readily specializes to classical transition systems. We show that discounting, besides its natural practical appeal, has also several mathematical benefits. First, the resulting theory is robust, in that small perturbations of a system can cause only small changes in the properties of the system. Second, the theory is computational, in that the values of discounted properties, as well as the discounted bisimilarity distance between states, can be computed to any desired degree of precision.

[19] Decentralized Structural Acoustic Control of a Launch Vehicle Payload Fairing

Frampton, K.D., Journal of the Acoustical Society of America, Vol. 111, No. 5, Part 2, pp. 2453 .

Abstract: The development of smart structures and active noise and vibration control technologies promised to revolutionize the design, construction and, most importantly, the performance of many complex engineering. However, the early promise of these technologies has not been realized in large-scale systems primarily because of the excessive complexity, cost and weight associated with centralized control systems. Now, recent developments in MEMS sensors and actuators, along with networked embedded processor technology, have opened new research avenues in decentralized controls. Such a control system consists of numerous nodes, possessing limited computational capability, sensors and actuators. Each of these nodes is also capable of communicating with other nodes via a wired or wireless network. This results in a dramatic shift in the control system paradigm from that of a single, centralized computer to that of numerous decentralized, networked processors. This work describes the application of such a control system to the reduction of structural acoustic radiation in a launch vehicle payload fairing. A JAVA based simulation tool is employed to simulate the interactions of the physical system with the networked embedded controllers. Results will indicate the potential for such a control system as well as the limitations imposed by the networked embedded processor hardware.

[20] Decentralized Vibration Control in a Launch Vehicle Payload Fairing

Frampton, K.D., Proceedings of the ASME International Mechanical Engineering Conference and Exposition, November 2002, New Orleans, LA.

Abstract: The vibro-acoustic environment inside a launch vehicle payload fairing is extremely violent resulting in excessive development costs for satellites and other payloads. The development of smart structures and active noise and vibration control technologies promised to revolutionize the design, construction and, most importantly, the acoustic environment within these fairings. However, the early promise of these technologies has not been realized in such large-scale systems primarily because of the excessive complexity, cost and weight associated with centralized control systems. Now, recent developments in MEMS sensors and actuators, along with networked embedded processor technology, have opened new research avenues in decentralized controls based on networked embedded systems. This work describes the development and comparison of decentralized control systems that utilize this new control paradigm. The controllers are hosted on numerous nodes, possessing limited computational capability, sensors and actuators. Each of these nodes is also capable of communicating with other nodes via a wired or wireless network. The constraints associated with networked embedded systems control that the control systems be relatively simple computationally, scalable and robust to failures. Simulations were conducted that demonstrate the ability of such a control architecture to attenuate specific structural modes.

[21] Embedded Systems for the Distributed Structural Acoustic Control in a Launch Vehicle Payload Fairing

Frampton, K.D., Invited to the special session on Interior Noise in Aircraft and Rocket Fairings, Acoustical Society of America Meeting, Nashville TN May 2003.

Abstract: Numerous investigations have been conducted with the purpose of attenuating the acoustic environment within rocket payload fairings. These, to date, theoretical and experimental laboratory studies have demonstrated a great deal of success. However, practical applications to this, and other large-scale noise control problems, have been limited in their success. These limitations are due to non-scalable control systems, weight constraints and complexity. This work seeks to address these limitations by investigating the use of an array of networked embedded processors to control the interior acoustics of a rocket fairing is investigated. This networked embedded system consists of numerous computationally elements, paired with appropriate sensors and actuators, that are that communicate with each other over a wired or wireless network. The goal of the network is to minimize the interior acoustic level while expending a minimum amount of energy. Results from the simulation of such a control system will demonstrate the effectiveness of such an approach. These results will also be compared with those obtained by traditional, centralized control architectures.

[22] Distributed Group-Based Vibration Control with a Networked Embedded System

Frampton, K.D., submitted to Journal of Smart Materials and Structures, April 15, 2003.

Abstract: The purpose of this work is to demonstrate the performance of a distributed vibration control system based on a networked embedded system. The platform from

which control is affected consists of a network of computational elements called nodes. Each node possesses its own computational capability, sensor, actuator and the ability to communicate with other nodes via a wired or wireless network. The primary focus of this work is to employ existing group management middleware concepts to enable vibration control with such a distributed network. Group management middleware is distributed software that provides for the establishment and maintenance of groups of distributed nodes and that provides for the network communication among such groups. This objective is met by designing distributed feedback compensators that take advantage of node groups in order to affect their control. Two types of node groups are considered: groups based on physical proximity and groups based on modal sensitivity. The global control objective is to minimize the vibrational response of a rectangular plate in specific modes while minimizing spillover to out-of-bandwidth modes. Results of this investigation demonstrate that such a distributed control system can achieve vibration attenuations comparable to that of a centralized controller. The importance of efficient use of network communications bandwidth is also discussed with regard to the control architectures considered.

[23] Counterexample-Guided Control

Henzinger*, Thomas A., Ranjit Jhala, and Rupak Majumdar*
Proc. of the 30th International Colloquium on Automata, Languages, and Programming (ICALP), 2003.

Abstract: A major hurdle in the algorithmic verification and control of systems is the need to find suitable abstract models, which omit enough details to overcome the state-explosion problem, but retain enough details to exhibit satisfaction or controllability with respect to the specification. The paradigm of counterexample-guided abstraction refinement suggests a fully automatic way of finding suitable abstract models: one starts with a coarse abstraction, attempts to verify or control the abstract model, and if this attempt fails and the abstract counterexample does not correspond to a concrete counterexample, then one uses the spurious counterexample to guide the refinement of the abstract model. We present a counterexample-guided refinement algorithm for solving omega-regular control objectives. The main difficulty is that in control, unlike in verification, counterexamples are strategies in a game between system and controller. In the case that the controller has no choices, our scheme subsumes known counterexample-guided refinement algorithms for the verification of omega-regular specifications. Our algorithm is useful in all situations where omega-regular games need to be solved, such as supervisory control, sequential and program synthesis, and modular verification. The algorithm is fully symbolic, and therefore applicable also to infinite-state systems.

[24] Schedule Carrying Code

Henzinger , Thomas A., Christoph M. Kirsch, and Slobodan Matic
To appear, EMSOFT 2003.

Abstract: To guarantee the correct execution of a hard real-time program on a given platform, the scheduler must ensure that all deadlines are met. We introduce the paradigm of *schedule-carrying code* (SCC), where the compiler proves the existence of a feasible schedule by generating such a schedule, which is then attached to the generated code in

the form of executable instructions that remove the need for a system scheduler. In this way, the schedule is produced once, and revalidated and executed with each use. We evaluate SCC both in theory and practice. In theory, we give two scenarios, of nonpreemptive and distributed scheduling for Giotto programs, where the generation of a feasible schedule is hard, while the validation of scheduling instructions that are attached to the code is easy. In practice, we implement SCC and show that explicit scheduling instructions can reduce the scheduling overhead up to 35 percent, and can provide an efficient, flexible, and verifiable means for compiling Giotto on complex architectures, such as the TTA.

[25] HyVisual: A Hybrid System Visual Modeler

Hylands, Christopher, Edward A. Lee, Jiu Liu, Xiaojun Liu, Stephen Neuendorffer, Haiyang Zheng
Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, July 17, 2003.

Abstract: 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 information.

[26] On the Use of Graph Transformations for the Formal Specification of Model Interpreters

Karsai G., Agrawal A., Shi F., Sprinkle J., Journal of Universal Computer Science, Special issue on Formal Specification of CBS, (submitted), 2003.

Abstract: Model-based development necessitates the transformation of models between different stages and tools of the design process. These transformations must be precisely, preferably formally, specified, such that end-to-end semantic interoperability is maintained. The paper introduces a graph-transformation-based technique for specifying these model transformations, gives a formal definition for the semantics of the transformation language, describes an implementation of the language, and illustrates its use through an example.

[27] Structure and Interpretation of Signals and Systems

Lee, Edward A and Pravin Varaiya
Addison-Wesley, 2003

Abstract: This book provides an accessible introduction to signals and systems for electrical engineering, computer engineering, and computer science students, and is based on several years of successful classroom use at the University of California, Berkeley. The material starts with an early introduction to applications, well before students have built up enough theory to fully analyze the applications. This motivates students to learn the theory and allows students to master signals and systems at the sophomore level. The material motivates signals and systems through sound and images, as opposed to circuits, and as such calculus is the only prerequisite.

The book is accompanied by a robust web site with detailed notes and illustrative applets for most every topic. These applets include interactive manipulation of sound and images and making the material dynamic and understandable to all students. The book also contains extensive lab material. This lab material is based on Matlab and Simulink, and helps students build a bridge between the "what is" aspects of signals and systems as taught in the text, and the "how to" aspects of signals and systems used in the real world.

[28] Actor-Oriented Control System Design: A Responsible Framework Perspective

Liu, Jie, Johan Eker, Jörn W. Janneck, Xiaojun Liu, and Edward A. Lee.
To appear in IEEE Transactions on Control System Technology, 2003.

Abstract: Complex control systems are heterogeneous, in the sense of discrete computer-based controllers interacting with continuous physical plants, regular data sampling interleaving with irregular communication and user interaction, and multilayer and multimode control laws. This heterogeneity imposes great challenges for control system design in terms of end-to-end control performance modeling and simulation, traceable refinements from algorithms to software/hardware implementation, and component reuse. This paper presents an actor-oriented design methodology that tackles these issues by separating the data-centric computational components (a.k.a. actors) and the control-flow-centric scheduling and activation mechanisms (a.k.a. frameworks). Semantically different frameworks are composed hierarchically to manage heterogeneous

models and achieve actor and framework reuse. We introduce a notion of responsible frameworks to characterize the property that a framework can aggregate individual actor's execution into a well-defined composite execution such that heterogeneous models can be composed.

This methodology is implemented in the Ptolemy II software environment. We discuss how some of the most useful models for control system design are implemented as responsible frameworks. As an example, the methodology and the Ptolemy II software environment is applied to the design of a distributed, real-time software implementation of a pendulum inversion and stabilization system.

[29] Computer-Automated Multi-Paradigm Modeling in Control Systems Technology

Mosterman, P., Sztipanovits, J., Engell, S.,
accepted paper in IEEE Transactions on Automatic Control (to be published in 2004)

Abstract. The use of model based technologies has made it imperative for the development of a feedback control system to deal with many different tasks such as: plant modeling in all its variety; model reduction to achieve a complexity or level of abstraction suitable for the design task at hand; synthesis of control laws that vary from discrete event reactive control to continuous model predictive control, their analysis, and testing; design of the implementation; modeling of the computational platform and its operating system; analysis of the implementation effects; software synthesis for different platforms to facilitate rapid prototyping, hardware in the loop simulation, etc. Throughout these tasks, different formalisms are used that are very domain specific (e.g., tailored to electrical circuits, multi-body systems, reactive control algorithms, communication protocols) and that often need to be coupled, integrated, and transformed (e.g., a block diagram control law in the continuous domain has to be discretized and then implemented in software). Significant improvements in many aspects (performance, cost, development time) of the development process can therefore be achieved by (i) relating and integrating these formalisms, (ii) automatically deriving of different levels of modeling abstractions, and (iii) rigorous and tailored design of the different formalisms by capturing the domain (or meta) knowledge. The emerging field of Computer Automated Multi-Paradigm Modeling (CAMPaM) presented in this paper in the context of control system design, aims to develop a domain-independent formal framework that leverages and unifies different activities in each of these three dimensions.

[30] Constraint-Based Design-Space Exploration and Model Synthesis,

Neema, S., Sztipanovits, J., Karsai, G.
accepted paper EMSOFT'2003, Philadelphia, PA October 12-15, 2003

Abstract. An important bottleneck in model-based design of embedded systems is the cost of constructing models. This cost can be significantly decreased by increasing the reuse of existing model components in the design process. This paper describes a tool suite, which has been developed for component-based model synthesis. The DESERT tool suite can be interfaced to existing modeling and analysis environments and can be inserted in various, domain specific design flows. The modeling component of DESERT supports the modeling of design spaces and the automated search for designs that meet

structural requirements. DESERT has been introduced in automotive applications and proved to be useful in increasing design productivity.

[31] NP-Click: A Programming Model for the Intel IXP1200

Shah, Niraj, William Plishker, and Kurt Keutzer
Proc. Of 2nd Workshop on Network Processors (NP-2), 9th International Symposium on High Performance Computer Architectures (HPCA), Feb. 2003.

Abstract: The architectural diversity and complexity of network processor architectures motivate the need for a more natural abstraction of the underlying hardware. In this paper, we describe a programming model, NPClick, which makes it possible to write efficient code and improve application performance without having to understand all of the details of the target architecture. Using this programming model, we implement the data plane of an IPv4 router on a particular network processor, the Intel IXP1200, and compare results with a hand-coded implementation. Our results show the IPv4 router written in NP-Click performs within 7% of a hand-coded version of the same application using a realistic packet mix.

[32] Model-Based Fault-Adaptive Control of Complex Dynamic Systems

Simon, Gyula, Gábor Karsai, Gautam Biswas, Sherif Abdelwahed, Nagabhushan Mahadevan,Tivadar Szemethy, Gábor Péceli, and Tamás Kovácsházy
Proc. IEEE Instrumentation and Measurement Technology Conference, Vail, Co. May, 2003.

Abstract: Complex control applications require capabilities for accommodating faults in the controlled plant. Fault accommodation involves the detection and isolation of faults, and then taking appropriate control actions to mitigate the fault effects and maintain control. This requires the integration of fault diagnostics with control in a feedback loop. This paper discusses a generic framework for building fault-adaptive control systems using a model-based approach, with special emphasis on the modeling schemes that describe different aspects of the system at different levels of abstraction and granularity. The concepts are illustrated by a fault adaptive airplane fuel system control example.

[33] A Domain-Specific Visual Language For Domain Model Evolution

Sprinkle J., Karsai G., Journal of Visual Languages and Computing, 15, 2 (submitted), April, 2004.

Abstract: Domain-specific visual languages (DSVLs) are concise and useful tools that allow the rapid development of the behavior and/or structure of applications in well-defined domains. These languages are typically developed specifically for a domain, and have a strong cohesion to the domain concepts, which often appear as primitives in the language. The strong cohesion between DSVL language primitives and the domain is a benefit for development by domain experts, but can be a drawback when the domain evolves – even when that evolution appears insignificant. This paper presents a domain specific visual language developed expressly for the evolution of domain-specific visual languages, and uses concepts from graph-rewriting to specify and carry out the transformation of the models built using the original DSVL.

[34] Domain Evolution in Visual Languages Using Graph Transformations

Sprinkle J., Agrawal A., Levendovszky T., Shi F., Karsai G., OOPSLA, 2nd Workshop on Domain-Specific Languages, Seattle, WA, November 4, 2002.

Abstract: Domain-specific visual programming is a convenient way to hide complexity from the pro-grammer. The careful thought and design that precede the development of any domain-specific visual language restrict the programmer from illegal formalisms, and allow for the rapid determination of the validity of the "program". Usually, the domain-specific visual language is designed and produced using a metamodel of some sort. However, changes in the metamodel can lead to disastrous results when attempting to process domain-models built according to the original specifications. This paper presents a visual language for trans-forming domain-models that can express the mapping between the meta-models of the "in-put" (i.e. the "old" language) and the "output" (i.e. the "new" language), and uses graph-rewriting techniques to transform the "old" domain-models into the appropriate "new" form.

[35] Model-Integrated Computing Infrastructure for Fault Management

Sztipanovits, J., International Conference on Principles of Diagnosis (DX'2003), Arlington, VA June, 2003 (invited talk)

Abstract. In functional design of large systems, complexity is managed by vertical and horizontal composition. In vertical composition, systems are designed in layers utilizing fundamentally different technologies. Commonly used layers in information systems are: Material, Device and Circuit Layer, Hardware/System Layer, OS/Communication Layer, Middleware Layer(s), Application Layer. The individual layers are designed by using layer-specific abstractions and composition technologies. The core design task on each layer is to use resources provided by a lower layer to implement functionalities demanded by a higher layer. Horizontal composition is performed in a single layer for creating aggregate components using the dominant compositionality principle of the layer. The presentation summarizes research directions in model-integrated computing for addressing multi-layer optimization of fault management architectures in resource constrained embedded systems.

[36] Decentralized Vibration Control with Networked Embedded Systems

Tao, T. and K.D. Frampton, Proceedings of the SPIE 10th Annual International Symposium on Smart Structures and Materials, March 2-6, 2003, San Diego, California.

Abstract: The results of simulations to demonstrate decentralized vibration control with a networked embedded system are presented in this work. Conventional vibration control designs rest on centrality, and the central processor deals with information of the entire system. When large-scale systems are considered, decentralized vibration control system provides an alternative design. The simulated system in this work is a simply supported beam that is collocated with 50 localized processor nodes which can communicate with each other. Each node will calculate and apply the control force to control the beam vibration according to the shared sensor information among the nodes and an optimal direct velocity feedback algorithm. The simulation results demonstrate that decentralized vibration control can achieve a global control objective, making it suitable for large-scale

systems. The effects of network communication delay and feedback architecture on control performance are demonstrated.

2.3. Project Training and Development

As part of setting up CHESS (the new UCB Center for Hybrid and Embedded Software Systems), we have created a CHESS Software Lab, which is focused on supporting the creation of publication-quality software supporting embedded systems design. The lab is physically 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

Our agenda is 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. Based on the ongoing effort at UCB, we have set out to rearchitect and retool undergraduate teaching at the participating institutions, and to make the results widely available to encourage critical discussion and facilitate adoption. In addition, have recruited new undergraduate students (mostly juniors) from minority institutions through the established REU programs SUPERB-IT at UCB and SURGE at VU to participate in the research of the project.

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.

We have published a textbook, Structure and Interpretation of Signals and Systems [33], that supports this approach. This textbook has been tested in two offerings of EECS 20n, a sophomore-level course at UCB that introduces hybrid systems, automata-based reasoning about discrete systems, and frequency-domain reasoning about continuous and discrete-time signals and systems. The textbook pays special attention to the transition between the more recent hybrid systems material and the more traditional systems material.

At many institutions, introductory courses like this are quite large (at Berkeley, each offering draws approximately 200 students). 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. This staff has initially focused on building 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 will generate PDF files for exams, 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 been extending and upgrading these applets.

Finally, our staff has been creating a suite of Powerpoint slides for use by instructors. One of the challenges has been to integrate animated content like that in the applets in a robust and portable way.

2.4.2. Undergraduate Curriculum Insertion and Transfer

We have formed a 'curriculum council' to advise us and help us transition our curriculum innovations to other institutions. The term for the CC is 3 years, and the charter members of the board, effective February 15, 2003, are:

- Thomas Boegel, City College of San Francisco, tboegel@ccsf.edu
- Kenneth Derucher. Chico State, KDerucher@csuchico.edu
- Roger Doering, Cal State Hayward (CS), rdoering@csuhayward.edu
- Ping Hsu, San Jose State, phsu@email.sjsu.edu
- Sung Hu, San Francisco State, shu@sfsu.edu
- George V. Krestas, DeAnza College, krestasGeorge@fhda.edu
- Thomas Murphy, Diablo Valley College, TMurphy@contracosta.cc.ca.us
- Dan Pitt, Santa Clara University, dpitt@scu.edu

- Saeid Rahimi, Sonoma State, saeid.rahimi@sonoma.edu
- Helen Zong, Cal State Hayward (Engineering), hzong@csuhayward.edu

Ex-Officio members (UC Berkeley Personnel):

- Bob Giomi, UC Berkeley, giomi@uclink4.berkeley.edu
- Paula Hawthorn, UC Berkeley, pbhawthorn@mindspring.com
- Edward A. Lee, UC Berkeley, eal@eecs.berkeley.edu
- Neil Turner, UC Berkeley, net@eecs.berkeley.edu
- Pravin Varaiya, UC Berkeley, varaiya@eecs.berkeley.edu

The curriculum council can be reached at curriculum@chess.eecs.berkeley.edu

The mission statement for the CC is as follows:

> The Curriculum Council of the Chess Center provides strategic leadership and advocacy within the community for curriculum modernization and reform in engineering and computer science. It serves as the principal liaison between the Chess Center and education professionals, and will strive to facilitate and promote improvements in the way systems science is taught.

The first Curriculum Council meeting was held Saturday, March 1, 2003. Attending were:

- Thomas Boegel, City College of San Francisco
- Paula Hawthorn, UC Berkeley
- Ping Hsu, San Jose State
- Sung Hu, San Francisco State
- Edward Lee, UC Berkeley
- Dan Pitt, Santa Clara University
- Neil Turner, UC Berkeley
- Pravin Varaiya, UC Berkeley
- Helen Zong, Cal State Hayward

This working meeting was intended to establish the objectives of the curriculum council and to develop a plan of action for curriculum development. We hope to develop further dialogue and collaborations with California Colleges in teaching system science.

There were five topics discussed:

- What the curriculum council should do.
- What is Chess.
- What is EECS 20 (course content, lab, course evaluation).
- What changes are anticipated in the upper division.
- What could a summer institute/workshop provide.

Details can be found at http://chess.eecs.berkeley.edu/publications/talks/03/cckickoff3-03.pdf.

The group agreed that restructuring the Systems Science courses is necessary. Some institutions may take a long time to do it, however, due to a lack or resources and due to inertia. The current California budget crisis also puts many of the represented institutions in a very difficult position, with no resources for introducing new courses, hiring new instructors, or developing new laboratory infrastructure. It was decided that a summer institute/workshop in the summer of 2003 would be premature, and that instead, the CC should convene meetings at each of the participating institutions to help develop a consensus for action. The first such meeting is scheduled at Santa Clara University for July 17, 2003.

Despite the overwhelming obstacles, Roger Doering of Cal State Hayward has gotten approval for a pilot course in the Winter quarter of 2003/2004 that follows UCB's EECS 20n. We have agreed to assist with the expense of obtaining software to equip the lab and to assist with adaptation of the course for a quarter format. Cal State Hayward is a unique position because it is starting a new engineering program, and hence does not have the legacy inertia that makes it difficult to change.

2.4.3. 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 increase diversity in the graduate school pipeline by affirming students' motivation for graduate study and strengthening their qualifications.

SUPERB-IT participants spent eight weeks at UC Berkeley during the summer (June 16 - August 8, 2003) 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 is supporting a group six SUPERB-IT students, and has organized projects (described below) in hybrid and embedded software systems technology, primarily in the area of software tools supporting the design process. The students are being hosted by the Chess center at Berkeley (Center for Hybrid and Embedded Software Systems).

SUPERB-IT participants receive a $3,500 stipend, room and board on campus in the International House, and up to $600 for travel expenses. In addition, Chess has provided these students with one laptop computer each, configured with appropriate software, plus laboratory facilities for construction of associated hardware.

The students being supported at Berkeley are Antonio Yordan-Nones, Ismael Sarmiento, Rakesh Reddy, Colin Cochran, Mike Okyere, and Philip Baldwin. At the time of writing this report, the

projects are about halfway completed. The six students are building a suite of applications and infrastructure for embedded systems design using the Ptolemy II software infrastructure. Their eight week projects began with an intensive group training in the Chess software lab that familiarized them with the use a CVS code repository, the Eclipse integrated development environment, construction of applications in Ptolemy II, and design and construction of actors for Ptolemy II. They have been guided to use an Extreme Programming (XP) software engineering style, which includes pair programming, extensive use of automated test suites, and design and code reviews. All but one of the students has sufficient experience with Java programming that little time has been required for familiarization with Java. The one student with little Java experience (Philip Baldwin) is focused on building models of distributed, wireless, real-time systems in Ptolemy II. These models will use infrastructure built in Java by the other students. In the XP context, he functions as the 'customer.' Three graduate student mentors have been identified to facilitate the process, and the operation is being coordinated and directed by Professor Edward Lee. Although the students are working together and interacting extensively, each will be responsible for an individual project, as described below:

Student: Antonio Yordan-Nones
Project: Interactive embedded systems showcase

This student has a strong interest in art and technology, with background building applications using Java servelets and server pages, video and sound. His responsibility is to design and construct an embedded systems showcase that will occupy a glassed-in-case outside the Chess software lab. This showcase will include a computer, microphone, video camera, display, X-10 devices to control lights, and possibly computer-controlled motors. The objective will be to use one or more Ptolemy II applications to engage viewers of the showcase in interactive displays, using for example video tracking and audio stimulus as input. The student will be given freedom to be creative, and will be encouraged to create a 'whimsical embedded system' that showcases techniques created by the other students in the team. As such, this student has the role of a 'customer' in the XP scenario.

Student: Philip Baldwin
Project: Wireless systems modeling

This is the one student with no Java experience. However, he has been involved in a project that modeled bluetooth devices at the networking and application level. His responsibility is to construct Ptolemy II models of distributed wireless embedded systems such as sensor nets and web-integrated embedded applications. This student has the role of a 'customer' in the Extreme Programming scenario.

Student: Ismael Sarmiento
Project: Actor-oriented construction of interactive 2-D graphics

This student has experience using Java 2-D to build interactive graphics-intensive games. His responsibility is to build a suite of Ptolemy II actors that construct and dynamically morph 2-D scenes, and to show how these actors can be used to build customized user interfaces and displays for embedded systems models. The resulting graphics infrastructure can be used by the

first student in the interactive showcase and by the wireless systems modeling project for animated interactive displays of the models used in those contexts.

Student: Rakesh Reddy
Project: Secure transport of mobile models and data in distributed applications

This student has experience with encryption and decryption technologies and with Java software design. His responsibility is to construct Ptolemy II actors support secure distributed models, and to demonstrate their use in with mobile, web-integrated applications. The resulting technology can be used to create a web interface for the showcase to be constructed by the first student, and by the second student for modeling affects of security strategies in distributed embedded systems.

Student: Colin Cochran
Project: Actor-oriented design of smart-home systems

Like the first student, this student has an interest in art and technology, with experience in avionics software testing, web page design and construction, and Java software design. His responsibility is to create a suite of Ptolemy II actors that interact through the serial port of a host computer with X-10 controllers (which communicate over power lines to control lights and electrical devices) and motor controllers. The resulting technology can be used to control lights and motors in the 'showcase.'

Student: Mike Okyere
Project: Actor-oriented design of web-integrated string manipulation

This student has quite a bit of Java experience, primarily with e-commerce applications. His responsibility is to construct Ptolemy II actors for processing textual data, including for example XML data and text embedded in HTTP coming from HTML forms. This can be used as part of the web interface for the showcase and well as for building realistic distributed models for the wireless systems modeling project.

Student: Iyibo Jack - University of Washington (Sangiovanni's group)
Project: Platform Based Reconfigurable Hardware Exploration via Boolean Constraints

This project looks to take a high level description of an application's requirements and transform this via a series of constraints into an abstraction of the possible configurations of a reconfigurable hardware device. Taking this abstraction (a platform), it then estimates what the performance of various instances of this platform would be on the device (Cypress Semiconductor's PSOC). This methodology is both top down and bottom up in its use of constraints and performance estimation. It frames the construction of platform instances as Boolean constraint formulations and solves them using the principles of Boolean Satisfiability.

2.4.4. 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/fountain/Teaching/

In the SIPHER activities, we have organized a summer internship for six participants from underrepresented groups. The students are organized into three groups who solve different embedded software development problems. We have developed a small modeling language to enable the modeling of embedded systems, we have created two implementations of a run-time platform (one in Java, one in C++), and created model transformers that map models expressed in the modeling language into code that executes on the run-time platform. The students are using the 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 projects and the students supported at Vanderbilt are:

- 'Visual Tracking': Bina P. Shah, Edwin Vargas, and Trione Vincent (REU),
- 'LEGO Mindstorm Robot Control': Rachael A. Dennison, David Garcia, and Danial Balasubramanian (REU),
- 'TAB Robot Control': Michael J. Rivera Jackson, Nickolia Coombs (REU),
- 'Control of Adaptive Structures': Shantel Higgins (with Efosa Omojo).

The students are working on four small, team-oriented projects related to development of embedded software. In this work they are using software tools available at ISIS, and they are being supervised by professors and senior graduate students. During first few weeks they did undergo rigorous training to learn how to use our design tools. The training was provided by lecturers who deliver our Model-Integrated Computing classes. All of the students have backgrounds in programming, and thus are able to solve the project problems. Similarly to UCB, three graduate student mentors have been identified, who did assist and guide the student projects. During the Spring semester the three mentors have created prototype solutions for the projects that serve as 'reference' for the student projects. The brief bio of the students and the description of projects are below.

Student: Bina P. Shah

Bina is from Birmingham, Alabama, where she is a senior at the University of Alabama at Birmingham majoring in computer science. She is a member of the Golden Key International Honor Society, Phi Kappa Phi Honor Society, and is in the National Society of Collegiate Honors. She is very involved on her campus as well. She is part of Trail Blazers, UAB's official student recruitment team for the past 2 years, as well as serving as Vice-President for the International Mentors program. She is also actively involved in the Association for Computing Machinery (ACM), where she represented UAB's C++ team at he Southeast USA Regional Programming Contest. As a community servant, she participates with the Indian Cultural Association (ICA) doing volunteer work at homeless shelters and participating in canned food

drives. Bina wants to pursue graduate study in Electrical and Computer Engineering, specializing in the design and development of new software.

Student: Edwin Vargas

Edwin is a senior at Middle Tennessee State University, where he is a double major in computer science and mathematics. He is originally from Bogota, Colombia, where political and social unrest forced him to come to the United States. Edwin has been involved in martial arts for over seventeen years, specializing in Tae-Kwon-Do, and he practices with the martial arts club at MTSU. He has been teaching and competing actively in the time, and he won the national tournament in 1996 and 1997. Also at MTSU, Edwin is a member of the Hispanic Student Association and the local chapter of the Association for Computer Machinery (ACM). He has been on the Dean's list at MTSU and is a 2002-2003 recipient of the National Science Foundation CSEM scholarship. Edwin has a strong background in computer architecture and systems design and hopes to use his mathematical and computer science skills to pursue a PhD in Computer Science. He currently lives in Murfreesboro with this lovely wife.

Student: Trione Vincent

Trione is a rising senior at Fisk University double majoring in Computer Science and Computer Engineering. She is from New Orleans, Louisiana. She has spent her first three years at Fisk and will complete her engineering degree at Vanderbilt. While at Fisk, Trione is a member of the Big Sistas mentoring program and the Fisk University Pep Squad. She has received a Fisk Academic Scholarship and a scholarship from NASA. Trione is interested in research especially in the field of embedded systems and software. She wants to build her career working in the hardware aspect of computer science and engineering.

Project: Visual Tracking

Bina, Edwin, and Trione are working on a project that creates a control system for the visual tracking of objects using a PC and a remote controller camera. The objective is to create a model of the control system, develop the individual components (e.g. camera controller, object recognition module, etc), and then use the model-integrated computing tools to put together the final application.

Student: Rachael A. Dennison

Rachael is a senior at the University of Alabama at Birmingham, where she is also majoring in computer science. Her hometown is Greenville, Alabama. Rachael is very active and loves the outdoors. She enjoys fishing, hunting, swimming, snorkeling, scuba diving, reading, fossil hunting, and camping. She excels in academics at UAB by being on the Presidential Honor Roll and being a member of the Phi Kappa Phi Honor Society as well as the Golden Key International Honor Society. Also, she was nominated by the Computer Science Department for the Dean's Award. After graduation, Rachael wants to work in the field of software engineering and research into ways to make software more in tune with the physical environment that it

describes. Rachael wants to develop software applications and model hardware design to handle real-time information in a safe, reliable, and accurate way.

Student: David Garcia

David is a rising senior at Vanderbilt University double-majoring in Computer Science and Mathematics. He is originally from California but now resides in Las Vegas. David is active on campus by being a member of the Society of Hispanic and Professional Engineers and the Vanderbilt Association of Hispanic Students, where he currently serves as a board member. He has also used his computer skills by working with Vanderbilt's Information Technology Services (ITS) as a support technician and helped incoming freshmen with configure their network system and troubleshooting local problems. David has also been honored as receiving High Honors on the Dean's list at Vanderbilt. David's career interests lie in the area of game development and design. He wants to pursue graduate degrees in computer science specializing in gaming software and conduct research in artificial intelligence and computer graphics. Because of his avid interest in games, he received his PlayStation Technician certification working for PlayStation as a Level 1 Support specialist last summer as part of the E3 summer gaming conference.

Student: Daniel Balasubramanian

Daniel is a rising senior at Tennessee Technological University majoring in Computer Science. He is originally from Nashville, Tennessee. Daniel is a member of several organizations on campus, including the Association for Computing Machinery (ACM) and the Jazz Band, as well as being very active in the Honors program at Tennessee Tech. He is on the chair of the Tutoring Committee and a member of the Ecology Committee and the Program Big-Sib. Academically, he has received several honors and distinctions. These include a NASA Scholarship, a Rotary Club Scholarship, a TTU Housing Scholarship, and the Earl McDonald Academic Achievement Award, which covers full tuition at Tennessee Tech. Daniel has a real passion for learning, not only in his but also in other areas of science and mathematics. He has done extensive work in website development at Vanderbilt and at TTU. He is very interested in research, specifically in NASA's Gravity Probe B project, where he would like to combine his computer programming skills with the physical aspects that actually describes the system.

Project: LEGO Mindstorm Robot Control

Rachael, David, and Daniel are working on developing model-based control software for Lego robots. The challenge problem is to develop the models for the control software that will allow the robot to navigate in an environment, react to unforeseen objects (obstacles) and execute exploration tasks. They are using a modeling language for creating high-level models of the controllers, develop the code for individual components, and generate the full Java code for the controller to be executed on the RCX.

Student: Michael J. Rivera-Jackson

Michael is a rising junior on a full academic scholarship at Morehouse College in Atlanta, Georgia, where he is a double major in Computer Science and Spanish. Michael is originally from Belle Chasse, Louisiana. He is very involved on campus at Morehouse, participating in the Louisiana Club, the Spanish, French, and Japanese Clubs, SGA, the Feminist Majority Leadership Alliance, and the Hip-Hop Collective. In addition to these organizations, he has time to give back to his community by volunteering at the Charles Drew Charter Elementary School, mentoring and tutoring children in reading, mathematics, and science, for which he was recognized as Mentor of the Month in November of 2002. Michael's hobbies include surfing the net, learning languages, poetry, reading, and composing. Michael is interested in internet research and wants to continue to give back to his community by aspiring to become CEO of a software company that produces learning software specifically for children to explore all different types of areas.

Student: Nickolia S. Coombs

Nickolia is a junior at North Carolina A&T State University where is a computer science major. He is originally from Jacksonville, North Carolina. He enjoys following OTC stocks, racquetball, and public speaking. On campus, he is a member of the National Society of Black Engineers, the History Club, and is a tutor for in discrete mathematics and computer science. Also, he is the fund-raising chair for the Association for Computing Machinery. He has been on the Dean's list and is a recipient of a NSBE Scholarship, the Honors 4.0 award, honors in the ACM Programming Competition, and participated in the IBM Project Breakthrough Summer internship last summer. Nickolia's research interests include understanding more about embedded software, especially in the mathematical aspect of simulation. After graduation, Nickolia wants to work in industry, but is interested in also pursuing graduate degrees in computer science and engineering.

Project: TAB Robot Control

Michael and Nickolia are working on an another robot control problem. Their robot has better sensors, and it is much smaller than the LEGO robot. They are building embedded software for the robot using model-based techniques that allow the robot to solve a maze problem, as well as build a map of the maze. They are using the same design tools as the other projects but in a different problem setting.

Student: Shantel Higgins

Shantel is a rising senior at Vanderbilt University majoring in Electrical Engineering. Shantel is originally from Sugarland, Texas, a small suburb of Houston. She is currently the treasurer of her sorority, Delta Sigma Theta and was the chair of their First Annual Aids Awareness Walk. She is the community service chair for the Black Student Alliance at Vanderbilt as well as a volunteer for the local YMCA in Nashville. Shantel is an active member of the National Society of Black Engineers and the Society of Women Engineers, as well as a coach for an intramural women's basketball team. Shantel has received honors from the School of Engineering at Vanderbilt and she is the recipient of the Sam McCleskey Engineering Honor Scholarship. Shantel has enjoyed her time at Vanderbilt and she credits her interest and enthusiasm in

research and technology from her Vanderbilt curriculum. Shantel's interest is in the field of wireless communications, sparked by her semiconductors class and integrated circuit design and fabrication class. She hopes to pursue a master's and doctoral degrees in this area.

Project: Control of Adaptive Structures

Shantel (with Efosa Ojomo, who is independently supported) are working on developing a real-time controller for a 'smart structure': a vibrating steel beam. The objective is to create a small, embedded system that detects the onset of vibrations in a beam and actively compensate for them by acting on the beam. First, they build a simulation model for the plant and the controller in Simulink/Stateflow. Once the control algorithm is determined that will create an implementation of it on a PC-104 embedded processor platform. The physical implementation includes a piezo element, which acts both as sensor and actuator.

3. Publications and Products

3.1. Journal Publications

- Abdelwahed, S., G. Karsai, and G. Biswas, "Online Safety Control of a Class of Hybrid Systems," *Proc. 41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2002.

- Agrawal A., Karsai G., Ledeczi A., "An End-to-End Domain-Driven Development Framework," to appear in *8th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, (under revision), Anaheim, California, October 26, 2003.

- Agrawal A., Levendovszky T., Sprinkle J., Shi F., Karsai G., "Generative Programming via Graph Transformations in the Model-Driven Architecture," *OOPSLA*, *Workshop on Generative Techniques in the Context of Model Driven Architecture,* Seattle, WA, November 5, 2002.

- Balister, Paul, Béla Bollobás, Mark Walters, "Continuous Percolation," Submitted to *Annals of Applied Probability*, 2003.

- Balogh ,Jozsef, and Béla Bollobás, "Sharp thresholds in Bootstrap Percolation," to appear in *Physica*, 2003.

- Benveniste, Albert, Luca P. Carloni, Paul Caspi, and Alberto L. Sangiovanni-Vincentelli, "Heterogeneous Reactive Systems Modeling and Correct-by-Construction Deployment," to appear, *EMSOFT* 2003.

- Berger, Noam, Béla Bollobás, Christian Borgs, Jennifer Chayes, and Oliver Riordan, "Degree Distribution of the FKP Network Model," *Proc. ICALP* 2003, Lecture Notes in Computer Science, LNCS 2719, Springer-Verlag, 2003.

- Biswas, G., G. Simon, G. Karsai, S. Abdelwahed, N. Mahadevan, T. Szemethy, J. Ramirez, G. Péceli and T. Kovácsházy, "Self Adaptive Software for Fault Adaptive Control," *Proc. International Workshop on Self Adaptive Software*, Washington D.C., June 2003.

- Bollobás, Béla, Christian Borgs, Jennifer Chayes, and Oliver Riordan, "Directed Scale-free Graphs," to appear in *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, 2003.

- Bollobás, Béla, Peter Keevash and Benny Sudakov, "Multicolored Extremal Problems," Submitted.

- Bollobás, Béla, and Oliver Riordan, "Coupling Scale-free and Classical Random Graphs," to appear in *Internet Mathematics*, 2003.

- Bollobás Béla., and A.D. Scott, "Max Cut for random graphs with a planted partition," Submitted to *Combinatorics, Probability and Computing*, 2003.

- Chakrabarti, Arindam, Luca de Alfaro, Thomas A. Henzinger, and Marielle Stoelinga, "Resource Interfaces," to appear, *EMSOFT* 2003.

- Chatterjee, Krishnendu, Marcin Jurdzinski, and Thomas A. Henzinger, "Simple Stochastic Parity Games," In *Proceedings of the International Conference for Computer Science Logic* (CSL), 2003.

- Chatterjee, Krishnendu, Di Ma, Rupak Majumdar, Tian Zhao, Thomas A. Henzinger, and Jens Palsberg, "Stack Size Analysis for Interrupt-driven Programs," *Proceedings of the Tenth International Static Analysis Symposium* (SAS), 2003.

- de Alfaro, Luca, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Marielle Stoelinga, "The Element of Surprise in Timed Games," *Proceedings of the 14th International Conference on Concurrency Theory* (CONCUR), 2003.

- de Alfaro, Luca, Thomas A. Henzinger and, Rupak Majumdar, "Discounting the Future in Systems Theory," *Proc. of the 30th International Colloquium on Automata, Languages, and Programming* (ICALP), 2003.

- Frampton, K.D., "Decentralized Structural Acoustic Control of a Launch Vehicle Payload Fairing," *Journal of the Acoustical Society of America*, Vol. 111, No. 5, Part 2, pp. 2453.

- Frampton, K.D., "Decentralized Vibration Control in a Launch Vehicle Payload Fairing," *Proceedings of the ASME International Mechanical Engineering Conference and Exposition*, November 2002, New Orleans, LA.

- Frampton, K.D., "Embedded Systems for the Distributed Structural Acoustic Control in a Launch Vehicle Payload Fairing," Invited to the special session on *Interior Noise in Aircraft and Rocket Fairings*, Acoustical Society of America Meeting, Nashville TN May 2003.

- Frampton, K.D., "Distributed Group-Based Vibration Control with a Networked Embedded System," submitted to *Journal of Smart Materials and Structures*, April 15, 2003.

- Henzinger, Thomas A., Ranjit Jhala, and Rupak Majumdar, "Counterexample-Guided Control," *Proc. of the 30th International Colloquium on Automata, Languages, and Programming* (ICALP), 2003.

- Henzinger , Thomas A., Christoph M. Kirsch, and Slobodan Matic, "Schedule Carrying Code," to appear, *EMSOFT* 2003.

- Karsai G., Agrawal A., Shi F., Sprinkle J., "On the Use of Graph Transformations for the Formal Specification of Model Interpreters," *Journal of Universal Computer Science*, Special issue on Formal Specification of CBS, (submitted), 2003.

- Liu, Jie, Johan Eker, Jörn W. Janneck, Xiaojun Liu, and Edward A. Lee, "Actor-Oriented Control System Design: A Responsible Framework Perspective," To appear in *IEEE Transactions on Control System Technology*, 2003.

- Mosterman, P., Sztipanovits, J., Engell, S., "Computer-Automated Multi-Paradigm Modeling in Control Systems Technology," accepted paper in *IEEE Transactions on Automatic Control* (to be published in 2004).

- Neema, S., Sztipanovits, J., Karsai, G., "Constraint-Based Design-Space Exploration and Model Synthesis," accepted paper *EMSOFT'2003*, Philadelphia, PA October 12-15, 2003.

- Shah, Niraj, William Plishker, and Kurt Keutzer, "NP-Click: A Programming Model for the Intel IXP1200," *Proc. Of 2nd Workshop on Network Processors (NP-2), 9th International Symposium on High Performance Computer Architectures* (HPCA), Feb. 2003.

- Simon, Gyula, Gábor Karsai, Gautam Biswas, Sherif Abdelwahed, Nagabhushan Mahadevan,Tivadar Szemethy, Gábor Péceli, and Tamás Kovácsházy, "Model-Based Fault-Adaptive Control of Complex Dynamic Systems," *Proc. IEEE Instrumentation and Measurement Technology Conference*, Vail, Co. May, 2003.

- Sprinkle J., Karsai G., "A Domain-Specific Visual Language For Domain Model Evolution," *Journal of Visual Languages and Computing*, 15, 2 (submitted), April, 2004.

- Sprinkle J., Agrawal A., Levendovszky T., Shi F., Karsai G., "Domain Evolution in Visual Languages Using Graph Transformations," *OOPSLA*, 2nd Workshop on Domain-Specific Languages, Seattle, WA, November 4, 2002.

- Sztipanovits, J., "Model-Integrated Computing Infrastructure for Fault Management," *International Conference on Principles of Diagnosis* (DX'2003), Arlington, VA June, 2003 (invited talk).

- Tao, T. and K.D. Frampton, "Decentralized Vibration Control with Networked Embedded Systems," *Proceedings of the SPIE 10th Annual International Symposium on Smart Structures and Materials*, March 2-6, 2003, San Diego, California.

3.2. Books and Other Non-Periodical, One-Time Publications

- Agrawal A., Karsai G., Shi F., "Interpreter Writing using Graph Transformations," Technical Report ISIS-03-401, 2003.

- Hylands, Christopher, Edward A. Lee, Jiu Liu, Xiaojun Liu, Stephen Neuendorffer, Haiyang Zheng, "HyVisual: A Hybrid System Visual Modeler," Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, July 17, 2003.

- Lee, Edward A and Pravin Varaiya, *Structure and Interpretation of Signals and Systems*, Addison-Wesley, 2003

3.3. Internet Dissemination

The Chess website, http://chess.eecs.berkeley.edu, includes publications and software distribution. 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.

3.4. Other Specific Product

The following software packages have been made available on the Chess website, http://chess.eecs.berkeley.edu:

- HyVisual 2.2, a block-diagram editor and simulator for continuous-time and hybrid systems. 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 2.2 is a subset of Ptolemy II.

- CHIC (Checking Interface Compatibility): a modular verifier for behavioral compatibility of software and hardware component interfaces. Chic is a software tool that allows the

specification of behavioral interfaces for software and hardware components, either stand-alone or as annotations to Java code. Interfaces may express synchronous and asynchronous communication constraints, software call graph and pre/postcondition contraints, as well as resource (e.g. memory, power) use constraints. Chic checks if the constraints of two or more interfaces are compatible, and if so, it derives the constraints for the interface of the composite system.

4. Contributions

4.1. Within Discipline

4.1.1. Hybrid Systems Theory

- We introduced discounting, which is well-studied in economics, into the analysis of software and hardware systems. In this way we achieve a robust and computational theory of discrete dynamical systems.

- We extended the successful theories of abstract interpretation and counterexample-guided abstraction refinement from verification problems to control problems.

- We developed compositional formalisms for expressing and checking resource (time, space, power) constraints of software and hardware components.

- We developed and evaluated the new paradigm of a virtual machine for real-time scheduling, which replaces traditional real-time operating systems and offers greater flexibility and efficiency.

- We have identified a difficulty that many programs have where exceptions are not handled effectively and resources are not released, and we have devised a way that programs can easily specify resources that must be released even in fault conditions.

- We have developed a modeling language that is based on the physical phenomena of energy flows between components. This language is component oriented and explicit about causality.

- We have applied stochastic hybrid systems to the modeling of biological systems.

- We have constructed precise mathematical models of large-scale, graph-structured networks, and have applied these models to build a directed graph model of the world-wide web.

- We have developed algorithms for diagnosing hybrid systems using a combination of qualitative and quantitative reasoning.

4.1.2. Model-Based Design

- We have developed a structure for meta-modeling of domain-specific modeling languages.
- We have constructed an operational semantics of hybrid systems that emphasizes avoiding accidental nondeterminism, thus enabling repeatable simulation.
- We have developed a programmers model for network processors that is based on the Click abstraction, which extends dataflow formalisms with explicit push/pull semantics.
- We have extended the hybrid system interchange format (HSIF) to represent faulty behavior.
- We have developed meta-modeling method for extending domain-specific modeling languages to represent design spaces.
- We have developed a UML-based way to specify graph transformations and have built a graph rewriting engine (GRE) that functions as a virtual machine for graph transformers.
- We have built a code generator that translates graph transformation specifications into executable code, and have built a debugger tool for GRE.
- We have further developed our Metropolis tool to support jointly verifying functional behavior and exploring mapping to architectural resources.

4.1.3. Advanced Tool Architectures

- We have developed and released a hybrid system visual modeling tool called HyVisual and used it to explore the operational semantics of hybrid systems.
- We have identified causality as a key interface property that must be propagated through hierarchy, much the way type information is propagated through hierarchy.
- We are developing a mathematical framework for synchronization policies between components that gives designers freedom of choice between different synchronization policies at different stages of the design process.
- We have developed algorithms and built a tool (Chic, checking interface compatibility) for checking whether a set of interfaces that specify various constraints are compatible. We have created an experimental framework that integrates Chic into Ptolemy II to allow experimentation with interface specification and compatibility checking in design.
- We have developed a novel method for debugging formal temporal specifications and have created a tool called Cable that helps debug specifications produced by Strauss, our specification miner.

- We have created a virtual machine for scheduling that can be used to replace a real-time operating system (or its scheduler) in Giotto implementations. Giotto is our time-triggered language for modal real-time systems.

- We have developed a tool chain for constraint-based design-space exploration.

4.1.4. Experimental Research

- We have made progress on models for formation maneuvers of unmanned aerial vehicles (UAVs) for national and homeland security.

- We have developed a means for modifying the control software in fly-by-wire aircraft to restrict the airspace that an aircraft will fly into. The scheme is called Soft Walls.

- We have developed an online approach to the safety control of a general class of hybrid systems (switching hybrid systems).

- We have made progress on methods for computing worst-case execution time bounds for embedded software.

- We have applied our Metropolis design environment to developing novel micro-architectures, with emphasis on refinement verification.

- We have made progress on low energy coordination in wireless ad-hoc sensor networks.

- We have devised a simple modeling language for the component-based construction of event-driven embedded systems, implemented a virtual machine for it, and developed a translator that generates for the virtual machine. The language and its tools have been used in the SIPHER program by undergraduates.

- We have designed and constructed a simple smart structures based experimental platform. Instructions for construction of this test bed will be posted online soon.

- We have begun designing more complex structural control experimental platforms.

- We have developed embedded software foundations for the implementation of structural control with distributed systems. These include improved network services for high-bandwidth real-time feedback control.

4.2. Other Disciplines

- We developed new efficient algorithms for solving stochastic games, which have applications in other fields such as economics.

4.3. Human Resource Development

The project has engaged a large number of graduate students.

4.4. Research and Education

See the outreach portion of this report.

4.5. Beyond Science and Engineering

Embedded software is everywhere.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-00225610

UNIVERSITY OF CALIFORNIA AT BERKELEY
VANDERBILT UNIVERSITY
UNIVERSITY OF MEMPHIS

MAY 31, 2004

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2003 – MAY 31, 2004

Contents

Contents 2
1. Participants 5
1.1. People 5
1.2. Partner Organizations 7
1.3. Collaborators 7
2. Activities and Findings 8
2.1. Project Activities 8
2.1.1. Hybrid Systems Theory 8
2.1.1.a. Deep Compositionality 9
Non-Zero-Sum Games as Compositional System Models *9*
Hybrid Systems Modeling Tools: a Survey *9*
Composing Quantitative Aspects of Systems *9*
2.1.1.b. Robust Hybrid Systems 10
Design and Verification of Robust System Models *10*
Affine Hybrid Systems *10*
Blowing Up Affine Hybrid Systems *11*
2.1.1.c. Computational Hybrid Systems 12
Algorithms for the Control of Stochastic Systems *12*
Reach Set Calculations using Ellipsoidal Approximations *12*
A Deterministic Operational Semantics for Hybrid System Simulations *12*
2.1.1.d. Stochastic Hybrid Systems 13
Application of Stochastic Hybrid Systems to Biological Systems *13*
2.1.2. Model-Based Design 14
2.1.2.a. Composition of Domain Specific Modeling Languages 15
Metamodeling *15*
Compositional Metamodeling *16*
Integration of Metamodeling with Hybrid Systems *16*
Semantic Foundations for Heterogeneous Systems *16*
Generalized causality analysis *17*
2.1.2.b. Extensions to Distributed Models of Embedded systems 18
Compositional Theory of Heterogeneous Reactive Systems *18*
2.1.2.c. Model Transformation 19
Meta Generator Technology *19*
Pattern-Based Model Synthesis *19*
2.1.2.d. Real-Time Programming Models 19
Event-Triggered Programming *19*
2.1.3. Advanced Tool Architectures 19
2.1.3.a. Syntax and Semantics 21
Modularity Mechanisms in Actor-Oriented Design *21*
Code Generation from Actor-Oriented Models *21*
Metropolis Framework *22*
2.1.3.b. Interface Theories 23
A Component Model for Heterogeneous Systems *23*

Interface Modeling and Models of Computation *24*
2.1.3.c. Virtual Machine Architectures 24
Types for Real-Time Programs *24*
Separating Reactivity from Schedulability *24*
Implementing Event Scoping *24*
Real-Time Software using Ptolemy-II, Giotto, and the E and S Virtual Machines *25*
2.1.3.d. Components for Embedded Systems 25
Mapping Network Applications to Multiprocessor Embedded Platforms *25*
Prospector: Code Assistant based on Jungloid Mining *25*
2.1.3.e Verification of Embedded Software 25
Model Checking Quantitative Properties of Systems *25*
Run-Time Error Handling *26*
Memory Safety Enforcement: *26*
2.1.4. Experimental Research 26
2.1.4.a. Embedded Control Systems 27
Conflict Detection for Aircraft *27*
2.1.4.b. Embedded Software for National and Homeland Security 27
Soft Walls: Restricting Navigable Airspace *27*
2.1.4.c. Networks of Distributed Sensors 27
VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems *27*
Large Scale Sensor Networks *28*
Programming Models for Sensor Networks *29*
Programming by Sketching for Bitstream Programs *29*
Elder Care *30*
2.1.4.d. Fault-Driven Applications 30
2.1.4.e Design Space Exploration in a Multi-media Subsystem 33
2.2. Project Findings 34
2.3. Project Training and Development 72
2.4. Outreach Activities 72
2.4.1. Curriculum Development for Modern Systems Science (MSS) 72
2.4.2. SUPERB-IT Program 73
Project: Interactive embedded systems showcase *74*
Project: Wireless systems modeling *74*
Project: Actor-oriented construction of interactive 2-D graphics *74*
Project: Secure transport of mobile models and data in distributed applications *75*
Project: Actor-oriented design of smart-home systems *75*
Project: Actor-oriented design of web-integrated string manipulation *75*
Project: Platform Based Reconfigurable Hardware Exploration via Boolean Constraints *75*
Plans for 2004 76
Framework: Hyper *76*
Project: Ordinary Differential Equation Solver *76*
Project: Switched System Simulator *76*
Project: Visualization interfaces *76*
2.4.3. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program 77

Project: Visual Tracking *78*
Project: LEGO Mindstorm Robot Control *80*
Project: TAB Robot Control *81*
Project: Control of Adaptive Structures *81*
3. Publications and Products 81
3.1. Journal Publications 81
3.2. Conference Papers 82
3.3. Books, Reports, and Other One-Time Publications 86
3.4. Dissemination 89
3.5. Other Specific Product 89
4. Contributions 92
4.1. Within Discipline 92
4.1.1. Hybrid Systems Theory 92
4.1.2. Model-Based Design 92
4.1.3. Advanced Tool Architectures 93
4.1.4. Experimental Research 94
4.2. Other Disciplines 95
4.3. Human Resource Development 95
4.4. Research and Education 95
4.5. Beyond Science and Engineering 95

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, ELECTRICAL AND COMPUTER ENGINEERING)

FACULTY INVESTIGATORS:

ALEX AIKEN, (UC BERKELEY, CS)
RUZENA BAJCSY, (UC BERKELEY, EECS)
GAUTAM BISWAS, (VANDERBILT, COMPUTER SCIENCES)
RASTISLAV BODIK, (UC BERKELEY, EECS)
BELLA BOLLOBAS, (UNIVERSITY OF MEMPHIS, MATHEMATICS)
JEROME A. FELDMAN (UC BERKELEY, EECS)
KENNETH FRAMPTON, (VANDERBILT, MECHANICAL ENGINEERING)
J. KARL HEDRICK, (UC BERKELEY, ME)
GABOR KARSAI, (VANDERBILT, ELECTRICAL AND COMPUTER ENGINEERING)
KURT KEUTZER, (UC BERKELEY, EECS)
WAGDY H. MAHMOUD (TENNESSEE TECH. UNIVERSITY)
GEORGE NECULA, (UC BERKELEY, EECS)
SRINI RAMASWAMY (TENNESSEE TECH. UNIVERSITY)
PRAVIN VARAIYA, (UC BERKELEY, EECS)

POST DOCTORAL RESEARCHERS:

MASSIMO FRANCESHETTI (UC BERKELEY)
CHRISTOPH KIRSH (UC BERKELEY)

GRADUATE STUDENTS:

ALESSANDRO ABATE (UC BERKELEY)
AARON AMES (UC BERKELEY)
LUCA CARLONI (UC BERKELEY)
KAI CHEN (VANDEREBILT)
ELAINE CHEONG (UC BERKELEY)
ABHIJIT DAVARE (UC BERKELEY)
DOUG DENSMORE (UC BERKELEY)

ANDREW D. DIXON (VANDERBILT)
MATTHEW J. EMERSON (VANDERBILT)
JOYTI GANDHE (VANDERBILT)
MATTHEW HARREN (UC BERKELEY)
ETHAN JACKSON (VANDERBILT)
FARINAZ KOUSHANFAR (UC BERKELEY)
NARAYANAN KRISHNAN (UC BERKELEY)
ALEXANDER KURZHANSKIY(UC BERKELEY)
GABOR MADL (VANDERBILT)
XIAOJUN LIU (UC BERKELEY)
DAVID P. MANDELIN (UC BERKELEY)
ELEFTERIOUS MATSIKOUDIS (UC BERKELEY)
STEPHEN NEUENDORFFER (UC BERKELEY)
SONGHWAI OH (UC BERKELEY)
WILLIAM PLISHKER (UC BERKELEY)
KAUSHIK RAVINDRAN (UC BERKELEY)
PANNAG SANKETI (UC BERKELEY)
PETER SCHMIDT (VANDERBILT)
TIVADAR SZEMETHY (VANDERBILT)
GUANG YANG (UC BERKELEY)
YANG ZHAO (UC BERKELEY)
HAIYANG ZHENG (UC BERKELEY)
YE ZHOU (UC BERKELEY)

UNDERGRADUATE STUDENTS:
DANIEL BALASUBRAMANIAN
NICKOLIA COOMBS (VANDERBILT)
RACHAEL DENNISON (VANDERBILT)
DAVID GARCIA (VANDERBILT)
SHANTEL HIGGINS (VANDERBILT)
JOHN KILBY (VANDERBILT)
EFOSA OJOMO (VANDERBILT)
MICHAEL RIVERA-JACKSON (VANDERBILT)
BINA SHAH (VANDERBILT)
EDWIN VARGAS (VANDERBILT)
TRIONE VINCENT (VANDERBILT)

TECHNICAL STAFF, PROGRAMMERS:
CHRISTOPHER HYLANDS BROOKS (UC BERKELEY)
NATHAN JEW (UC BERKELEY)

BRADLEY A. KREBS (UC BERKELEY)
MARVIN MOTLEY (UC BERKELEY)
GUNNAR PROPPE (UC BERKELEY)
MARY STEWART (UC BERKELEY)
BRIAN WILLIAMS (VANDERBILT)

BUSINESS ADMINISTRATORS:
SUSAN B. GARDNER (UC BERKELEY)
ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)

1.2. Partner Organizations

- University of California at Berkeley
- Vanderbilt University
- University of Memphis

1.3. Collaborators

- Albert Benveniste (IRISA INRIA, Rennes)
- 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)
- Kim 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 second Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software". This research activity is primarily organized through a Center at Berkeley CHESS (the Center for Hybrid and Embedded Systems and Software, http://chess.eecs.berkeley.edu), the Vanderbilt ISIS (Institute for Software Integrated Systems, http://www.isis.vanderbilt.edu), and the Department of Mathematical Sciences, (http://msci.memphis.edu) at 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 second year were:

1. NSF Onsite Review, December 3rd, 2003, UC Berkeley. The program and the presentations are available at http://chess.eecs.berkeley.edu/conferences/03/NSFonsiteReview12-03/AgendaNSFReview12-03.htm
2. The annual Chess Review for our industrial partners, advisory board members, and friends of the project. The program and presentations are available at http://chess.eecs.berkeley.edu/conferences/04/05/index.htm
3. A weekly Chess workshop was held at Berkeley. Presentations for the workshop are available at http://chess.eecs.berkeley.edu/workshop.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.

3. We have been developing and evaluating several methods for the computational treatment of hybrid systems. In particular, in the past year we designed and implemented 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.

2.1.1.a. Deep Compositionality

Non-Zero-Sum Games as Compositional System Models

The components of a complex system can be viewed as players in a game with different objectives. Each component attempts to satisfy its own specification, but the specifications of different components may be neither identical nor complementary; so the right formalization is that of a non-zero-sum games. Classically, Nash equilibria capture the notion of rationality for such games. We argue, however, that for achieving compositionality in system design, a special kind of Nash equilibrium is appropriate. In the past year, we defined and studied these so-called "secure" equilibria. This work is reported in [34].

Hybrid Systems Modeling Tools: a Survey

In a collaborative project with the European Columbus project, we evaluated a set of tools, languages and formalisms for the simulation, verification and specification of hybrid systems. In this survey we evaluated the following languages and tools: 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 are reported in [32].

Composing Quantitative Aspects of Systems

We developed a theory for composing robust models of systems, where each trace does not have a boolean value ("possible" or "impossible") but a real value, which indicates the distance to a possible trace. Then, composition is not the intersection of traces, but an operation on distances. A preliminary report has been submitted for publication [40].

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 last year, we developed algorithms for computing the real value of discounted properties expressed in temporal logic over state transition systems. This work is reported in [38].

Affine Hybrid Systems

Affine hybrid systems are hybrid systems where the discrete domains are affine sets, and the transition maps between discrete domains are affine transformations. This definition differs from other definitions of hybrid systems that have been proposed, but the underlying ideas involved in the definition of affine hybrid systems have been seen in the literature. We give a formal framework to these ideas.

Affine hybrid systems are simple, and it is this simplicity that allows us to say some useful things about them. The structure of affine hybrid systems contains a wealth of intrinsic information. Affine sets can be described in terms of matrix inequalities, and affine transformations are characterized by elements of SE(n). In this paper, we use the geometric information intrinsic in affine hybrid systems to develop the idea of *spatial equivalence* between an affine hybrid system **H** and an affine hybrid system **G.**

In the literature on hybrid systems, it typically is assumed that all of the transition maps of a hybrid system are the identity; all switched systems are essentially hybrid systems where the transition maps are the identity. This assumption is very restrictive; some of the simplest hybrid systems do not satisfy this assumption, e.g., the hybrid system modeling the torus (for a visual interpretation of this hybrid system, see Figure 1). For this reason, it is desirable to find a way to bridge the gap between hybrid systems where all the transition maps are the identity and hybrid systems where this is not the case. For example, the results obtained in [9] assume that all of the transition maps are the identity.

Given an affine hybrid system **H**, we would like to construct an affine hybrid system $\mathbf{H}_{id}$ such that all of the transition maps are the identity. We also would like this affine hybrid system $\mathbf{H}_{id}$ to be as similar to **H** as possible. In what way should these two affine hybrid systems be considered similar? Spatial equivalence is introduced as a way to consider an affine hybrid system **H** as similar to an affine hybrid system **G**. Spatial equivalence can be thought of in an intuitive manner (see Figure 1 for a visual interpretation). Replace each edge of **H** by a sequence of edges and domains with vector fields such that if we "start" at the source of the edge, the target of the edge will be reached in some time. If the affine hybrid system obtained by appending these edges, domains and vector fields to **H** is **G**, then **H** is spatially equivalent to **G**. A formal definition of spatial equivalence is given in [1].

An affine hybrid system H is compact if each of its domains is compact. The main theorem of this paper is:

Main Theorem: *Every compact affine hybrid system* **H** *is spatially equivalent to an affine hybrid system* $\mathbf{H}_{id}$ *in which every transition map is the identity. Moreover,* $\mathbf{H}_{id}$ *is computable.*

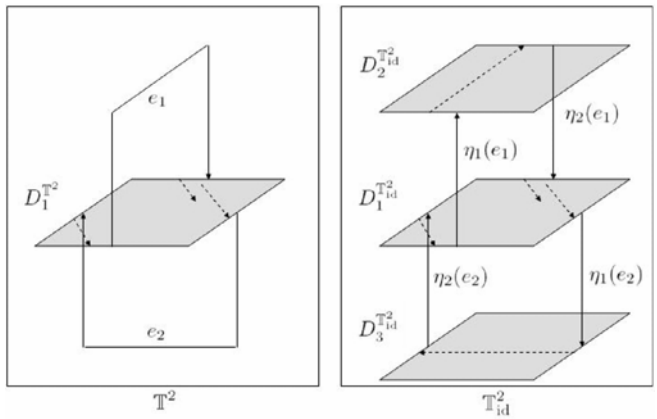

Figure 1. Left: A graphical representation of the hybrid system modeling the two-torus. Right: A graphical representation of the hybrid system that is spatially equivalent to the two-torus but has identity transition maps.

Blowing Up Affine Hybrid Systems

If **H** is an affine hybrid system, in [2] we introduce its blow up, Bl(**H**), which is also an affine hybrid system; for a graphical representation of the blow up, see Figure 2. The primary benefit of considering Bl(**H**) is that it is not Zeno, although its structure suggests many other interesting properties not generally found in affine hybrid systems. In order to demonstrate that Bl(**H**) is in some way equivalent to **H**, $\mathcal{P}$-stability equivalence is introduced. If $\mathbf{O^H}$ is the set of equilibrium points and periodic orbits of **H**, then two affine hybrid systems **H** and **G** are $\mathcal{P}$-stability equivalent if there exists a bijection $\mathbf{Y}: \mathbf{O^H} \rightarrow \mathbf{O^G}$ such that μ in $\mathbf{O^H}$ is $\mathcal{P}$-stable if and only if $\mathbf{Y}(\mu)$ in $\mathbf{O^G}$ is $\mathcal{P}$-stable, where $\mathcal{P}$ is stability in the sense of Lyapunov, asymptotic stability or exponential stability. The main purpose of [9] was to prove the following theorem:

Main Theorem: *The affine hybrid systems* **H** *and* Bl(**H**) *are* $\mathcal{P}$*-stability equivalent, and* Bl(**H**) *is not Zeno.*

The importance of the Main Theorem is that rather than attempting to determine whether an affine hybrid system is Zeno (which currently is not possible), analysis can be carried out on Bl(**H**) where there is no Zeno behavior. Additionally, most analysis on the stability of hybrid systems, or even switched systems, assumes that such systems are not Zeno. Because of the Main Theorem, this assumption automatically holds for Bl(**H**) and Bl(**H**) is $\mathcal{P}$-stability equivalent to **H**, so the assumption is not restrictive. Bl(**H**) displays additional desirable properties that are not found in general affine hybrid systems. Its structure closely resembles a switched system, implying that Bl(**H**) might provide a way to apply the analysis carried out on switched systems to affine hybrid systems; since there are considerably more results for switched systems, this would be an important connection. In the future, these and other properties of Bl(**H**) will be investigated.

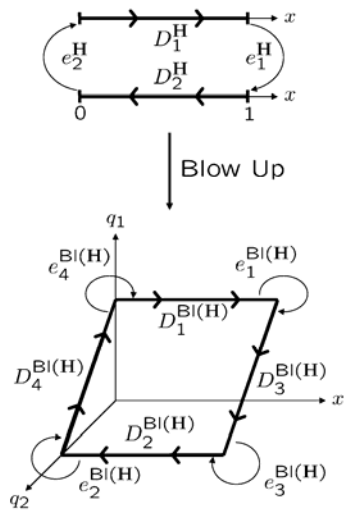

Figure 2. An illustration of the blow up construction; in this case, the blow up construction applied to the thermostat.

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. This work is reported in [36].

Reach Set Calculations using Ellipsoidal Approximations

Linear dynamic systems are considered. We studied the application of the external ellipsoidal approximations of the reach sets to the internal point problem. An algorithm was developed that allows us to calculate control and initial state that bring the system to the given point at the given time. A closed-form solution is obtained, or else, if the target point is unreachable, then the closest reachable point is found.

The ellipsoidal toolbox provides the implementation of the ellipsoidal methods used for the computation of the reach sets. It includes external and internal ellipsoidal approximation algorithms as well as visualization routines. The API allows the user to run the algorithms with given precision.

The work is in progress, and being pursued by Alexander Kurzhanskiy and Pravin Varaiya.

A Deterministic Operational Semantics for Hybrid System Simulations

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 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 are currently developing 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.

2.1.1.d. Stochastic Hybrid Systems

Stochastic hybrid systems are a natural extension of the deterministic counterpart. We have obtained a stability result [1] and we investigated a problem in optimal control.

Application of Stochastic Hybrid Systems to Biological Systems

We applied the stochastic hybrid system to the modeling of genetic regulatory network. This modeling approach can demonstrate the inherent randomness in molecular interactions and protein production in a cell, phenomena that are observed in biological systems but can not be realized adequately using conventional macroscopic modeling techniques.

A stochastic hybrid system approach is applied in [57] to model the genetic network regulating the biosynthesis of an antibiotic called subtilin in Bacillus subtilis. Each B. subtilis cell is modeled as a stochastic hybrid system, whose continuous variables are the concentrations of various proteins inside the cell, and whose discrete variables are the ON/OFF states of random switches for protein production. The dynamics of the continuous variables follow ordinary differential equations with parameters dependent on the discrete variables, namely, a protein is being produced at a higher rate if the corresponding switch in the genetic network is ON. On the other hand, the discrete variables (switches) change values randomly according to Markov chains whose transition probabilities are functions of the concentrations of modulating proteins,

modeling the activations and deactivations of the production of these protein species. While the dynamics of the continuous variables are deterministic, the randomness of the system arises from the probabilistic switching of the discrete variables, which indirectly makes the evolution of the continuous variables random as well.

In our model, the interaction of a B. subtilis cell with the environment is modeled through the input and output of the corresponding stochastic hybrid system: it takes as input the environmental signal such as the nutrient level, and its output is the amount of subtilin released to the environment. Whenever the nutrient level is below a certain threshold, the production of various regulatory proteins will pick up. As a result, the amount of subtilin produced and released to the environment will increase. The released subtilin will then kill competing species in the environment and in turn preserve food supply for the cell. In addition to cell level modeling, we also propose a population level model for a colony of B. subtilis cells. The dynamics of population density follows a logistic equation, which converges from any initial value to an equilibrium population density determined by the available food and the maximal carrying capacity of the medium.

We analyze the above models, both numerically and analytically. It is found that the system dynamics consist of two parts: a slow part that evolves along a certain limit cycle, and a fast part that adds fluctuation around that limit cycle. The variance of the fluctuation depends on the switching rates of the Markov chains modeling the random switches in the genetic network. In particular, we compare the simulation results of our stochastic hybrid system model with those of the deterministic averaged method, a commonly used approach in computational biology. It is found that our approach can better demonstrate the cell density dependent, sigmoid-like switching behavior observed in subtilin production in B. subtilis cells.

2.1.2. Model-Based Design

Model-based design focuses on the formal representation, composition, and manipulation of models during the design process. It addresses system specification, model transformation, synthesis of implementations, model analysis and validation, execution, and design evolution. The semantic frameworks in which these models are applied may be domain-specific, offering embedded system designers methods and syntaxes that are closer to their application domain. 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.

2.1.2.a. Composition of Domain Specific Modeling Languages

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.

Formally, a DSML [82] is a five-tuple of concrete syntax (C), abstract syntax (A), semantic domain (S) and semantic and syntactic mappings (M_S, and M_C):

$$L = < C, A, S, M_S, M_C >$$

The concrete syntax C defines the specific notation used to express models, which may be graphical, textual or mixed. The abstract syntax A defines the concepts, relationships, and integrity constraints available in the language. The semantic domain S is usually defined in some formal, mathematical framework, in terms of which the meaning of the models is explained. The $M_C : A \rightarrow C$ mapping assigns syntactic constructs (graphical, textual or both) to the elements of the abstract syntax. The $M_S: A \rightarrow S$ semantic mapping relates syntactic concepts to those of the semantic domain.

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 [44]. Our work on MIC 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 [66][7][68].

Compositional Metamodeling

The GME-based metamodeling environment provides support for specifying DSML-s via metamodel composition [67]. 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 metamodeling environment for DSML composition.

Integration of Metamodeling with Hybrid Systems

We have developed a metamodel for the abstract syntax of the Hybrid System Interchange Format (HSIF). This work, which was part of our earlier projects, was lately extended with developing a translator between HSIF and Simulink/Stateflow models [5]. This work helped us understanding the role of model transformations for defining semantics of DSML-s via the formal specification of translators [64].

Semantic Foundations for Heterogeneous Systems

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.

This work is reported in [85].

Generalized causality analysis

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.

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.

We have derived a set of theorems that support effective techniques to generate automatically correct-by-construction adaptors between designs formulated using different coordination paradigms [14]. We have applied these concepts to two scenarios that are of particular relevance for the design of embedded systems: the deployment of a synchronous design over a globally-asynchronous locally-synchronous (GALS) architecture and over a loosely time-triggered architecture (LTTA). The idea followed in these applications is to abstract away from the synchronous specifications the constraints among events of different signals due to the synchronous paradigm and, then, to map the unconstrained design into a different architecture characterized by a novel set of intended behavior of the system is retained. This is achieved by relying on a formal notion of semantics-preserving transformations that we developed based on the idea of morphisms over tag sets.

In the original formulation [14], we restricted ourselves to tagged systems in which parallel composition is by intersection, meaning that unifiable events of each component must have identical variable, data, and tag. While this restriction has allowed us to handle GALS and LTTA models of design, it does not cover all cases of interest. For example, causality relations, scheduling constraints, or earliest execution times are not compatible with parallel composition by intersection. Yet, these are all very important aspects to consider when implementing an embedded system. To capture them, we have proposed an extension of tagged systems where the unification rule for tags is itself parameterized and we have shown how the formal results on the preservation of semantics hold also for these cases [21].

More recently [22], we have introduced an algebra of tag structures to define heterogeneous parallel composition formally. Morphisms between tag structures are used to define relationships between heterogeneous models at different levels of abstraction. In particular, they can be used to represent design transformations from tightly-synchronized specifications to loosely-synchronized implementations. This theory has an important application in the problem of ``matching" a specification and an implementation that are heterogeneous.

2.1.2.c. Model Transformation

Meta Generator Technology

We have developed a language and a suite of supporting tools for the formal, high-level, yet executable specification of model transformations (GREAT (Graph Rewriting and Transformations) [63][95]. The language is based on graph transformation technology, where a complex model transformation is specified in terms of sequenced graph rewriting steps consisting of rewriting rules. This high-level specification facilitates not only the compact, formal representation of what a model translator should do, but also allows formal reasoning about the transformations. The semantics of GREAT has been formally defined (in Z, see the JUCS paper from the publication list), and it comes with a set of supporting tools. The tools suite includes an interpreter for GREAT ("Meta-Programmable Transformation Tool"), a code generator for compiling GREAT rules into C++ [100], and a debugger (coupled with the interpreter).

Pattern-Based Model Synthesis

We have introduced patterns in modeling on two levels. First, our meta-model composition technology [67] enables to define abstract metamodeling constructs such *templates* and *hierarchy* as language design patterns and compose those with DSML metamodels [82][81]. Second, using the *template* construct, we are able to build large design spaces centered around domain architectures [82] and automatically synthesize models that satisfy user defined constraints. .

The graph transformation formalism also enables the introduction of reusable idioms and patterns in model transformations [6]. We plan to further explore this opportunity in our future research.

A new direction in model synthesis is aspect weaving [55]. We have experimented with the development and application of model weaving technology for generating complex, integrating models by merging different modeling aspects according to formally represented rules.

2.1.2.d. Real-Time Programming Models

Event-Triggered Programming

In previous work, we had developed a time-triggered language, called Giotto, based on the Logical Execution Time (LET) model. In Giotto, software tasks are invoked and terminated strictly by clock events. This achieves deterministic behavior, which is of paramount importance in safety-critical systems. Last year, we extended the LET model to non-clock events, and called the resulting language xGiotto. This language permits the invocation and termination of tasks by arbitrary events in a way that still maintains determinism. This work is reported in [46].

2.1.3. 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 meta modeling, 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.

Conceptual concurrence between our frameworks is evolving. Whereas previously GME had mastered meta modeling of abstract syntactic properties of actor-oriented models, today it is moving aggressively towards meta modeling of their semantic properties, focusing initially on the 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 meta modeling. 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 meta modeling 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.

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.

Actor-oriented languages, like object-oriented languages, are about modularity of software. We argue that we can adapt for actor-oriented languages many (if not all) of the innovations of OO design, including concepts such as the separation of interface from implementation, strong typing of interfaces, subtyping, classes, inheritance, and aspects. We have realized some preliminary implementations of these mechanisms in Ptolemy II and have published a preliminary report [71].

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.

Metropolis Framework

Features and efficiency are critical factors for simulators, which are still the dominating tools for design validation. During the past year, we improved the Metropolis simulator in several ways:

1. *Add quantity annotation support*. Quantities can be used to model various kinds of performance indices, such as time, and power. They can also be generalized to model scheduling policies. Both of the usages usually appear in architecture models. A simulator's job is to ensure the quantity annotation requests are made at the proper time, and quantity resolution algorithms are executed when necessary to reach a fixed point solution.

2. *Add mapped behavior simulation support*. This capability is essential to evaluate behavior-architecture mappings, which is the key idea to explore design space in platform-based design. With this feature, a user's behavioral model can run simultaneously with a particular architecture under evaluation. Then, performance of the architecture running this behavior can be determined by simulation. During simulation, not only the events in both behavior and architecture models need to be synchronized, but also values need to be passed between. Both requirements present challenges to efficient simulation.

3. *Improve simulation efficiency*. In today's design, both behavior models and architecture models can be huge; combining them only makes the scale problem worse. We applied several techniques to improve the Metropolis simulator's performance. Simulation results show orders of magnitude performance improvement. The optimization techniques include named event reduction, a medium-centric simulation algorithm, mapped behavior handling with equivalent classes, efficient hierarchy traversal and an interleaving-concurrency-based simplification. The following table shows the simulation efficiency improvement for a picture-in-picture application:

(unit: second)	**PiP Yapi**				**PiP TTL**			
	User Time	**System Time**	**Total Time**	**Speed up**	**User Time**	**System Time**	**Total Time**	**Speed up**
previous version	7223.56	52.43	7275.99	1	10214.95	73.72	10288.67	1
improved version	89.26	1.20	90.46	80	187.82	2.47	190.29	54

4. *Parallel simulation on symmetric-multi-processor (SMP) machines*. The current implementation of Metropolis simulator is based on the SystemC simulation kernel, which is an interleaving concurrent single kernel process environment. We detached the Metropolis simulator from this foundation and realized the parallelism with the pthreads library, which gives kernel-level processes (true concurrent) on some SMP machines. The preliminary experiments show sub-linear speed up on SMPs.

2.1.3.b. Interface Theories

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.

The component model also makes it easier to study certain formal properties of the components. For example, we can classify discrete event components as either reactive or proactive. A DE component is reactive if all output events and state changes are triggered by input events and proactive otherwise. From the output and next state functions of a DE component, we can derive the relations among the time stamps of its input and output signals, and use these relations to analyze whether a DE composite component is reactive. Current work includes defining a behavioral type system based on pattern matching as the component programming model.

Interface Modeling and Models of Computation

One research effort on interface modeling was centered around the fundamental principles of interface theories and the evaluation of their pragmatic impact on component-based designs. We have constructed a general experimentation framework within Ptolemy II, and integrated the Checker for Interface Compatibility (Chic) tool into Ptolemy II. We have experimented with the various supported formalisms and evaluated their applicability to the wide spectrum of computational models supported in Ptolemy II. The overall outcome stressed the inherent subjectivity of interface theories, and suggested a more model-of-computation oriented approach in their development.

An extended effort has been made towards the specification of behavioral types at the dynamic interaction level through interface automata. We focused on augmenting the interface automata theory in order to handle explicitly more elaborate forms of concurrency. Based on a variant of interface automata we were able to develop an assume guarantee reasoning for mutual exclusion in open systems. In the process of generalizing our results, we revealed the fundamental limitations of the interface automata theory associated with the inherent computational capacity of finite automata.

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. A report has been submitted for publication [57].

Separating Reactivity from Schedulability

We implemented two virtual machines, one to react to environment events (the E or Embedded machine), and the other to react to CPU events (the S or Scheduling Machine), and their interaction. This architecture allows great flexibility (portability, extensibility, and composibility) in the implementation of reactive software. . A report has been submitted for publication [88].

Implementing Event Scoping

We extended the E (Embedded) Machine, a virtual machine with reactive real-time behavior, to accommodate the event scoping mechanism of xGiotto. . A report has been submitted for publication [54].

Real-Time Software using Ptolemy-II, Giotto, and the E and S Virtual Machines

We have created a software infrastructure for designing hard real-time systems using Ptolemy II, Giotto, and an implementation of the E and S machines on KURT Linux (from University of Kansas). We use the Giotto domain within Ptolemy II, which offers a graphical syntax for Giotto models. Systems with periodic tasks are specified with a timing resolution of milliseconds. The first stage of our implementation takes a graphical model as an input and generates C code, E code, and S code. The second stage is an implementation of the Embedded Machine interpreter, which reads in the E Code and interprets it to release the tasks for execution as per the timing requirements stated. The released tasks can either be assigned to a standard scheduler such as EDF, or the scheduling can be preformed by our implementation of the Scheduling Machine, an interpreter for S code.

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. The results are reported in the paper “Automated Task Allocation on Single Chip, Hardware Multithreaded, Multiprocessor Systems” at WEPA-1 2004 [1]. 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.

Prospector: Code Assistant based on Jungloid Mining

When programming with rich component frameworks based on object-oriented technologies, the programmer often struggles with how to make the provided classes perform the desired functionality. For example, given a task as simple as "open a connection," it may be that an object seemingly unrelated to the task at hand needs to be created, after which it must be passed into a method from a class that, too, appears unrelated to the connection. We call the arcane code that performs a common task a "jungloid", to reflect the complexities the programmer faces when developing the code by navigating through the jungle of the framework's documentation. We have developed an interactive tool called Prospector, which assists the programmer to find such jungloids by mining the source code of the framework and the client code using the framework. The results have not yet been published but they are based on the results in [1]. Prospector is implemented in the Eclipse IDE, and will be available at the end of summer 2004.

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. A report on this work has been submitted for publication [34].

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 [94]. Among the systems that we have debugged with our tool is the Ptolemy software.

Memory Safety Enforcement:

In previous work we have developed Ccured, a static analysis tool for C programs. CCured discovers for each pointer what kind of run-time checks, if any, are necessary in order to ensure memory safe execution. On average, Ccured discovers that 80% of the pointers used in a typical C program do not require any run-time checks. However, CCured has serious difficulties for programs that use libraries whose source is not available. For those libraries, CCured must not only make conservative assumptions about how they manipulate pointers, but must also refrain from changing the run-time representation of pointers and objects they point to. These difficulties arise in all static analysis and instrumentation systems. We have developed a solution based on user-defined polymorphic wrapper functions. These wrappers act both as a proxy for the missing library source code for the purpose of the static analysis, and also as the basis for generating code that changes the representation of pointers at the library boundary, to allow the co-existence of Ccured-controlled pointers and standard backwards-compatible pointers for the library objects. See [54].

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 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

Conflict Detection for Aircraft

Correlation between the wind perturbations to the aircraft positions is largely ignored in the current literature on aircraft conflict detection. We introduce a model of a two-aircraft encounter with a random field term to address this issue. Base on this model, one can effectively estimate the probability of conflict by using a Markov chain approximation scheme. Simulation results show that the correlation between wind perturbations does affect the values of the probability of conflict.

One advantage of the approach proposed in this paper is that, upon completion, one has the probability of conflict not only at the current time, but also at all future time instants, which eliminates the need for re-computation if the flight plans remain unchanged. The approach can also be extended to address some more general cases such as, for example, computing the probability of conflict when the current aircraft positions are uncertain, or estimating the probability of intrusion into a protected area of the airspace with an arbitrary shape.

The relevance to hybrid systems is that we can compute the probability of conflict for a fairly general class of aircraft maneuvers, in particular, the maneuvers that can be modeled as the executions of hybrid systems: aircraft follow linear trajectories until certain way points are reached or certain switching criteria are satisfied, then linear trajectories with new velocities are followed. In addition, in each segment, pilot may perform some feedback control to correct the cross track deviation. Based on our achievement in this paper, we are now extending the method to model realistic aircraft dynamics. This work is described in [59].

2.1.4.b. Embedded Software for National and Homeland Security

Soft Walls: Restricting Navigable Airspace

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 George Pappas of the University of Pennsylvania, we have begun to investigate collision avoidance methods based on computational geometry methods. Interim results are reported in [33].

2.1.4.c. Networks of Distributed Sensors

VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems

Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. VisualSense is a software framework designed to support a component-based construction of such 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, and is reported in [13].

Large Scale Sensor Networks

We looked at the problem of routing in large scale sensor networks, where only local connectivity information is used at each sensor node to decide the next-hop destination. The problem is studied in the context of small-world networks, providing a stochastic model that bridges between the well known 'six degrees of separation' property in social science and network routing protocols. Results are summarized in [50] and have also been presented in different seminars and workshops. Some of this work began at Caltech and some under the DARPA NEST project, but it has been refined and revised under this project and will be carried forward under this project.

Another important issue in sensor networks regards how global connectivity is achieved in the network. Links are inherently unreliable and probabilistic connectivity models based on random connection models have been developed. In [51] we obtain basic percolation results for different random connection models that account for the unreliability of the wireless channel.

The unreliability of the wireless channel is also the motivation for the work in [91], with the difference that in this case the effect on the control feedback loop is explicitly taken into account. The estimation problem is central for control systems (e.g. in pursuit evasion games with autonomous robots) and is based on the automatic refinement of the estimation as new observation data are collected in real time. When some of these data are lost due to the unreliability of the link the performance of such estimator obviously starts degrading. We explicitly characterize such degradation giving analytic bounds and showing a sharp transition to instability as the probability of link failure exceeds a given threshold.

The data collection and distribution problems in sensor networks are studied in [46] using a discrete mathematical model that allows to obtain basic limits on the time delay performance of different distribution and retrieval algorithms.

In [47] and [49] we addressed the basic problem of wireless signal propagation. Characterizing the physics of propagation is one of the key problems in wireless, as fundamental information theory limits strongly depend on the accuracy of the channel model. At the same time, this is a very complex task, due to complexity of the propagating environment that does not allow to solve Maxwell equations explicitly. Hence, often stochastic models based on few tunable parameters that can be analytically treated are employed. We proposed a stochastic model based on the theory of random walks and we obtained expressions for the path loss and power delay profile of a transmitted signal.

Finally in [7] we present results on the throughput capacity of wireless networks where nodes are randomly located on the plane. Our bounds improve on previous results in the literature and also show a connection with the field of percolation theory.

Distributed Control in Smart Structures

The control of Smart Structures presents a unique challenge for distributed control systems. The challenge is unique for two reasons: 1) the physics of Smart Structures are such that all nodes in a distributed sensing/control network will experience strongly connected dynamics and (2) the

bandwidth of Smart Structures tends to be very high requiring high sampling rates and fast, efficient inter-node communications. In the past year we have begun experiments in the distributed control of Smart Structures experimental platforms that were constructed in the previous year. As noted, the primary challenges in these experiments center on (1) the design of distributed control systems that accommodate strongly connected inter-node dynamics and (2) the development of control systems and distributed network infrastructure that permit the relatively high sampling and communication rates needed. We have achieved some success in the distributed control development [Tao and Frampton, 2004]. However, the issue of sufficiently fast inter-node communication remains a bottle neck in performance.

Programming Models for Sensor Networks

We have created galsC [http://galsc.sourceforge.net] [37], a language and compiler designed for use with the TinyGALS programming model, which uses TinyOS as the underlying component model. TinyGALS is a globally asynchronous, locally synchronous model for programming event-driven embedded systems, especially sensor networks. At the local level, software components communicate with each other synchronously via method calls. Components are composed to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors without explicitly passing messages. The TinyGALS programming model is structured such that code for all inter-actor communication, actor triggering mechanisms, and access to guarded global variables can be automatically generated from a high level specification. By raising concurrency concerns above the level of TinyOS components, the TinyGALS programming model allows programmers to focus on the main tasks that the application must execute. Programs developed using this task-oriented model are thread safe and easy to debug.

The original TinyGALS code generation toolset was designed to be compatible with software components written for TinyOS 0.6.1. The new implementation is designed to be compatible with TinyOS 1.x, which uses the nesC programming language. Our new language, galsC, extends the nesC language, which allows for better code generation and static analysis of programs. We have also redesigned the TinyGUYS mechanism to have better scoping. Having a well-structured concurrency model at the application level greatly reduces the risk of concurrency errors, such as deadlock and race conditions. These features are especially important in severely memory-constrained and safety-critical systems.

Programming by Sketching for Bitstream Programs

In programming by sketching, the programmer writes down a sketch of the program she has in mind and the compiler fills in the implementation details omitted in the sketch. This magic works because she also describes the external behavior of the desired program (i.e., the model), which the compiler uses to complete the sketch by deriving an implementation that behaves as desired.

Sketching is a good fit for domains where the semantic gap between the algorithm (model, behavior) and the implementation makes writing an effective compiler uneconomical. One such domain is bitstream programs, such as cryptographic algorithms.

We developed StreamBit, our prototype case study of programming by sketching. StreamBit guarantees correctness by construction, and sketching allows the programmer to rapidly prototype and test even unproven implementation ideas.

In our implementation of the DES cipher, sketching allowed the user to specify high-level optimizations very concisely, achieving performance improvements of 600%, within 25% of the best hand-crafted code. The results are described in [91].

Elder Care

We have been experimenting with three axis accelerometer attached to a person and studying the sensitivity and robustness of these measurements for activities: Change from sitting to standing, Free fall, Walking, Running. We also experimented with two different locations where the accelerometer is attached to the human body: on the chest and on the waist. The data is preliminary but it suggests that the data analysis will have to be customized.

Networks of Distributed Sensors

Distributed acoustic sensing has received a lot of attention in sensor network work. The possible applications include self-localization of sensors, target identification and tracking as well as environmental monitoring. We have developed a distributed acoustic sensing experimental platform based on PC-104 stack hardware (the same used in the Smart Structures control experiments) and begun testing and evaluation. The primary objective is to develop sensing and sensor fusion algorithms that are distributed in nature (i.e. the computations are carried out in a parallel fashion among the sensor nodes) and without the support of a centralized controller or processor. Initial experiments have demonstrated that accuracies comparable to more sophisticated centralized solvers are possible [Amundson, Frampton and Schmidt, 2004]

2.1.4.d. Fault-Driven Applications

Distributed Diagnosis of Complex Physical Systems

The size and complexity of present day systems motivates the need for developing distributed fault diagnosis algorithms. This work extends the TRANSCEND [3,4] qualitative framework for fault diagnosis in continuous dynamic systems, to develop a methodology that partitions the set of possible fault candidates in a physical system to independent sets of faults given a set of measurements. Separate diagnosers that do not interact with each other can be constructed for each set of independent faults while maintaining complete diagnosability of the system. Our approach is to partition the set of faults into subsets in such a way that we can construct independent diagnosers for each subset. Two diagnosers are independent if they do not have to share information in establishing unique diagnosis results that are globally valid. We establish this by ensuring that the two fault subsets corresponding to the two diagnosers do not require the same set of measurements to achieve complete diagnosability. Complete diagnosability is the ability to uniquely isolate every fault candidate in the system given a set of measurements. Although such system decomposition will not always be possible, our approach can be the first step for designing distributed diagnosers. The implication of this approach is that a large

computationally expensive diagnosis task is decomposed into a set of smaller tasks that can be performed independently, thus reducing the overall complexity of online diagnosis.

Definition: Two sets of faults $F_1, F_2, F_1 \cap F_2 = \phi$ are said to be *independent* for diagnosis given measurement set *M* if there exists two sets $M_1, M_2 \subset M$ such that,

- F_1 is diagnosable for the measurement set M_1
- F_2 is diagnosable for the measurement set M_2
- $M_1 \cap M_2 = \phi$

To develop a systematic formulation for this problem we define a fault signature matrix given the

Set of faults, *F* and the set of measurements, *M*. $FSM = \lfloor FS(f_i, m_j) \rfloor_{l \times n}$, where $\lfloor FS(f_i, m_j) \rfloor$ is the fault signature for measurement m_j given fault f_i.

Definition: The *distinguishing measurement set* for fault $f_1 \in F$ is defined by the map

$Dis: F \rightarrow P^2(M)$, where $Dis(f_i) = \{M' \subseteq M \mid f_i \; is \; diagnosable \; given \; M'\}$.

In general, Dis(f_i) will contain multiple measurement subsets. (We assume $Dis(f_i) \neq \phi$). The

$$(\forall p_i, p_j \in P)\left[\bigcup_{f_i \in p_i} Dis(f_i)\right] \cap \left[\bigcup_{f_j \in p_j} Dis(f_j)\right] = \phi$$

partitioning problem is finds a maximal size partition *P* of *F* that satisfies

It is clear that the solution to the partitioning problem is at least as complex as set covering problem, which is NP-complete. We develop a sub-optimal but time constrained practical solution to this problem using domain-specific heuristics based on fault signatures. This method has been successfully applied to develop distributed diagnosers for a number of practical problems. The next step is to extend this algorithm to practical situations where some overlapping measurements have to be considered among the fault sets.

On-line Hierarchical Fault Adaptive Control

This work extends previous work we have done on online approaches for the safety control of a general class of hybrid systems. This year we have focused on a hierarchical online fault-adaptive control approach for complex systems made up of a number of interacting subsystems. To avoid complexity in the models and online analysis, diagnosis and fault-adaptive control is achieved by local units. To maintain overall performance, the problem of resource management for contending concurrent subsystems has to be addressed. We have developed a control structure, where predefined set-point specifications for system operation are used to derive

optimizing utility functions for the subsystem controllers. We apply this approach in situations where a fault occurs in a system, and once the fault is isolated and identified, the controllers use the updated system model to derive new set point specifications and utility functions for the faulty system.

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 following discrete-time form of the state space equations describes the continuous dynamics of this class of hybrid systems:

$$x(k+1) = \Phi(x(k), u(k), q(k))$$
$$q(k+1) = \delta(q(k), x(k))$$

where k is the time index, $x(k) \subseteq \Re^n$ is the sampled form of the continuous state vector at time k, $u(k) \subseteq \Re^m$ is the discrete valued input vector at time k, and $q(k) \in Q$ is the mode (discrete state) at time k. Q is a finite set of discrete states that the system can be in. δ is the (partial) transition relation. We use X and U to denote the state space and the finite input set for the system, respectively. For each mode, $q \in Q$ the function Φ is continuous in X and meets the conditions for existence and uniqueness of solutions for a set of initial states $X_o \subseteq X$. 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 online control approach, the controller explores only a limited forward horizon in the system state space and selects the next event based on the available information. For set-point specification, the selection of the next step is based on a distance map that defines how close the current state is to the desired set point. The distance map can be defined for each state $x \subseteq \Re^n$ as $D(x) = \|x - x_s\|$, where $\|.\|$ is a proper norm for $\Re^n$. In the case of performance specification, the input that minimizes (maximizes) a given multi-attribute 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 have been simple weighted functions of the form $V_i(P_i) = w_i P_i$, where the weights take on values in the interval [−1 1], and represent the importance of the parameter in the overall operation of the system. The supervisory controller uses the system model to predict possible behaviors corresponding to different action sequences for a finite forward time horizon, and then selects the action (i.e., control input) that maximizes the utility function. This process is then repeated for the next time step, and so on.

We have successfully demonstrated the use of this system for a NASA application that involves components of the Advanced Life Support Systems for long-term manned missions.

Safety-Critical Distributed Applications

We developed a design methodology and the supporting tools to address feedback control problems with fault-tolerant requirements (e.g. automotive safety-critical applications).

The flow lets the designer specify independently:
- the algorithmic solution
- the distributed execution platform
- the fault behavior

The three aspects of the design are represented using respectively
- a flavor of synchronous dataflow called fault-tolerant dataflow (FTDF)
- a bipartite graph (channels and electronic control units) with performance annotations
- a relation between failure patterns (subsets of the architecture graph that may fail in a same iteration) and the corresponding subset of the algorithm that must be guaranteed

Based on these three aspects of the specification, an automatic synthesis tool introduces redundancy in the algorithms and schedules the FTDF actors on the distributed architecture, so that the fault behavior is met. In doing so, the scheduling tool aims at minimizing latency (the critical path from sensors to actuators). Finally some verification tools analyze the solution to extract timing and to verify replica determinism and fault behavior.

The basic flow is presented in the paper "Fault-Tolerant Deployment of Embedded Software for Cost-Sensitive Real-Time Feedback-Control Applications" [85].

We tested the entire flow on a drive-by-wire example from BMW and a steer-by-wire example from General Motors. The experiments support the value of the methodology and suggested directions for further improvement.

2.1.4.e Design Space Exploration in a Multi-media Subsystem

We have pushed an industrial design example, the Picture-in-Picture (PiP) subsystem for digital television, through the entire Metropolis flow. Functional modeling, architectural modeling, communication and architecture refinement, mapping, and simulation have been carried out for the PiP design within the Metropolis framework. We have also demonstrated that design space exploration is greatly facilitated by clearly separating various aspects of the design process. We

are currently involved in further exercising and extending the capabilities of Metropolis by evaluating new applications and architectures, and by adding new verification, refinement and optimization capabilities to Metropolis.

2.2. Project Findings

Abstracts for key publications representing project findings during this reporting period, are provided here. These are listed alphabetically by first author. 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] Online Safety Control of a Class of Hybrid Systems.

S. Abdelwahed,, G. Karsai, and G. Biswas,, 41st IEEE Conference on Decision and Control, Las Vegas, NV, pp 1988-1990

> **Abstract:** In this paper we outline a supervisor synthesis procedure for safety control of a class of hybrid systems. The procedure is conducted online based on a limited exploration of the state space. We establish feasibility conditions for online controllability with respect to the safety specifications, and provide an upper limit for the accuracy error of the online controller.

[2] Online Hierarchical Fault-Adaptive Control for Advanced Life Support Systems

S. Abdelwahed, J. Wu, G. Biswas, J. W. Ramirez, and E. J. Manders, International Conference On Environmental Systems, Denver, CO, July 2004.

> **Abstract:** This paper discusses a hierarchical online fault-adaptive control approach for Advanced Life Support (ALS) Systems. ALS systems contain a number of complex interacting sub-systems. To avoid complexity in the models and online analysis, diagnosis and fault-adaptive control is achieved by local units. To maintain overall performance, the problem of resource management for contending concurrent subsystems has to be addressed. We implement a control structure, where predefined set-point specifications for system operation are used to derived optimizing utility functions for the subsystem controllers. We apply this approach in situations where a fault occurs in a system model to derive new set point specifications and utility functions for the faulty system.

[3] A stability criterion for Stochastic Hybrid Systems

Alessandro Abate, Ling Shi, Slobodan Simic, Shankar Sastry, accepted to MTNS 04.

> **Abstract**: This paper investigates the notion of stability for Stochastic Hybrid Systems. The uncertainty is introduced in the discrete jumps between the domains, as if we had an underlying Markov Chain. The jumps happen every fixed time T; moreover, a result is given for the case of probabilistic dwelling times inside each domain. Unlike the more classical Hybrid Systems setting, the guards here are time-related, rather than space-related. We shall focus on vector fields describing input-less dynamical systems. Clearly, the uncertainty intrinsic to the model forces to introduce a fairly new definition of

stability, which can be related to the classical Lyapunov one though. Proofs and simulations for our results are provided, as well as a motivational example from finance.

[4] Robust Model Predictive Control through Adjustable Robust variables: an application to Path Planning

Alessandro Abate, Laurent El Ghaoui, submitted to CDC04.

Abstract: Robustness in Model Predictive Control (MPC) is the main focus of this work. After a definition of the conceptual framework and of the problem's setting, we will analyze how a technique developed for studying robustness in Convex Optimization can be applied to address the problem of robustness in the MPC problem. Therefore, exploiting this relationship between Control and Optimization, we will tackle robustness issues for the first setting through methods developed in the second framework. Proofs for our results are included. As an application of this Robust MPC result, we shall consider a Path Planning problem and discuss some simulations thereabout.

[5] Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT

Aditya Agrawal, Gyula Simon, Gabor Karsai, Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT) 2004. To appear in Electronic Notes on Theoretical Computer Science, Elsevier

Abstract: Embedded systems are often modeled using Matlab's Simulink and Stateflow (MSS), to simulate plant and controller behavior but these models lack support for formal verification. On the other hand verification techniques and tools do exist for models based on the notion of Hybrid Automata (HA) but there are no tools that can convert Simulink/Stateflow models into their semantically equivalent Hybrid Automata models. This paper describes a translation algorithm that converts a well-defined subset of the MSS modeling language into the equivalent hybrid automata. The translation has been specified and implemented using a metamodel-based graph transformation tool. The translation process allows semantic interoperability between the industry-standard MSS tools and the new verification tools developed in the research community.

[6] Reusable Idioms and Patterns in Graph Transformation Languages

A. Agrawal, Zs. Kalmar, A. Narayanan, F. Shi, A. Vizhanyo, G. Karsai, submitted to the 2004 International Conference on Graph Transformations

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

[7] An End-to-End Domain-Driven Software Development Framework

Agrawal A., Karsai G., Ledeczi A., 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Domain-Driven Development Track, Anaheim, CA, October, 2003

Abstract: This paper presents a comprehensive, domain-driven framework for software development. It consists of a meta-programmable domain-specific modeling environment and a model transformation generator toolset based on graph transformations. The framework allows the creation of custom, domain-oriented programming environments that support end-user programmability. In addition, the framework could be considered an early, end-to-end implementation of the concepts advocated by the OMG's Model Driven Architecture initiative.

[8] Affine Hybrid Systems

A. D. Ames and S. Sastry, in Hybrid Systems: Computation and Control, LNCS Vol. 2993, pg. 16-31, Springer-Verlag, 2004.

Abstract: Affine hybrid systems are hybrid systems in which the discrete domains are affine sets and the transition maps between discrete domains are affine transformations. The simple structure of these systems results in interesting geometric properties; one of these is the notion of *spatial equivalence*. In this paper, a formal framework for describing affine hybrid systems is introduced. As an application, it is proven that every compact hybrid system **H** is spatially equivalent to a hybrid system $\mathbf{H}_{id}$ in which all the transition maps are the identity. An explicit and computable construction for $\mathbf{H}_{id}$ is given.

[9] Blowing Up Affine Hybrid Systems

A. D. Ames and S. Sastry, Submitted to CDC 2004.

Abstract: In this paper we construct the "blow up" of an affine hybrid system **H**, i.e., a new affine hybrid system Bl(**H**) in which **H** is embedded, that does not exhibit Zeno behavior. We show the existence of a bijection **Y** between periodic orbits and equilibrium points of **H** and Bl(**H**) that preserves stability; we refer to this property as **Π**-stability equivalence.

[10] Debugging Temporal Specifications with Concept Analysis

Glenn Ammons, David Mandelin, Rastislav Bodik, James Larus, ACM SIGPLAN Conference on Programming Language Design and Implementation, San Diego, CA, June 2003.

Abstract: Program verification tools (such as model checkers and static analyzers) can find many errors in programs. These tools need formal specifications of correct program behavior, but writing a correct specification is difficult, just as writing a correct program is difficult. Thus, just as we need methods for debugging programs, we need methods for debugging specifications. This paper describes a novel method for debugging formal, temporal specifications. Our method exploits the short program execution traces that program verification tools generate from specification violations and that specification miners extract from programs. Manually examining these traces is a straightforward way

to debug a specification, but this method is tedious and error-prone because there may be hundreds or thousands of traces to inspect. Our method uses concept analysis to automatically group the traces into highly similar clusters. By examining clusters instead of individual traces, a person can debug a specification with less work. To test our method, we implemented a tool, Cable, for debugging specifications. We have used Cable to debug specifications produced by Strauss, our specification miner. We found that using Cable to debug these specifications requires, on average, less than one third as many user decisions as debugging by examining all traces requires. In one case, using Cable required only 28 decisions, while debugging by examining all traces required 224.

[11] A Decentralized Approach to Sound Source Localization with Sensor Networks

I. Amundson, P. Schmidt and K.D. Frampton, Accepted by the 2004 ASME International Mechanical Engineering Conference and Exposition, Anaheim CA, November 2004.

Abstract: A sound source localization system has been developed based on a decentralized sensor network. Decentralization permits all nodes in a network to handle their own processing and decision-making, and as a result, reduce network congestion and the need for a centralized processor. The system consists of an array of battery operated COTS Ethernet-ready embedded systems with an attached microphone circuit. The localization solution requires groups of at least four nodes to be active within the array to return an acceptable two-dimensional result. Sensor nodes, positioned randomly over a ten square meter area, recorded detection times of impulsive sources with microsecond resolution. In order to achieve a scalable system, nodes were organized in groups of from 4 to 10 nodes. Grouping was determined by the selecting the nodes farthest apart from each other. A designated leader of each group analyzed the sound source arrival times and calculated the sound source location based on time-differences of arrival. Experimental results show that this approach to sound source localization can achieve accuracies of about 30 cm . Perhaps more importantly though, it is accomplished in a decentralized manner which can lead to a more flexible, scalable distributed sensor network.

[12] Modeling of Sensor Nets in Ptolemy II

Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, In Proc. of Information Processing in Sensor Networks, (IPSN), April 26-27, 2004, Berkeley, CA, USA.

Abstract: This paper describes a modeling and simulation framework called VisualSense for wireless sensor networks that builds on and leverages Ptolemy II. This framework supports actor-oriented definition of sensor 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.

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

Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, Technical Memorandum UCB/ERL M04/08, University of California, Berkeley, CA 94720, USA, April 23, 2004.

Abstract: This paper describes a modeling and simulation framework called VisualSense for wireless sensor networks that builds on and leverages Ptolemy II. This framework supports actor-oriented definition of sensor 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.

[14] Continuum Percolation with Steps in an Annulus

P. Balister, B. Bollobás and M. Walters, to appear in Annnals of Applied Probability (in 2004)

Abstract: Let A be the annulus in the plane centered at the origin with inner and outer radii $r(1-\varepsilon)$ and r respectively. Place points $x(i)$ in the plane according to a Poisson process with intensity and let $G(A)$ be the random graph with vertex set $x(1)$, $x(2)$, ... , and edges $x(i)x(j)$ whenever $x(i)$-$x(j)$ is in A. Trivially, if the area of A is large then $G(A)$ almost surely has an infinite component. Moreover, if we fix a positive ε, increase r and let $n(c)$ be the critical area of A when this infinite component appears, then $n(c)$ tends to 1 as ε tends to 0. This is in contrast to the case of a `square' annulus where we show that $n(c)$ is bounded away from 1. The result for the circular annulus has also been proved independently by Franceschetti et al.

[15] Critical Probabilities in Continuum Percolation

P. Balister, B. Bollobás and M. Walters, submitted

Abstract: In this paper we consider some continuous percolation questions. The general question we shall consider is the following. Consider a Poisson process of density λ in the plane and let A be a symmetric body. We join two points a,b of Λ if b is in a+A(since A is symmetric the relationship is symmetric). We wish to know for what values of λ percolation occurs. Standard results show that there is a critical density $\lambda(c)$ such that percolation occurs if $\lambda > \lambda(c)$ and does not occur if $\lambda < \lambda(c)$. We prove some bounds on λ(c) for the square and the disc, the two most frequently studied models.

The general method we use is the following. We reduce the continuous model to a bond percolation on the square lattice. The bonds will not be independent in this model but there will be sufficient independence to enable us to prove that percolation occurs if the probability that a bond occurs is high enough. This reduces the problem to that of evaluating a very large but finite numerical integral. This is impractical to evaluate rigorously so we use Monte-Carlo methods.

These models have been widely studied since their introduction by Gilbert in 1961. Various rigorous bounds have been proved but they are very weak (upper and lower bounds are a factor of four apart). Simulation methods have been used extensively; these are significantly different from our results in that they model on the situation on a finite grid and assume that this implies bounds for the infinite grid. Indeed it should be noted that more recent results are often not within the error terms given by earlier papers. We prove that for the disc the critical area is between 4.505 and 4.512 with confidence 99.99, and for the square it is between 4.392 and 4.398 with confidence 99.99.

[16] Random transceiver networks

P. Balister, B. Bollobás and M. Walters, submitted

Abstract: In the paper we consider randomly scattered radio transceivers in*d* dimensions (in practical applications, in the plane or three dimensional space) each of which can transmit signals to all transceivers in a given randomly chosen region about itself. If a signal is retransmitted by every transceiver that receives it, under what circumstances will a signal propagate to a large distance from its starting point. Put more formally, place points $x(1)$, $x(2)$, ... in space according to a Poisson process with intensity 1. Then, independently for each point $x(i)$, choose a bounded region $A(i)$ from some fixed distribution and let G be the random directed graph with vertex set $x(i)$ and edges $x(i)x(j)$ whenever $x(j)$ is in $x(i)+A(i)$. In the paper we show that for any positive eta, if the regions $x(i)+A(i)$ do not overlap too much (in a sense that we shall make precise), then G has an infinite directed path provided the expected number of transceivers that can receive a signal directly from $x(i)$ is at least $1+\eta$. One example where these conditions hold, and we obtain percolation, is in dimension d with $A(i)$ a hypersphere of volume $1+\eta$, where η tends to zero as d tends to infinity. Another example is in two dimensions, where $A(i)$ are randomly oriented sectors of a disk of a given angle and area $1+\eta$.

[17] Fast Transmission in Ad Hoc Networks

P. Balister, B. Bollobás, M. Haenggi and M. Walters, to appear

Abstract: This paper heavily relies on the previous one, and is aimed at genuine applications.

We are interested in various transmission strategies for sending information over large distances in ad hoc wireless networks. We model our network in the standard way by assuming that we have transmitters randomly distributed in the plane; formally we assume they are distributed as a Poisson process of intensity 1. We wish to transmit information from one point, the `source'*s*, to another point, the `target'*t*, with the following three properties: reliably, i.e., with probability $1-\varepsilon$, quickly, i.e., through few hops, economically, i.e., using little power.

By making use of the (rather heavy) results in the previous paper, we show that using directional transmitters and a certain amount of randomness, we can achieve an essentially best possible result.

[18] Connectivity of Random Geometric Graphs

P. Balister, B. Bollobás, A. Sarkar and M. Walters, submitted

Abstract: Suppose n radio transceivers are scattered at random over a desert. Each radio is able to establish a direct two-way connection with the k radios nearest to it. In addition, messages can be routed via intermediate radios, so that a message can be sent indirectly from radio S to radio T through a series of radios $S(1)$, $S(2)$, $S(3)$, ... , $S(n)=T$, each one having a direct connection to its predecessor. How large does k have to be to ensure that any two radios can communicate (directly or indirectly) with each other?

To model the situation, we construct a random geometric graph $G(n,k)$ by joining each point of a Poisson process in the square to its k nearest neighbors. Recently, Xue and Kumar proved that if k=0.074 log n then the probability that $G(n,k)$ is connected tends to zero as n tends to infinity, while if k=5.1774 log n then the probability that $G(n,k)$ is connected tends to one as n tends to ∞. They conjectured that the threshold for connectivity is k=log n. In this paper we improve these lower and upper bounds to k=0.3043 log n and k=0.5139 log n respectively, disproving this conjecture. We also prove bounds for some generalizations of this problem.

[19] A Jump to the Bell Number for Hereditary Graph Properties

J. Balogh, B. Bollobás and D. Weinreich, to appear in Journal of Combinatorial Theory B (In 2004 or 2005)

Abstract: In the last decade and a half it was discovered that even very general graph properties undergo phase transitions: the number of graphs `jumps' at certain places. To be more specific, a hereditary property of graphs is an infinite class P of isomorphism classes of graphs closed under the deletion of vertices. Writing $P(n)$ for the set of graphs in P with vertex set 1, 2, ... , n, and f(n) for the number of graphs in $P(n)$, the question is the behavior of this function $f(n)$. Several such jumps have been proved, and the present paper shows that there is a jump in a hitherto unexpected range: we show that the jump from speeds somewhat below the nth power of n to the penultimate range is in fact clean, and provide a sharp lower bound for hereditary properties in this range. It turns out that the jump is up to the nth Bell number, the number of partitions of a set with n distinguishable elements.

[20] Heterogeneous Reactive Systems Modeling and Correct-by-Construction Deployment

A. Benveniste, L.P. Carloni, P. Caspi, and A.L. Sangiovanni-Vincentelli, Proceedings of the Third International Conference on Embedded Software (EMSOFT), LNCS 2855, Springer-Verlag, 2003.

Abstract: We propose a mathematical framework to deal with the composition of heterogeneous reactive systems. Our theory allows theorems from which design techniques can be derived. We illustrate this by two cases: the deployment of synchronous designs over GALS architectures, and the deployment of synchronous designs over the so-called Loosely Time-Triggered Architectures.

[21] Causality and Scheduling Constraints in Heterogeneous Reactive Systems Modeling

A. Benveniste, B. Caillaud, L. P. Carloni, P. Caspi, and A. L. Sangiovanni-Vincentelli, To appear in the LNCS proceedings of FMCO 2003.

Abstract: Recently we proposed a mathematical framework offering diverse models of computation and a formal foundation for correct-by-construction deployment of synchronous designs over distributed architecture (such as GALS or LTTA). In this paper, we extend our framework to model explicitly causality relations and scheduling constraints. We show how the formal results on the preservation of semantics hold also for these cases and we discuss the overall contribution in the context of previous work on desynchronization.

[22] Heterogeneous Reactive Systems Modeling: Capturing Causality and the Correctness of Loosely Time-Triggered Architectures

A. Benveniste, B. Caillaud, L.P. Carloni, P. Caspi, and A.L. Sangiovanni-Vincentelli, Submitted to the Fourth International Conference on Embedded Software (EMSOFT 2004)

Abstract: We present an extension of a mathematical framework proposed by the authors to deal with the composition of heterogeneous reactive systems. Our extended framework encompasses diverse models of computation and communication such as synchronous, asynchronous, causality-based partial orders, earliest execution times, and more. Models of computation and communication can be combined (e.g., synchrony with causality and/or time). Our theory allows theorems, from which design techniques for correct-by-construction deployment of abstract specifications can be derived. We illustrate our theory by providing a complete formal support for correct-by-construction deployment over an LTTA medium.

[23] Degree distribution of the FKP network model

N. Berger, B. Bollobás, C. Borgs, J. CHayes and O. Riordan, to appear

Abstract: Recently, Fabrikant, Koutsoupias and Papadimitriou introduced a natural and beautifully simple model of network growth involving a trade-off between geometric and network objectives, with relative strength characterized by a single parameter which scales as a power of the number of nodes. In addition to giving experimental results, they proved a power-law lower bound on part of the degree sequence, for a wide range of scalings of the parameter. Here we prove that, despite the FKP results, the overall degree distribution is very far from satisfying a power law. In particular, we establish that for almost all scalings of the parameter, either all but a vanishingly small fraction of the nodes have degree 1, or there is exponential decay of node degrees.

[24] A robust method for hybrid diagnosis of complex systems

G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, G. Karsai, 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS), Washington, D.C., pp. 1125-1130

Abstract: The model-based diagnosis community has developed a variety of qualitative reasoning techniques for fault isolation in dynamic systems. However, most of the

emphasis in this community has been on the fault isolation algorithms, and very little attention has been paid to the problem of robust online detection and symbol generation that are essential components of a complete diagnostic solution. In this paper, we develop a diagnosis scheme that combines fault detection with a combined qualitative and quantitative fault isolation scheme for robust and accurate diagnosis in complex hybrid systems. We focus on fault detection, symbol generation, and parameter estimation, and illustrate the effectiveness of this algorithm by running experiments on the fuel transfer system of fighter aircraft.

[25] Toward Distributed Diagnosis of Complex Physical Systems

G. Biswas, S. Abdelwahed, X. Koutsoukos, J. Gandhe and E. Manders, in review, 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas, December 2004

Abstract: The complexity of present day systems motivates the need for distributed fault diagnosis and isolation. This paper presents the theory behind coupling within a physical system and uses the discriminatory power of fault signatures in a qualitative fault diagnosis framework to define tight coupling within a physical system. This notion of tight coupling is used as a basis for dividing up systems into sub-systems each of which is treated as a stand-alone unit for purposes of fault diagnosis.

[26] Robustness and vulnerability of scale-free random graphs

B. Bollobás and O. Riordan, has appeared as the first paper in the first issue of Internet Mathematics, 2004, 1—31

Abstract: Recently many new 'scale-free' random graph models have been introduced, motivated by the power-law degree sequences observed in many large-scale real-world networks. Perhaps the best known, the Barabasi-Albert model, has been extensively studied from heuristic and experimental points of view.

In this paper we consider mathematically two basic characteristics of a precise version of this model, the LCD model, namely robustness to random damage, and vulnerability to malicious attack. These characteristics are extremely important whether the network is a collection of sensors, communication centers, or a network of computers.

We show that the LCD graph is much more robust than classical random graphs with the same number of edges, but also more vulnerable to attack. In particular, if vertices of the n-vertex LCD graph are deleted at random, then as long as any positive proportion remains the graph induced on the remaining vertices has a component of order n. In contrast, if the deleted vertices are chosen maliciously, a constant fraction less then 1 can be deleted to destroy all large components. For the Barabasi-Albert model these questions have been studied experimentally and heuristically by several groups.

[27] The phase transition in a uniformly grown random graph has infinite order

B. Bollobás, S. Janson and O. Riordan, to appear in Random Structures and Algorithms (maybe in 2004)

The emergence of a giant component is one of the most frequently studied phenomena in the theory of random graphs. Much of the interest is due to the fact that a giant

component in a finite graph corresponds to an infinite component, or `infinite cluster', in percolation on an infinite graph. In fact, it can be argued that it is more important and more difficult to study detailed properties of the emergence of the giant component than to study the corresponding infinite percolation near the critical probability.

The quintessential example of the emergence of a giant component is in the classical `mean field' random graph model, usually called the Erdős-Renyi model.

Our task in this paper is considerably harder. In the model we shall study the giant component emerges much more slowly, in fact the proportion of vertices in the `giant' component is so small that at the critical probability all the derivatives of it are zero (with respect to the appropriate control parameter): the phase transition has infinite order. This is the first time that infinite order has been proved for a `real-life' random graph model.

[28] Max Cut for Random Graphs with a Planted Partition

B. Bollobás and A. D. Scott, to appear in Combinatorics, Probability and Computing (in 2004)

Abstract: Graph problems such as Max Cut, Max k-Cut and Min Bisection are well-known to be *NP*-hard; indeed even approximating Max Cut or Max k-Cut to within an factor $(1+o(1))$ is NP-hard. For random graphs in $G(n,p)$ on the other hand, it is often easy to find an approximate solution quickly and with high probability.

It is therefore interesting to consider graphs with cuts that are significantly larger than the expected value for graphs of that density. The main model (which we study in this paper), involves choosing a partition in advance, and then adding edges so that, with high probability, the chosen partition will be a maximum cut. More precisely, a random graph G with `planted partition' is obtained by partitioning the vertex set into some k classes, taking edges within each class independently with probability p, and edges between the two classes independently with probability r. Thus we expect the planted partition to be a good cut of G, provided r-p is not too small.

Random graphs with small cuts have been studied by numerous mathematicians and computer scientists, including Bui, Chaudhuri, Leighton and Sipser; Boppana; Jerrum and Sorkin; Carson and Impagliazzo; Feige and Kilian; Condon and Karp; Ben-Or, Shamir and Yakhini. In this paper we address planted problems in which different classes may have different sizes. We give an algorithm that runs in time $O(km + n)$, and with high probability recovers the planted partition provided p-r is at least a certain size (which is much smaller than has been proved so far). The algorithm has several advantages over previous algorithms: it is fast (running in time $O(km+n)$) and comparatively simple, and does not require the number of vertex classes to be specified in advance. The algorithm is also easily adapted to related problems such as partitioning hypergraphs or Boolean matrices with planted partitions.

[29] Ptolemy II Coding Style

Christopher Hylands Brooks and Edward A. Lee, Technical Memorandum UCB/ERL M03/44, University of California at Berkeley, November 24, 2003.

Abstract: Collaborative software projects benefit when participants read code created by other participants. The objective of a coding style is to reduce the fatigue induced by unimportant formatting differences and differences in naming conventions. Although individual programmers will undoubtedly have preferences and habits that differ from the recommendations here, the benefits that flow from following these recommendations far outweigh the inconveniences. Published papers in journals are subject to similar stylistic and layout constraints, so such constraints are not new to the academic community.

Software written by the Ptolemy Project participants follows a variant of this style guide. Although many of these conventions are arbitrary, the resulting consistency makes reading the code much easier, once you get used to the conventions. We recommend that if you extend Ptolemy II in any way, that you follow these conventions. To be included in future versions of Ptolemy II, the code must follow the conventions.

[30] A Formal Modeling Framework for Deploying Synchronous Designs on Distributed Architectures

L.P. Carloni and A.L. Sangiovanni-Vincentelli, First International Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures (FMGALS 2003).

Abstract: Synchronous specifications are appealing in the design of large scale hardware and software systems because of their properties that facilitate verification and synthesis. When the target architecture is a distributed system, implementing a synchronous specification as a synchronous design may be inefficient in terms of both size (memory for software implementations or area for hardware implementations) and performance. A more elaborate implementation style where the basic synchronous paradigm is adapted to distributed architectures by introducing elements of asynchrony is, hence, highly desirable. This approach has to conjugate the desire of maintaining the theoretical properties of synchronous designs with the efficiency of implementations where the constraints imposed by synchrony are relaxed. Two interesting avenues have been recently pursued to achieve this goal:

- Latency-insensitive protocols motivated by hardware implementations, where long paths between the design components may introduce delays that force the overall clock of the system to run too slow in order to maintain synchronous behavior. This approach introduces additional elements in the design to allow the implementation to maintain the throughput that could have been achieved with communication delays of the same order of the clock of the subsystems at the price of additional latency.

- Desynchronization motivated by software implementations, where processes that compose the large scale system are locally implemented synchronously while their communication is implemented in an asynchronous style. This approach allows also to run each of the process at its own speed. By using the Lee and Sangiovanni-Vincentelli (LSV) tagged-signal model as a common framework, we offer a comparative exposition

of these approaches and we show their precise relationship. In doing so, we also provide some insight on the role of signal absence in synchronous, asynchronous, and globally-asynchronous locally-synchronous (GALS) design styles.

[31] HyVisual: A Hybrid System Visual Modeler

A. Cataldo, C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, H. Zheng, Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, CA 94720, July 17, 2003.

Abstract: 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 information.

[32] Modeling Techniques, Programming Languages, and Design Toolsets for Hybrid Systems

Luca Carloni, Maria Domenica DiBenedetto, Alessandro Pinto and Alberto Sangiovanni-Vincentelli, Columbus IST-2001-38314 WPHS.

Abstract: This report is a critical review of existing modeling techniques, programming languages, and design toolsets for hybrid systems. We analyzed industrial and academic tools with the intent of comparing their applicability and the different models used to support the tools. In addition to the review, we also provide comments and recommendations on how to build a standard interchange format and a standard representation language for hybrid systems that will enable better interaction between

groups working on the design of embedded controllers based on hybrid system technology.

[33] Control Algorithms for Soft Walls

Adam Cataldo, Master's Report, Technical Memorandum UCB/ERL M03/42, University of California, Berkeley, CA 94720, January 21, 2004.

Abstract: This master's report documents two of the Soft Walls control algorithms we have tried.

[34] A Natural Extension of Automata

Arindam Chakrabarti, Krishnendu Chatterjee, Thomas A. Henzinger, Orna Kupferman, and Rupak Majumdar, Submitted.

Abstract: Formal methods have been successfully used for the verification of finite-state systems that are defined with respect to a set of boolean propositions. We study "quantitative" verification problems, where every state is labeled by a finite set of quantitative propositions, each taking an integer value, and where the properties are quantitative, rather than boolean. For example, the label at each state may represent power consumption, and the value of a property on a system may be the maximal achievable lifetime (in number of transitions) of a battery with a given initial power. We define quantitative properties using quantitative automata, which maintain a set of integer-valued registers that can be updated and tested. While a traditional automaton on infinite words accepts or rejects each input word, a quantitative automaton maps each input word to an integer. In the nondeterministic case, the result is the maximum value achieved over all possible runs. While a traditional automaton can be interpreted over a system state existentially or universally, a quantitative automaton is interpreted by taking either the maximum or the minimum value over all paths from the state. We show how quantitative automata can specify interesting properties, such as battery lifetime, and we characterize the expressive power of the formalism, in terms of number of registers, determinism vs. nondeterminism, and relation between the linear-time, automaton-based view and a corresponding quantitative branching-time, mu-calculus-based view.

Model checking and game solving for quantitative automata, which correspond to the verification of closed and open systems, are obviously undecidable. However, many interesting quantitative properties can be specified by giving, in addition to a quantitative automaton, also a bound function that turns the model-checking and game-solving problems for the given property over any finite structure into finite-state problems. For example, for all quantitative systems, the battery lifetime is either less than a certain known function of the number of states, initial battery power, and peak consumption, or infinity. We study several kinds of bound functions, e.g., functions that map structures to maximal register values, and analyze the complexity of the resulting model-checking and game-solving algorithms.

[35] Games with Secure Equilibria

Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski, Proceedings of the 19th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press, 2004.

Abstract: 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 often 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 objectives, which include the games that arise in verification, there may be several Nash equilibria, but there is always a unique maximal payoff profile of secure equilibria. We show how this equilibrium can be computed in the case of omega-regular objectives, and we characterize the memory requirements of strategies that achieve the equilibrium.

[36] Quantitative Stochastic Parity Games

Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger, Proceedings of the 15th Annual Symposium on Discrete Algorithms (SODA), SIAM, 2004, pp. 114-123.

Abstract: We study perfect-information stochastic parity games. These are two-player nonterminating games which are played on a graph with turn-based probabilistic transitions. A play results in an infinite path and the conflicting goals of the two players are omega-regular path properties, formalized as parity winning conditions. The qualitative solution of such a game amounts to computing the set of vertices from which a player has a strategy to win with probability 1 (or with positive probability). The quantitative solution amounts to computing the value of the game in every vertex, i.e., the highest probability with which a player can guarantee satisfaction of his own objective in a play that starts from the vertex. For the important special case of one-player stochastic parity games (parity Markov decision processes) we give polynomial-time algorithms both for the qualitative and the quantitative solution. The running time of the qualitative solution is O(d m^1.5) for graphs with m edges and d priorities. The quantitative solution is based on a linear-programming formulation. For the two-player case, we establish the existence of optimal pure memoryless strategies. This has several important ramifications. First, it implies that the values of the games are rational. This is in contrast to the concurrent stochastic parity games of de Alfaro et al.; there, values are in general algebraic numbers, optimal strategies do not exist, and epsilon-optimal strategies have to be mixed and with infinite memory. Second, the existence of optimal pure memoryless strategies together with the polynomial-time solution for one-player case implies that the quantitative two-player stochastic parity game problem is in NP and co-NP. This generalizes a result of Condon for stochastic games with reachability objectives. It also constitutes an exponential improvement over the best previous algorithm, which is based

on a doubly exponential procedure of de Alfaro and Majumdar for concurrent stochastic parity games and provides only epsilon-approximations of the values.

[37] galsC: A Language for Event-Driven Embedded Systems

Elaine Cheong and Jie Liu, Technical Memorandum UCB/ERL M04/7, University of California, Berkeley, CA 94720, USA, 20 April 2004.

Abstract: We introduce galsC, a language designed for programming event-driven embedded systems such as sensor networks. galsC implements the TinyGALS programming model. At the local level, software components are linked via synchronous method calls to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors via parameters without explicitly passing messages. The galsC compiler extends the nesC compiler, which allows for better type checking and code generation. In galsC programs, all inter-actor communication, actor triggering mechanisms, and access to guarded global variables are automatically generated by the compiler. Having a well-structured concurrency model at the application level greatly reduces the risk of concurrency errors, such as deadlock and race conditions. The galsC language is implemented on the Berkeley motes and is compatible with the TinyOS/nesC component library. We use a multi-hop wireless sensor network as an example to illustrate the effectiveness of the language.

[38] The Best of Both Worlds: The Efficient Asynchronous Implementation of Synchronous Specifications

Abhijit Davare, Kelvin Lwin, Alex Kondratyev, Alberto Sangiovanni-Vincentelli, ACM/IEEE Design Automation Conference, 2004, San Diego, CA. (To appear)

Abstract: The desynchronization approach combines a traditional synchronous specification style with a robust asynchronous implementation model. The main contribution of this paper is the description of two optimizations that decrease the overhead of desynchronization. First, we investigate the use of clustering to vary the granularity of desynchronization. Second, by applying temporal analysis on a formal execution model of the desynchronized design, we uncover significant amounts of timing slack. These methods are successfully applied to industrial RTL designs.

[39] Model Checking Discounted Temporal Properties

Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Marielle Stoelinga, Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Lecture Notes in Computer Science 2988, Springer-Verlag, 2004, pp. 77-92.

Abstract: 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.

[40] Approximate Composition

Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Marielle Stoelinga, Submitted.

Abstract: We propose a compositional quantitative semantics for transition systems whose states are labeled by real numbers. The semantics replaces the notion of language with a notion of distance: rather than defining which traces (or trees) are accepted by a component, the semantics specifies, for each trace (or tree), a real number that indicates how far the trace is from acceptance. The move from languages to distances leads to an approximate approach to composition, where it is not necessary for input and output values to match exactly, allowing for imprecision in the numerical characterization of components. Our quantitative semantics supports also a quantitative notion of refinement, and we show how the approach admits powerful rules for approximate compositional reasoning and algorithms for quantitative refinement checking.

[41] The Semantics and Execution of a Synchronous Block-Diagram Language

Stephen A. Edwards and Edward A. Lee, Science of Computer Programming, Vol. 48, no. 1, July 2003.

Abstract: We present a new block diagram language for describing synchronous software. It coordinates the execution of synchronous, concurrent software modules, allowing real-time systems to be assembled from precompiled blocks specified in other languages. The semantics we present, based on fixed points, is deterministic even in the presence of instantaneous feedback. The execution policy develops a static schedule - a fixed order in which to execute the blocks that makes the system execution predictable.

We present exact and heuristic algorithms for finding schedules that minimize system execution time, and show that good schedules can be found quickly. The scheduling algorithms are applicable to other problems where large systems of equations need to be solved.

[42] Congestion Control and Fairness for Many-to-One Routing in Sensor Networks

Cheng Tien Ee and R.Bajcsy, to be presented as work in progress at the real time and embedded technology symposium, August, 2004, Toronto, Canada.

Abstract: In this paper we propose a distributed and scalable algorithm that eliminates congestion within a sensor network.

[43] CAL Language Report: Specification of the CAL actor language

Johan Eker and Jorn W. Janneck, Technical Memorandum No. UCB/ERL M03/48, University of California, Berkeley, CA, 94720, USA, December 1, 2003.

Abstract: This report describes CAL, an actor language created as a part of the Ptolemy II project at the UC Berkeley. It is intended primarily as a repository for technical information on the language and its implementation and contains very little introductory material. After a short motivation, we will outline the goals and the guiding principles of the language design. We will also give a short outline of the actor model, and the context that the actors written in CAL are embedded into, describing the kinds of assumptions an actor may and may not, in general, make about it.

[44] MIC, MDA and MOF

Matthew J. Emerson, Janos Sztipanovits and Ted Bapty, IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS 2004

Abstract: The goal of this paper is to describe our latest work on changing the MIC metamodeling environment from UML/OCL to MOF. The work gave us opportunity to evaluate MOF as a metamodeling language, particularly in terms of its support for DSML composition. Our implementation of the MOF-based metamodeling environment (GME-MOF) used the metaprogammable Graphical Model Editor (GME), a core tool of the MIC technology. We believe that this implementation serves also as an example for the power of formally well defined metamodeling and metamodel based model transformation approaches. First, we will provide a short summary of the formal specification of DSML-s. This summary will be followed by an overview and evaluation of MOF as a metamodeling language. The last section of the paper describes the implementation of GME-MOF using metamodeling and model transformations.

[45] Using Interaction Costs for Microarchitectural Bottleneck Analysis

Brian Fields, Rastislav Bodik, Mark D. Hill, Chris J. Newburn, The 36th Annual IEEE/ACM International Symposium on Microarchitecture, San Diego, CA, December 2003.

Abstract: Attacking bottlenecks in modern processors is difficult because many microarchitectural events overlap with each other. This parallelism makes it difficult to both (a) assign a cost to an event (e.g., to one of two overlapping cache misses) and (b) assign blame for each cycle (e.g., for a cycle where many, overlapping resources are active). This paper introduces a new model for understanding event costs to facilitate processor design and optimization. First, we observe that everything in a machine (instructions, hardware structures, events) can interact in only one of two ways (in parallel or serially). We quantify these interactions by defining interaction cost, which

can be zero (independent, no interaction), positive (parallel), or negative (serial). Second, we illustrate the value of using interaction costs in processor design and optimization. Finally, we propose performance-monitoring hardware for measuring interaction costs that is suitable for modern processors.

[46] Lower Bounds On Data Collection Times In Sensory Networks

C. Florens, M. Franceschetti, and R. J. Mc Eliece, IEEE Journal on Selected Areas in Communication 1(11), November 2004, in press.

Abstract: Special issue on fundamental performance limits in sensor networks. We study the data collection and data distribution problems in sensornetworks, using a simple discrete mathematical model.

[47] Acoustic Self-Localization in a Distributed Sensor Network

K.D. Frampton, submitted to IEEE Sensors Journal, October 2003

Abstract: The purpose of this work is to present a technique for determining the locations of nodes in a distributed sensor network. This technique is based on the Time Difference of Arrival (TDOA) of acoustic signals. In this scheme, several sound sources of known locations transmit while each node in the sensor network records the wave front time-of-arrival. Data from the nodes are transmitted to a central processor and the nonlinear TDOA equations are solved. Computational simulation results are presented in order to quantify the solution behavior and its sensitivity to likely error sources. Experimental self-localization results are also presented in order to demonstrate the potential for this approach in solving the challenging self-localization problem.

[48] Stochastic Rays Pulse Propagation

M. Franceschetti, IEEE Trans. on Antennas and Propagation. In press 2004.

Abstract: We analytically derive the power delay profile of a cluttered environment using the random walk model of wave propagation proposed in an earlier paper (see below [6]). We compare results with experimental data and with classical EM theory of wave propagation in random media.

[49] A Random Walk Model Of Wave Propagation

M. Franceschetti, J. Bruck, and L. Schulman, IEEE Trans. on Antennas and Propagation, 52(5), May 2004

Abstract: In this paper we show that a reasonably simple description of the propagation loss in small urban cells can be obtained with a simple "wandering photon" model based on the theory of random walks, and accounting for only two parameters: the amount of clutter, and the amount of absorption in the environment. Obtained analytic results are compared with experimental data.

[50] Navigation In Small World Networks, A Scale-free Continuum Model

M. Franceschetti, R. Meester, Preprint. Submitted 2004.

Abstract: In this paper, we depart from the common practice to use probabilistic combinatorial models to explain the small world phenomenon, and we study this phenomenon in a more natural continuum setting. Our focus is on routing with only local information at each node.

[51] Continuum Percolation With Unreliable And Spread Out Connections

M. Franceschetti L. Booth, J. Bruck, M.Cook and R. Meester, Preprint. Submitted 2004.

Abstract: In this paper we show how unreliable and spread out connections help achieving connectivity in a stochastic network.

[52] A Random Walk Model Of Wave Propagation

M. Franceschetti, J. Bruck, and L. Schulman, IEEE Trans. on Antennas and Propagation, 52(5), May 2004

Abstract: In this paper we show that a reasonably simple description of the propagation loss in small urban cells can be obtained with a simple "wandering photon" model based on the theory of random walks, and accounting for only two parameters: the amount of clutter, and the amount of absorption in the environment. Obtained analytic results are compared with experimental data.

[53] Event-Driven Programming With Logical Execution Times

Arkadeb Ghosal, Thomas A. Henzinger, Christoph M. Kirsch, and Marco A.A. Sanvido, Proceedings of the Seventh International Workshop on Hybrid Systems: Computation and Control (HSCC), Lecture Notes in Computer Science 2993, Springer-Verlag, 2004, pp. 357-371.

Abstract: We present a new high-level programming language, called xGiotto, for programming applications with hard real-time constraints. Like its predecessor, xGiotto is based on the LET (logical execution time) assumption: the programmer specifies when the outputs of a task become available, and the compiler checks if the specification can be implemented on a given platform. However, while the predecessor language Giotto was purely time-triggered, xGiotto accommodates also asynchronous events. Indeed, through a mechanism called event scoping, events are the main structuring principle of the new language. The xGiotto compiler and run-time system implement event scoping through a tree-based event filter. The compiler also checks programs for determinism (absence of race conditions) and time safety (schedulability).

[54] xGiotto and the Embedded Virtual Machine

Arkadeb Ghosal, Marco A.A. Sanvido, and Thomas A. Henzinger, Submitted.

Abstract: xGiotto is a domain-specific language for the implementation of embedded software applications with hard real-time constraints. The language is an extension of the original Giotto language. In this paper we present the xGiotto tool chain, consisting of the compiler and a specialized virtual machine, the Embedded Virtual Machine (EVM). The compiler checks for determinism (absence of races) and time safety (schedulability

within logical execution times) and generates code for the EVM. The EVM integrates an event filter (which handles aperiodic, asynchronous events and event scoping) and a modified Embedded Machine. We also extend the expressiveness of xGiotto by introducing an event calculus. The paper concludes with a case study of implementing an automotive engine controller.

[55] Two-level Aspect Weaving to Support Evolution in Model-Driven Software

Jeff Gray, Janos Sztipanovits, Ted Bapty Sandeep Neema, in Aspect-Oriented Programming

Abstract: This book chapter summarizes our work in applying AOSD techniques to domain-specific modeling and program synthesis. The use of *weavers*, which are translators that perform the integration of separated crosscutting concerns, is described at multiple levels of abstraction. Our research on Aspect-Oriented Domain Modeling (AODM) employs the following two-level approach to weaving: At the top-level, weavers are built for domain-specific modeling environments. The concept of applying AOSD techniques to higher levels of abstraction is based on our Constraint-Specification Aspect Weaver (C-SAW). The second level of weaving occurs during model interpretation. Synthesis of source code from models typically proceeds as a mapping from each modeling element to the generation of a set of semantically equivalent source code statements. When a library of components is available, the model interpreter can leverage a larger granularity of reuse by generating configurations of the available components. It is hard, however, to synthesize certain properties described in a model, e.g., those related to quality of service (QoS), due to the closed nature of the components. An aspect-oriented solution can provide the ability to instrument components with features that are specified in the model.

[56] Lightweight Wrappers for Interfacing with Binary code in Ccured

Matthew Harren and George Necula, In Proceedings of the 3rd International Symposium on Software Security (ISSS03), Tokyo, 2003.

Abstract: The wide use of separate compilation and precompiled libraries among programmers poses a challenge to source-code based security and analysis tools. These tools must have some way of understanding the operation of precompiled libraries without seeing the source code for the library itself. This paper describes the solution we use for CCured: a system of polymorphic wrapper functions. These wrappers check that library functions are invoked on valid arguments, and also maintain the extra runtime metadata and invariants imposed by CCured. We describe the design of these wrappers and our experiences using them, including the case where complex data structures are passed to or from the library.

[57] A Typed Assembly Language for Real-Time Programming

Thomas A. Henzinger and Christoph M. Kirsch, submitted.

Abstract: We present a type system for E code, which is an assembly language that manages the release, interaction, and termination of real-time tasks. E code specifies a deadline for each task, and the type system ensures that the deadlines are path-insensitive. We show that typed E programs allow, for given worst-case execution times of tasks, a

simple schedulability analysis. Moreover, the real-time programming language Giotto can be compiled into typed E code. This shows that typed E code identifies an easily schedulable yet expressive class of real-time programs. We have extended the Giotto compiler to generate typed E code, and enabled the runtime system for E code to perform a type and schedulability check before executing the code.

[58] Modeling Subtilin Production in Bacillus subtilis Using Stochastic Hybrid Systems

Jianghai Hu, Wei Chung Wu, and Shankar Sastry, The 7th International Workshop on Hybrid Systems: Computation and Control, Philadelphia, PA, vol. 2993 of Lecture Notes in Computer Science, pp. 417-431, Springer-Verlag, 2004.

Abstract: The genetic network regulating the biosynthesis of subtilin in Bacillus subtilis is modeled as a stochastic hybrid system. The continuous state of the hybrid system is the concentrations of subtilin and various regulating proteins, whose productions are controlled by switches in the genetic network that are in turn modeled as Markov chains. Some preliminary results are given by both analysis and simulations.

[59] Aircraft Conflict Prediction In Presence Of A Spatially Correlated Wind Field

Jianghai Hu, Maria Prandini, and Shankar Sastry, submitted to IEEE Trans. on Intelligent Transportation Systems, 2004.

Abstract -- In this paper the problem of automated aircraft conflict prediction is studied for two-aircraft midair encounters. A model is introduced to predict the aircraft positions along some look-ahead time horizon, during which each aircraft is trying to follow a prescribed flight plan despite the presence of additive wind perturbations to its velocity. A spatial correlation structure is assumed for the wind perturbations such that the closer the two aircraft the stronger the correlation between the perturbations to their velocities. Using this model, a method is introduced to evaluate the criticality of the encounter situation by estimating the probability of conflict, namely, the probability that the two aircraft come closer than a minimum allowed distance at some time instant during the look-ahead time horizon. The proposed method is based on the introduction of a Markov chain approximation of the stochastic processes modeling the aircraft motions. Several generalizations of the proposed approach are also discussed.

[60] Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II)

C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), Technical Memorandum UCB/ERL M03/27, University of California, Berkeley, CA USA 94720, July 16, 2003.

Abstract: 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.

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

C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), Technical Memorandum UCB/ERL M03/28, University of California, Berkeley, CA USA 94720, July 16, 2003.

Abstract: 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.

[62] Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)

C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), Technical Memorandum UCB/ERL M03/29, University of California, Berkeley, CA USA 94720, July 16, 2003.

Abstract: 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).

[63] Graph Transformations in OMG's Model-Driven Architecture

*Gabor Karsai and A. Agrawal, to appear Lecture Notes in Computer Science volume on Applications of Graph Transformation with Industrial Relevance, 2003***.**

Abstract: The Model-Driven Architecture (MDA) vision of the Object Management Group offers a unique opportunity for introducing Graph Transformation (GT) technology to the software industry. The paper proposes a domain-specific refinement of MDA, and describes a practical manifestation of MDA called Model-Integrated Computing (MIC). MIC extends MDA towards domain-specific modeling languages, and it is well supported by various generic tools that include model transformation tools based on graph transformations. The MIC tools are metaprogrammable, i.e. they can be tailored for specific domains using metamodels that include metamodels of transformations. The paper describes the development process and the supporting tools of MIC, and it raises a number of issues for future research on GT in MDA.

[64] On the Use of Graph Transformation in the Formal Specification of Model Interpreters

Karsai, G., Agrawal, A., Shi, F., Sprinkle, J. Journal of Universal Computer Science, Volume 9, Issue 11, 2003

Abstract: Model-based development necessitates the transformation of models between different stages and tools of the design process. These transformations must be precisely, preferably formally, specified, such that end-to-end semantic interoperability is maintained. The paper introduces a graph-transformation-based technique for specifying these model transformations, gives a formal definition for the semantics of the transformation language, describes an implementation of the language, and illustrates its use through an example.

[65] Automotive Software: A Challenge and Opportunity for Model-based Software Development

Gabor Karsai, to appear in LNCS volume on Automotive Software Development Workshop, San Diego, January, 2004.

Abstract: Embedded software development for automotive applications is widely considered as a significant source of innovation and improvements in cars. However, software development processes do not address well the needs of large-scale distributed real-time systems, like the ones automobiles do (or soon will) contain. The paper introduces a vision for the model-based development of embedded software, which is based on the broad-spectrum modeling of the applications in the context of a larger system, formal (and computer-supported) analysis of models, and automatic synthesis of the application(s). The paper also describes some initial steps taken to build the infrastructure for supporting such a process in the form of modeling and model transformation tools. The paper concludes with a list of challenging research problems.

[66] Model-Driven Architecture for Embedded Software: A Synopsis and an Example

G. Karsai, S. Neema, D. Sharp. To appear in Science of Computer Programming (Elsevier) on Model Driven Architecture: Foundations and Applications Model Driven Architecture

Abstract: MDA proposes a new paradigm for software development in general. We claim that MDA could be beneficial for embedded software development, especially if it is extended to address the special needs of embedded systems. The paper consists of two sections: the first is a brief synopsis on how MDA ought to be extended to handle embedded software development, while the second illustrates the concepts in practice using a prototype modeling language and tool chain designed for developing mission computing software.

[67] Composition and Cloning in Modeling and Meta-Modeling

Karsai, G. Maroti, M. Ledeczi, A. Gray, J. Sztipanovits, J. , IEEE Transactions on Control Systems Technology, March 2004, pp 263- 278, Volume 12, Issue: 2.

Abstract: The Generic Modeling Environment (GME) is a configurable tool suite that facilitates the rapid creation of domain-specific model-integrated program synthesis environments. 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. This paper explores the details of these three ideas and their implications.

[68] A Metamodel-Driven MDA Process and its Tools

Karsai G., Agrawal A., Ledeczi A, WISME, UML 2003 Conference, San Francisco, CA, October, 2003

Abstract: A domain-specific refinement of MDA, called DS-MDA is introduced, and a practical manifestation of it called MIC (for Model-Integrated Computing) is described. MIC extends MDA in the direction of domain-specific modeling languages. The MIC tools are metaprogrammable, i.e. are tailored for specific domains using meta-models. Meta-models capture the domain's and the target platform's general properties, as well as the transformation between the two. The paper introduces the tools and process that supports single domains, and proposes an extension towards multi-model processes.

[69] Tool Integration Patterns

Karsai G., Lang A., Neema S, Workshop on Tool Integration in System Development, ESEC/FSE, pp 33-38., Helsinki, Finland, September, 2003.

Abstract: Design tool integration is a highly relevant area of software engineering, which can greatly improve the efficiency of development processes. Design patterns have

been widely recognized as important contributors to the success of software systems. This paper describes and compares two large-grain, architectural design patterns that solve specific design tool integration problems. Both patterns have been implemented and used in real-life engineering processes.

[70] Cache Aware Scheduling for Synchronous Dataflow Programs

Sanjeev Kohli, Master's Report, Technical Memorandum UCB/ERL M04/03, University of California, Berkeley, CA 94720, February 23, 2004.

Abstract: The Synchronous Dataflow (SDF) model of computation [1] is an efficient and popular way to represent signal processing systems. In an SDF model, the amount of data produced and consumed by a data flow actor is specified a priori for each input and output. SDF specifications allow static generation of highly optimized schedules, which may be optimized according to one or more criteria, such as minimum buffer size, maximum throughput, maximum processor utilization, or minimum program memory. In this report, we analyze the effect of cache architecture on the execution time of an SDF schedule and develop a new heuristic approach to generating SDF schedules with reduced execution time for a particular cache architecture.

In this report, we consider the implementation of well-ordered SDF graphs on a single embedded Digital Signal Processor (DSP). We assume a simple Harvard memory architecture DSP with single-level caches and separate instruction and data-memory. In order to predict execution times, we propose a cache management policy for the data cache and argue that this policy outperforms traditional cache policies when executing SDF models. We also replace the instruction cache by a scratchpad memory with software-controlled replacement policy. Using our data cache and instruction scratchpad policies, we show that different schedules can have vastly different execution times for a given set of data cache and instruction scratchpad sizes. In addition, we show that existing scheduling techniques often create schedules that perform poorly with respect to cache usage. In order to improve cache performance, an optimal cache-aware scheduler would minimize the total cache miss penalty by simultaneously considering both data and instruction miss penalties. Unfortunately, reducing data cache misses often increases instruction scratchpad misses and vice versa. In this report, we show that the number of schedules that must be considered increases exponentially according to the vectorization factor of the schedule. To address this complexity, we develop an SDF scheduling algorithm based on a greedy, cache-aware heuristic. We compare the resulting schedules with schedules generated by existing SDF scheduling schemes. The schedule generated by our algorithm poses an interesting problem of code generation. We also propose a solution to address this problem.

This work is highly applicable in the design of SDF systems that are implemented as Systems on Chip (SoC) with DSP cores.

[71] Classes and Subclasses in Actor-Oriented Design

Edward A. Lee and Stephen Neuendorffer, invited paper, Conference on Formal Methods and Models for Codesign (MEMOCODE), June 22-25, 2004, San Diego, CA, USA.

Abstract: Actor-oriented languages provide a component composition methodology that emphasizes concurrency. The interfaces to actors are parameters and ports (vs. members and methods in object-oriented languages). Actors interact with one another through their ports via a messaging schema that can follow any of several concurrent semantics (vs. procedure calls, with prevail in OO languages). Domain-specific actor-oriented languages and frameworks are common (e.g. Simulink, LabVIEW, and many others). However, they lack many of the modularity and abstraction mechanisms that programmers have become accustomed to in OO languages, such as classes, inheritance, interfaces, and polymorphism. This extended abstract shows the form that such mechanisms might take in AO languages. A prototype of these mechanisms realized in Ptolemy II is described.

[72] Actor-oriented Models for Codesign

Edward A. Lee and Stephen Neuendorffer, In Sandeep Shukla and Jean-Pierre Talpin editors, Formal Methods and Models for System Design, Kluwer, 2004. To appear.

Abstract: Most current hardware engineering practice is deeply rooted in discrete-event modeling and synchronous design. Most current software engineering is deeply rooted in procedural abstractions. The latter says little about concurrency and temporal properties, whereas the former lacks many of modularity capabilities of modern programming languages. Actor-oriented design emphasizes concurrency and communication between components while maintaining modularity. Components called actors execute and communicate with other actors. In contrast to the interfaces in object-oriented design (methods, principally, which mediate transfer of the locus of control), interfaces in actor-oriented design (which we call ports) mediate communication. But the communication is not assumed to involve a transfer of control. This paper discuses the structure of actor-oriented models and shows how data and behavioral type systems enhance modularity and re-use potential while enabling designs that embrace concurrency and time. This paper shows how components can be designed for re-use through parameterization and behavioral polymorphism, and how component specialization can offset the performance costs of doing so.

[73] Actor-Oriented Design of Embedded Hardware and Software Systems

Edward A. Lee, Stephen Neuendorffer and Michael J. Wirthlin, Invited paper, Journal of Circuits, Systems, and Computers, Vol. 12, No. 3 pp. 231-260, 2003.

Abstract: In this paper, we argue that model-based design and platform-based design are two views of the same thing. A platform is an abstraction layer in the design flow. For example, a core-based architecture and an instruction set architecture are platforms. We focus on the set of designs induced by this abstraction layer. For example, the set of all ASICs based on a particular core-based architecture and the set of all x86 programs are induced sets. Hence, a platform is equivalently a set of designs. Model-based design is about using platforms with useful modeling properties to specify designs, and then synthesizing implementations from these specifications. Hence model-based design is the

view from above (more abstract, closer to the problem domain) and platform-based design is the view from below (less abstract, closer to the implementation technology).

One way to define a platform is to provide a design language. Any valid expression in the language is an element of the set. A platform provides a set of constraints together with known tradeoffs that flow from those constraints. Actor-oriented platforms, such as Simulink, abstract aspects of program-level platforms, such as Java, C++, and VHDL. Actor-oriented platforms orthogonalize the actor definition language and the actor composition language, enabling highly polymorphic actor definitions and design using multiple models of computation. In particular, we concentrate on the use of constrained models of computation in design. The modeling properties implied by well chosen constraints allow more easily understood designs and are preserved during synthesis into program-level descriptions. We illustrate these concepts by describing a design framework built on Ptolemy II.

[74] A Behavioral Type System and Its Application in Ptolemy II

Edward A. Lee and Yuhong Xiong, to appear in Aspects of Computing Journal, special issue on "Semantic Foundations of Engineering Design Languages." Draft version: November 10, 2003.

Abstract: Interface automata have been introduced as an interface theory capable of functioning as a behavioral type system. Behavioral type systems describe dynamic properties of components and their compositions. Like traditional (data) type systems, behavioral type systems can be used to check compatibility of components. In this paper, we use interface automata to devise a behavioral type system for Ptolemy II, leveraging the contravariant and optimistic properties of interface automata to achieve behavioral subtyping and polymorphism. Ptolemy II is a software framework supporting concurrent component composition according to diverse models of computation. In this paper, we focus on representing the communication protocols used in component communication within the behavioral type system. In building this type system, we identify two key limitations in interface automata formalisms; we overcome these limitations with two extensions, transient states and projection automata. In addition to static type checking, we also propose to extend the use of interface automata to the on-line reflection of component states and to run-time type checking, which enable dynamic component creation, morphing application structure, and admission control. We discuss the trade-offs in the design of behavioral type systems.

[75] Model-Driven Development - From Object-Oriented Design to Actor-Oriented Design

Edward A. Lee, extended abstract of an invited presentation at Workshop on Software Engineering for Embedded Systems: From Requirements to Implementation (a.k.a. The Monterey Workshop) Chicago Sept.24, 2003.

Abstract: Most current software engineering is deeply rooted in procedural abstractions. These say little about concurrency, temporal properties, and assumptions and guarantees in the face of dynamic system structure. Actor-oriented design contrasts with (and complements) object-oriented design by emphasizing concurrency and communication between components. Components called *actors* execute and communicate with other

actors. While interfaces in object-oriented design (methods, principally) mediate transfer of the locus of control, interfaces in actor-oriented design (which we call ports) mediate communication. But the communication is not assumed to involve a transfer of control. This paper explores the use of behavioral type systems in actor-oriented design.

[76] Soft Walls: Frequently Asked Questions

Edward A. Lee, Technical Memorandum UCB/ERL M03/31, University of California, Berkeley, CA 94720, July 21, 2003.

Abstract: In brief, modern aircraft all have electronics on board that is involved with the control and navigation of the aircraft. Many of the newer planes have computers on board that mediate the commands issued by the pilot and translate those commands into action, for example to bank and turn to the right. It is possible to modify the software in the computers in such a way that an airplane will refuse to enter pre-specified regions. We call these regions "no fly zones" and we call the boundaries of these regions "Soft Walls." If an aircraft is equipped with the Soft Walls system, then if the pilot attempts to enter a no-fly zone, the airplane will be diverted. This happens gently at first, but if the pilot does not cooperate, then the system becomes more assertive. The key principle is to give the pilot as much control over the aircraft as is consistent with the constraint that the airplane does not enter the no-fly zone.

Since its introduction shortly after September 11, 2001, the Soft Walls concept has generated considerable controversy and discussion. This paper collects frequently raised objections to the concept and presents a discussion of the objections. A glossary is provided at the end.

[77] Overview of the Ptolemy Project

Edward A. Lee, Technical Memorandum No. UCB/ERL M03/25, University of California, Berkeley, CA, 94720, USA, July 2, 2003.

Abstract: The Ptolemy Project is an informal group of researchers that is part of Chess (the center for hybrid and embedded software systems) at U.C. Berkeley; see "Acknowledgements" on page 28 for a list of participants. This project conducts foundational and applied research in software based design techniques for embedded systems. Ptolemy II is the current software infrastructure of the Ptolemy Project. For the participants in the Ptolemy Project, Ptolemy II is first and foremost a laboratory for experimenting with design techniques. It is published freely in open-source form. Distribution of open-source software complements more traditional publication media, and serves as a clear, unambiguous, and complete description of our research results. Also, the open architecture and open source encourages researchers to build their own methods, leveraging and extending the core infrastructure provided by the software. This creates a community where much of the dialog is through the software. In addition, the freely available software encourages designers to try out the new design techniques that are introduced and give feedback to the Ptolemy Project. This helps guide further research. Finally, the open source software encourages commercial providers of software tools to commercialize the research results, which then helps to maximize the impact of the work.

Ptolemy II is the third generation of design software to emerge from this group, with each generation bringing a new set of problems being addressed, new emphasis, and (largely) a new group of contributors.

[78] Automatic Verification of Component-Based Real-Time CORBA Applications

G. Madl, S. Abdelwahed, G. Karsai, submitted to the 2004 Conference on Real-time Software and Systems

Abstract: Distributed real-time embedded (DRE) systems often need to satisfy various time, resource and fault tolerance constraints. To manage the complexity of scheduling these systems many methods use Rate Monotonic Scheduling assuming a time-triggered architecture. This paper presents a solution that can prove stricter deadlines for periodic event-driven systems than the Rate Monotonic Analysis. This method captures the reactive behavior of event-driven systems and can automatically verify timed properties of component-based DRE applications that use the publisher/subscriber communication pattern. We demonstrate our approach on real-time CORBA avionics applications.

[79] Computer-Automated Multi-Paradigm Modeling in Control Systems Technology

Mosterman, P., Sztipanovits, J., Engell, S.: IEEE Transactions on Control System Technology, Vol. 12, pp. 223-234, March 2004.

Abstract: Computer automated multi-paradigm modeling provides a formal framework that leverages and unifies different design activities in the each of the following three dimensions: (i) multi formalism, (ii) levels of abstraction and (iii) metamodeling. This paper shows the design of a feedback control algorithm and describes how significant improvements in many aspects (performance, cost, development time) can be achieved by exploiting multi-paradigm notations to combine and relate model-based technologies throughout the control system development process in all stages and at different points of time. Development of control systems that comprise a wide range of control algorithms from disrete event reactive control to continuous PID control benefits from the (i) integration of multiple formalisms, (ii) derivation of different levels of modeling abstractions and (iii) rigorous specification of the different modeling formalisms via metamodeling.

[80] Diagnosis of continuous valued systems in transient operating regions

P. J. Mosterman and G. Biswas, IEEE Trans. Systems, Man, and Cybernetics – A, vol. 29, no. 6, pp. 554-565, 1999

Abstract: The complexity of present day embedded systems (continuous processes controlled by digital processors), and the increased demands on their reliability motivate the need for monitoring and fault isolation capabilities in the embedded processors. This paper develops monitoring, prediction, and fault isolation methods for abrupt faults in complex dynamic systems. The transient behavior in response to those faults is analyzed in a qualitative framework using parsimonious topological system models. Predicted transient effects of hypothesized faults are captured in the form of *signatures* that specify future faulty behavior as higher order time-derivatives. The dynamic effects of faults are analyzed by a *progressive monitoring* scheme till transient analysis mechanisms have to

be suspended in favor of steady state analysis. This methodology has been successfully applied to monitoring of the secondary sodium cooling loop of a fast breeder reactor.

[81] Model-Integrated Computing for Heterogeneous Systems

Neema, S., Dixon, A., Bapty, T., Sztipanovits, J., International Conference on Computing, Communications and Control Technologies (CCCT 2004), Austin TX, August, 2004 (in print)

Abstract: Modern embedded and networked embedded system applications are demanding very high performance from systems with minimal resources. These applications must also be flexible to operate in a rapidly changing environment. High performance with limited resources needs application-specific architectures, while flexibility requires adaptation capabilities. Design of these systems creates unique challenges, since the traditional decomposition of the design space to hardware and software components and to functional and non-functional requirements do not give acceptable performance. Model-Integrated Computing (MIC) is an emerging design technology, which integrates all essential aspects of system design in a general, but highly customizable framework. This paper provides an overview of MIC and shows its application in the design of reconfigurable processing system.

[82] Constraint-Based Design Space Exploration and Model Synthesis

Neema S., Sztipanovits J., Karsai G., Butts, K., EMSOFT 2003, LNCS 2855, pp 290-305.

Abstract: An important bottleneck in model-based design of embedded systems is the cost of constructing models. This cost can be significantly decreased by increasing the reuse of existing model components in the design process. This paper describes a tool suite, which has been developed for component-based model synthesis. The DESERT tool suite can be interfaced to existing modeling and analysis environments and can be inserted in various, domain specific design flows. The modeling component of DESERT supports the modeling of design spaces and the automated search for designs that meet structural requirements. DESERT has been introduced in automotive applications and proved to be useful in increasing design productivity.

[83] Hierarchical Reconfiguration of Dataflow Models

Stephen Neuendorffer and Edward A. Lee, Conference on Formal Methods and Models for Codesign (MEMOCODE), June 22-25, 2004, San Diego, CA, USA.

Abstract: This paper presents a unified approach to analyzing patterns of reconfiguration in dataflow graphs. The approach is based on hierarchical decomposition of the structure and execution of a dataflow model. In general, reconfiguration of any part of the system might occur at any point during the execution of a model. However, arbitrary reconfiguration must often be restricted, given the constraints of particular dataflow models of computation or modeling constructs. For instance, the reconfiguration of parameters that influence dataflow scheduling or soundness of data type checking must be more heavily restricted. The paper first presents an abstract mathematical model that is sufficient to represent the reconfiguration of many types of dataflow graphs. Using this model, a behavioral type theory is developed that bounds the points in the execution of a model when individual parameters can be reconfigured. This theory can be used to

efficiently check semantic constraints on reconfiguration, enabling the safe use of parameter reconfiguration at all levels of hierarchy.

[84] Implementation Issues in Hybrid Embedded Systems

Stephen Neuendorffer, Technical Memorandum No. UCB/ERL M03/22, University of California, Berkeley, CA, 94720, USA, June 24, 2003.

Abstract: This paper presents an approach to the implementation of electronic computation systems whose behavior is tightly integrated with the physical world. We call such systems \textit{hybrid embedded systems}. Such systems are challenging from a design perspective because their behavior is governed by both continuous-state dynamics from the physical world and discrete-state dynamics from the computation. There are several difficulties that appear in such systems. For instance, understanding of the passage of time during computation is critical to understanding how the computation system affects the state of the physical world. Hybrid embedded systems are also inherently concurrent; the computation system operates concurrently with the dynamics of the physical world, in addition to any concurrency that may be designed into the system. In addition, hybrid embedded systems must generally operate within the constraints of traditional embedded systems. They are inevitably constrained computationally, often have a complex computational architecture, and must perform predictably. This paper presents an approach to the design of embedded systems utilizing component-based system models capable of representing concurrency, the passage of time, and both continuous and discrete behaviors. These models allow for automatic generation of system implementations from high-level abstractions as well as the consideration of low-level architectural details where necessary. We show how this technique can be ued to approach difficulties in the design of a complex digital control system.

[85] Semantic Foundations for Heterogeneous Systems

Roberto Passerone, PhD dissertation, Department of EECS, University of California at Berkeley, May 2004.

Abstract: We propose the framework of Agent Algebra as a foundation for the study of heterogeneous systems. Different models of computation can be expressed in terms of a common algebraic structure that includes the usual operations of scoping, instantiation and parallel composition and a relation of refinement. The models can then be related by structure preserving maps. In particular, we study the concept of conservative approximation which preserves refinement verification results from abstract to concrete models. We investigate the relationship between conservative approximations and the established notion of abstract interpretation. We show that an abstract interpretation can be converted to a conservative approximation, provided there exists a second mapping that satisfies certain necessary and sufficient conditions.

The common algebraic structure is also used to study techniques that can be applied to all models of computation. In this work we focus on a characterization of the refinement relationship in terms of substitutability and compatibility, and provide the necessary and sufficient conditions for the existence for each component of a most general environment,

called "mirror". The mirror characterizes the refinement order and it can be used to solve the problem of deriving a local specification for a component given a global specification and a context. We show under which conditions the problem admits an algebraic solution in closed form.

[86] Fault-Tolerant Deployment of Embedded Software for Cost-Sensitive Real-Time Feedback-Control Applications

C. Pinello, L.P. Carloni, and A.L. Sangiovanni-Vincentelli, The Proceedings of the Conference on Design, Automation and Test in Europe (DATE), 2004.

Abstract: Designing cost-sensitive real-time control systems for safety critical applications requires a careful analysis of the cost/coverage trade-offs of fault-tolerant solutions. This further complicates the difficult task of deploying the embedded software that implements the control algorithms on the execution platform that is often distributed around the plant (as it is typical, for instance, in automotive applications). We propose a synthesis-based design methodology that relieves the designers from the burden of specifying detailed mechanisms for addressing platform faults, while involving them in the definition of the overall fault-tolerance strategy. Thus, they can focus on addressing plant faults within their control algorithms, selecting the best components for the execution platform, and defining an accurate fault model. Our approach is centered on a new model of computation, Fault Tolerant Data Flows (FTDF), that enables the integration of formal validation techniques.

[87] Automated Task Allocation on Single Chip, Hardware Multithreaded, Multiprocessor Systems

William Plishker, Kaushik Ravindran, Niraj Shah, and Kurt Keutzer, Workshop on Embedded Parallel Architectures (WEPA-1), February, 2004.

Abstract: The mapping of application functionality onto multiple multithreaded processing elements of a high performance embedded system is currently a slow and arduous task for application developers. Previous attempts at automation have either ignored hardware support for multithreading and focused on scheduling, or have overlooked the architectural peculiarities of these systems. This work attempts to fill the void by formulating and solving the mapping problem for these architectures. In particular, the task allocation problem for a popular multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor, is encoded into a 0-1 Integer Linear Programming problem. 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.

[88] A Programmable Microkernel for Real-Time Systems

Marco A. Sanvido, Christoph M. Kirsch, and Thomas A. Henzinger, Submitted.

Abstract: 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) and proactivity (task scheduling) are fully programmable. The microkernel,

which we implemented on a StrongARM processor, consists of two interacting 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, generally in the 0.2-0.3% range.

[89] A Distributed Algorithm for Acoustic Localization Using a Distributed Sensor Network

P. Schmidt, I. Amundson and K.D. Frampton, Journal of the Acoustical Society of America, Vol. 115, No. 5, Pt. 2, pp. 2578, 2004. To be presented at the 147th Meeting of the Acoustical Society of America, New York, May 24-28, 2004

Abstract: An acoustic source localization algorithm has been developed for use with large scale sensor networks using a decentralized computing approach. This algorithm, based on a time delay of arrival (TDOA) method, uses information from the minimum number of sensors necessary for an exactly determined solution. Since the algorithm is designed to run on computational devices with limited memory and speed, the complexity of the computations has been intentionally limited. The sensor network consists of an array of battery operated COTS Ethernet ready embedded systems with an integrated microphone as a sensor. All solutions are calculated as distinct values, and the same TDOA method used for solution is applied for ranking the accuracy of an individual solution. Repeated for all combinations of sensor nodes, solutions with accuracy equivalent to complex array calculations are obtainable. Effects of sensor placement uncertainty and multipath propagation are quantified and analyzed, and a comparison to results obtained in the field with a large array and a centralized computing capability using a complex, memory intensive algorithm is included.

[90] Comparing Network Processor Programming Environments: A Case Study

Niraj Shah, William Plishker, Kurt Keutzer,
Workshop on Productivity and Performance in High-End Computing (P-PHEC), February, 2004.

Abstract: Network processors have emerged as prominent examples of multiprocessor application-specific programmable architectures. While there have been significant

architectural developments in this field, widespread adoption will be predicated on productively programming high performance applications on these architectures. This paper presents a case study of two programming environments for a common network processor, the Intel IXP1200. We compare the development process, achievable performance, and resource usage of the final implementations using these two programming approaches and draw conclusions regarding the advantages and disadvantages of these approaches.

[91] Optimal Control for a class of Stochastic Hybrid Systems

Ling Shi, Alessandro Abate, Shankar Sastry, submitted to CDC04.

Abstract: In this paper, an optimal control problem over a "hybrid Markov Chain" (hMC) is studied. A hMC can be thought of as a traditional MC with continuous time dynamics pertaining to each node; from a different perspective, it can be regarded as a hybrid system with random discrete switches induced by an embedded MC. As a consequence of this setting, the index to be maximized, which depends on the dynamics, is the expected value of a non deterministic cost function. After obtaining a closed form for the objective function, we gradually suggest how to device a computationally tractable algorithm to get to the optimal value. Furthermore, the complexity and rate of convergence of the algorithm is analyzed. Proofs and simulations of our results are provided; moreover, an applicative and motivating example is introduced.

[92] Kalman Filtering With Intermittent Observations

B. Sinopoli, L. Schenato, M. Franceschetti,K. Poolla, M. Jordan, S. Sastry, IEEE Trans. on Automatic Control. In press 2004.

Abstract: In this paper, we show the existence of a critical probability for the arrival of the observation at each time step in the discrete Kalman filter, that is required to have bounded average estimation error covariance.

[93] Templating Transformations for Bitstream Programs,

Armando Solar-Lezama, Rastislav Bodik, HPCA Workshop on Productivity and Performance in High-End Computing (P-PHEC 2004), held in conjunction with HPCA 2004, Madrid, Spain.

Abstract: High-performance code is typically the product of a tight collaboration between a domain expert, who writes the high-level model, and a system expert, who tunes the high-level code for a particular machine. Lacking tool support, this collaboration is not very productive: the performance tuning process involves actually rewriting the original, clean code into a large, hard-to-maintain high-performance code. The result is that once the system expert is done tuning, the domain expert has a hard time modifying his model. We present a tool for making the collaboration between system and domain experts more productive. The domain expert first writes his algorithm in a high-level domain- specific language (DSL). (In this paper, we focus on the domain of bitstream programs.) The system expert then optimizes the original program not by manually rewriting it but instead by specifying his domain- and machine-specific transformations in a higher-order, transformation-specification language (TSL). By empowering the system expert with the TSL, we are essentially allowing him to create a

domain-specific optimizer. To support rapid prototyping of optimizations, we allow him to specify a template of the transformation, and have the tool fill in the details, by constraining the optimized code to behave like the original one. Templating not only improves productivity, but also makes the transformation applicable after the original model is modified.

[94] A Domain-Specific Visual Language For Domain Model Evolution

J. Sprinkle, G. Karsai, to appear in Journal of Visual Languages and Computing, 2004.

Abstract: Domain-specific visual languages (DSVLs) are concise and useful tools that allow the rapid development of the behavior and/or structure of applications in well-defined domains. These languages are typically developed specifically for a domain, and have a strong cohesion to the domain concepts, which often appear as primitives in the language. The strong cohesion between DSVL language primitives and the domain is a benefit for development by domain experts, but can be a drawback when the domain evolves---even when that evolution appears to be insignificant. This paper presents a domain-specific visual language developed expressly for the evolution of domain-specific visual languages, and uses concepts from graph rewriting to specify and carry out the transformation of the models built using the original DSVL

[95] Platform modeling and model transformations for analysis

T. Szemethy, G. Karsai, IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS 2004

Abstract: The paper discusses an approach for modeling software execution platforms, and the use of these models in determining the run-time properties of component-based applications that are executed on the platform. The modeling is based on the use of timed-automata, and on the transformation of the systems models.

[96] Model-Integrated Computing Infrastructure for Fault Management

Sztipanovits, J, in Proc. of DX-14, 14th International Workshop on Principals of Diagnosis, Washington DC, June 12, 2003

Abstract: In functional design, complexity of large systems is managed by vertical and horizontal composition. In vertical composition, systems are designed in layers utilizing fundamentally different technologies. Commonly used layers in information systems are: Material, Device and Circuit Layer, Hardware/System Layer, OS/Communication Layer, Middleware Layer(s), Application Layer. The individual layers are designed by using layer-specific abstractions and composition techniques and provide well defined behavior for each other. Separation of the layers via well defined, guaranteed interfaces is crucial in managing design complexity. The core design task on each layer is to use resources provided by a lower layer to implement functionalities demanded by a higher layer. Horizontal composition is performed in a single layer for creating aggregate components using the dominant compositionality principle of the layer. This paper describes a Model-Integrated Computing approach to systematically constructing later-specific models and interfaces for fault management.

[97] Design Space Construction and Exploration: A Model Integrated Computing Approach

Sztipanovits, J.: MPSOC 2003, Chamonix, France, June 8, 2003 (presentation in Workshop Proceedings)

Abstract: The presentation described the use of domain-specific modeling languages for constructing complex design spaces in the SoC domain.

[98] Model-Integrated Computing for Automotive Applications

Sztipanovits, J., Neema, S., Chen, K., Automotive Software Workshop San Diego, January 2004, LNCS, to appear.

Abstract: This paper describes Model-Integrated Computing, which comprises modeling, model analysis and model-based software generation technologies as foundation for embedded software composition. The primary focus is semantic anchoring of domain-specific modeling languages using model transformations and precise, formally specified Models of Computation (MoC).

[99] Experiments on the Decentralized Vibration Control with Networked Embedded Systems

Tao Tao and K.D. Frampton, Accepted by the 2004 ASME International Mechanical Engineering Conference and Exposition, Anaheim CA, November 2004

Abstract: The early promise of centralized active control technologies to improve the performance of large scale, complex systems has not been realized largely due to the inability of centralized control systems to "scale up"; that is, the inability to continue to perform well when the number of sensors and actuators becomes large. Now, recent advances in Micro-electro-mechanical systems (MEMS), microprocessor developments and the breakthroughs in embedded systems technologies, decentralized control systems may see these promises through. A networked embedded system consists of many nodes that possess limited computational capability, sensors, actuators and the ability to communicate with each other over a network. The aim of this decentralized control system is to control the vibration of a structure by using such an embedded system backbone. The key attributes of such control architectures are that it be scalable and that it be effective within the constraints of embedded systems. Toward this end, the decentralized vibration control of a simply supported beam has been implemented experimentally. The experiments demonstrate that the reduction of the system vibration is realized with the decentralized control strategy while meeting the embedded system constraints, such as a minimum of inter-node sensor data communication, robustness to delays in sensor data and scalability.

[100] Towards Generation of High-performance Transformations

Attila Vizhanyo, Aditya Agrawal, Feng Shi, submitted to and accepted for presentations at the Generative Programming and Component Engineering Conference 2004

Abstract: In this paper we introduce a graph rewriting language, called Graph Rewriting and Transformation (GReAT), and a code generator tool, which together provide a

programming framework for the specification and efficient realization of graph rewriting systems. We argue that the performance problems frequently associated with the implementation of the transformation can be significantly reduced by adopting language and algorithmic optimizations and partial evaluation.

[101] Finding and Preventing Run-Time Error Handling Mistakes

Westley Weimer and George Necula, In Proceedings of the Object-Oriented Programming Systems, Languages and Applications (OOPSLA04), Vancouver, 2004.

Abstract: 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 code and for restoring invariants in such exceptional situations. We 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. The analysis is unsound and produces false positives, but a few simple filtering rules suffice to remove them in practice. The remaining mistakes were manually verified. These mistakes cause sockets, files and database handles to be leaked along some paths. We present a characterization of the most common causes of those errors and discuss the limitations of exception handling, finalizers and destructors in addressing them. Based on those errors, we propose a programming language feature that keeps track of obligations at run time and ensures that they are discharged. Finally, we present case studies to demonstrate that this feature is natural, efficient, and can improve reliability; for example, retrofitting a 34kLOC program with it resulted in a 0.5% code size decrease, a surprising 17% speed increase (from correctly deallocating resources in the presence of exceptions), and more consistent behavior.

[102] A "Flight Data Recorder" for Enabling Full-system Multiprocessor Deterministic Replay

Min Xu, Rastislav Bodik, Mark Hill, The 30th International Symposium on Computer Architecture, San Diego, CA, June 2003.

Abstract: Debuggers have been proven indispensable in improving software reliability. Unfortunately, on most real-life software, debuggers fail to deliver their most essential feature - a faithful replay of the execution. The reason is non-determinism caused by multithreading and nonrepeatable inputs. A common solution to faithful replay has been to record the non-deterministic execution. Existing recorders, however, either work only for data-race-free programs or have prohibitive overhead. As a step towards powerful debugging, we develop a practical low-overhead hardware recorder for cache coherent multiprocessors, called Flight Data Recorder (FDR). Like an aircraft flight data recorder, FDR continuously records the execution, even on deployed systems, logging the execution for post-mortem analysis. FDR is practical because it piggybacks on the cache coherence hardware and logs nearly the minimal thread ordering information necessary to faithfully replay the multiprocessor execution. Our studies, based on simulating a four-

processor server with commercial workloads, show that when allocated less than 7% of system's physical memory, our FDR design can capture the last one second of the execution at modest (less than 2%) slowdown.

[103] Image and Video Processing Libraries in Ptolemy II

James Yeh, Master's Report, Technical Memorandum No. UCB/ERL M03/52, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

Abstract: The signal processing done in Ptolemy II has mainly been focused on onedimensional signal processing. The goal of this project was to be able to introduce both two-dimensional signal processing (in the form of images), and three dimensional signal processing (in the form of video) components into Ptolemy II.

[104] A Model of Computation with Push and Pull Processing

Yang Zhao, Master's Report, Technical Memorandum No. UCB/ERL M03/51, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

Abstract: This report studies a model of computation (MoC) that supports Push-Pull communication between software components. Push represents message transfer that is initiated by the producer. On the contrast, Pull represents message transfer that is initiated by the consumer.

Push-Pull messaging mechanisms have been used in various domains, such as the CORBA Event service, the Click modular router, inter-process communication among multi-processors, data dissemination in a peer to peer network. Formalizing the computation and communication in these domains to a MoC facilitates reusability of programming models and software architectures, and allows user to model and analyze more complex systems.

This model of computation has been implemented as the Component Interaction (CI) domain in Ptolemy II, a hierarchical and heterogeneous modeling environment. Composition of the CI domain with other domains in Ptolemy II is discussed. This report also connects CI to other data-flow models of computation, such as Process Networks (PN), Synchronous Data Flow (SDF) and Dynamic Data Flow (DDF), and discusses the uses of CI in the execution of distributed systems.

[105] Communication Systems Modeling in Ptolemy II

Ye Zhou, Master's Report, Technical Memorandum No. UCB/ERL M03/53, University of California, Berkeley, CA, 94720, USA, December 18, 2003.

Abstract: Synchronous Dataflow (SDF) is useful in modeling communications and signal processing systems. We describe a communication actor library based on the SDF semantics in Ptolemy II and show how to use these actors to simulate communication systems. However, many communication and signal processing systems nowadays use adaptive algorithms and control protocols. These sometimes violate SDF principles in that SDF actors must have fixed rates during execution. A. Girault, B. Lee, and E. A. Lee proposed in a new model of computation called Heterochronous Dataflow (HDF). HDF extends SDF by allowing rate changes in actors during execution. HDF is a

heterogeneous composition of SDF and Finite State Machines (FSM). We describe an HDF domain implementation in Ptolemy II. Examples and discussions are given to show how HDF can be used in various forms.

2.3. Project Training and Development

2.4. Outreach Activities

Our agenda is 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. Based on the ongoing effort at UCB, we have set out to rearchitect and retool undergraduate teaching at the participating institutions, and to make the results widely available to encourage critical discussion and facilitate adoption. In addition, have recruited new undergraduate students (mostly juniors) from minority institutions through the established REU programs SUPERB-IT at UCB and SURGE at VU to participate in the research of the project.

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.

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 an effort to define an upper division course in embedded software (aimed at juniors and seniors). We will be drawing on the extensive experience we have with graduate courses in this area.

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 increase diversity in the graduate school pipeline by affirming students' motivation for graduate study and strengthening their qualifications.

SUPERB-IT participants spent eight weeks at UC Berkeley during the summer of 2003 and have begun the summer of 2004 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 supported a group six SUPERB-IT students in 2003, and organized projects (described below) in hybrid and embedded software systems technology, primarily in the area of software tools supporting the design process. The students were hosted by the Chess center at Berkeley (Center for Hybrid and Embedded Software Systems). Since the 2004 session has only just started, we focus here on reporting on the 2003 session.

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 2003 were Antonio Yordan-Nones, Ismael Sarmiento, Rakesh Reddy, Colin Cochran, Mike Okyere, and Philip Baldwin. The six students built a suite of applications and infrastructure for embedded systems design using the Ptolemy II software infrastructure. Their eight week projects began with an intensive group training in the Chess software lab that familiarized them with the use a CVS code repository, the Eclipse integrated development environment, construction of applications in Ptolemy II, and design and

construction of actors for Ptolemy II. They were guided to use an Extreme Programming (XP) software engineering style, which includes pair programming, extensive use of automated test suites, and design and code reviews. All but one of the students had sufficient experience with Java programming that little time was required for familiarization with Java. The one student with little Java experience (Philip Baldwin) focused on building models of distributed, wireless, real-time systems in Ptolemy II, a project that eventually led to VisualSense (see above and [13]). These models used infrastructure built in Java by the other students. In the XP context, he functioned as the 'customer.' Three graduate student mentors facilitated the process, and the operation was coordinated and directed by Professor Edward Lee. Although the students worked together and interacting extensively, each was responsible for an individual project, as described below. Their project posters and reports are available at http://chess.eecs.berkeley.edu/projects/ITR/2003/superb.htm.

Project: Interactive embedded systems showcase

Student: Antonio Yordan-Nones

> This student had a strong interest in art and technology, with background building applications using Java servelets and server pages, video and sound. His responsibility was to design and construct an embedded systems showcase that now occupies a glassed-in-case outside the Chess software lab. This showcase includes a computer, microphone, video camera, display, and X-10 devices to control lights. The objective was to use one or more Ptolemy II applications to engage viewers of the showcase in interactive displays. The student was given freedom to be creative and was encouraged to use results of the other SUPERT-IT students.

Project: Wireless systems modeling

Student: Philip Baldwin

> This was the one student with no Java experience. However, he had previously been involved in a project that modeled bluetooth devices at the networking and application level. His responsibility was to construct Ptolemy II models of distributed wireless embedded systems such as sensor nets and web-integrated embedded applications. This project led to the creation of VisualSense (see above and [13]).

Project: Actor-oriented construction of interactive 2-D graphics

Student: Ismael Sarmiento

> This student had experience using Java 2-D to build interactive graphics-intensive games. His responsibility was to build a suite of Ptolemy II actors that construct and dynamically morph 2-D scenes, and to show how these actors can be used to build customized user interfaces and displays for embedded systems models. The resulting graphics infrastructure was used by the first student in the interactive showcase and by the wireless systems modeling project for animated interactive displays of the models used in those contexts.

Project: Secure transport of mobile models and data in distributed applications

Student: Rakesh Reddy

This student had previous experience with encryption and decryption technologies and with Java software design. His responsibility was to construct Ptolemy II actors support secure distributed models, and to demonstrate their use in with mobile, web-integrated applications. The resulting technology was evolved by our research staff and became part of the Ptolemy II 4.0 software release as a security library.

Project: Actor-oriented design of smart-home systems

Student: Colin Cochran

Like the first student, this student had an interest in art and technology, with experience in avionics software testing, web page design and construction, and Java software design. His responsibility was to create a suite of Ptolemy II actors that interacted through the serial port of a host computer with X-10 controllers (which communicate over power lines to control lights and electrical devices) and motor controllers. The resulting technology is being used to control lights in the 'showcase.'

Project: Actor-oriented design of web-integrated string manipulation

Student: Mike Okyere

This student had quite a bit of previous Java experience, primarily with e-commerce applications. His responsibility was to construct Ptolemy II actors for processing textual data, including for example XML data and text embedded in HTTP coming from HTML forms. His project evolved into an exploration of the use of location information in interactive map-based applications.

Project: Platform Based Reconfigurable Hardware Exploration via Boolean Constraints

Student: Iyibo Jack - University of Washington (Sangiovanni's group)

This project looked to take a high level description of an application's requirements and transform this via a series of constraints into an abstraction of the possible configurations of a reconfigurable hardware device. Taking this abstraction (a platform), it then estimated what the performance of various instances of this platform would be on the device (Cypress Semiconductor's PSOC). This methodology was both top down and bottom up in its use of constraints and performance estimation. It framed the construction of platform instances as Boolean constraint formulations and solves them using the principles of Boolean Satisfiability.

Plans for 2004

The students being supported by Chess in the summer of 2004 at Berkeley are Elizabeth Fatusin, Basil Etefia, and Rafael Garcia. At the time of writing this report, the projects are still in the planning phase. The students will be working on an infrastructure and toolbox which will be used for various portions of hybrid systems modeling, simulation, verification, validation, and code generation. The summer will begin with an introduction to hybrid systems, as well as immersive assignments in programming (both Python and C++) and source code management.

Two graduate student mentors have been identified to facilitate the process, and the operation is being coordinated and directed by Dr. Jonathan Sprinkle (a postdoc under the supervision of Professor Shankar Sastry). Although the students are working together and interacting extensively, each will be responsible for one (or more) of the individual projects described below. The students will be selected for the particular project based on their predisposition to a particular topic, as well as demonstrated skills in the first week of study.

Framework: Hyper

The framework into which students will be integrating toolboxes is a completely new design and implementation for hybrid systems modeling and simulation. Emphasis is placed on *internal* maintenance of system models, which are then exposed to any number of *toolboxes*. The toolboxes will be the major focus of the SUPERB-IT students, while the common components and internal representation/design of the core infrastructure (Hyper itself) will be undertaken by Dr. Sprinkle. Hyper's core infrastructure and visualization will be integrated mainly in Python, which will allow for execution on multiple platforms. The rewriting of computationally expensive portions of the framework in platform-independent C++ will allow for fast execution of frequently called routines, or complex analysis/solving toolboxes.

Project: Ordinary Differential Equation Solver

This project will be focused on producing numerical solutions to differential equations, based on models that are assumed to exist in a well-defined framework. The student working on this project will perform prototypes of the implementation in Python, and then (time permitting) re-implement the solution in C++. This toolbox will be an important implementation portion of the framework, since many of the future toolboxes will use it.

Project: Switched System Simulator

This project will be focused on providing a simulation of a switched system. The student will take an input set of states combined with conditions for switching between them. The visualization is independent of the *trace* of execution, which will be designed by the student to be semantically rich, while remaining decoupled from preconceived notions of execution style, or visualization.

Project: Visualization interfaces

Visualization of the simulation, verification, validation, and actual model entry are important for the success of the framework. This student will describe and devise visualization schemes, and how they should be used by machine-produced results (e.g., the execution trace of a switched

system may provide multiple visualizations of these switches based on the simulated behavior: a real-time display of the states and transitions, or a time-axis plot of values in the system). The student will likely do prototyping using MATLAB as the visualization tool, and follow up with Python implemented visualizations, to avoid requiring MATLAB as a runtime element for the framework.

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/fountain/Teaching/

In the SIPHER activities, we organized a summer internship in 2003 for six participants from underrepresented groups. The students are organized into three groups who solve different embedded software development problems. We developed a small modeling language to enable the modeling of embedded systems, we created two implementations of a run-time platform (one in Java, one in C++), and created model transformers that map models expressed in the modeling language into code that executes on the run-time platform. The students used the 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 projects and the students supported at Vanderbilt in 2003 were:

- 'Visual Tracking': Bina P. Shah, Edwin Vargas, and Trione Vincent (REU),
- 'LEGO Mindstorm Robot Control': Rachael A. Dennison, David Garcia, and Danial Balasubramanian (REU),
- 'TAB Robot Control': Michael J. Rivera Jackson, Nickolia Coombs (REU),
- 'Control of Adaptive Structures': Shantel Higgins (with Efosa Omojo).

The students worked on four 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, three graduate student mentors assisted and guided the student projects. During the preceding Spring semester the three mentors have created prototype solutions for the projects that serve as 'reference' for the student projects. The brief bio of the students and the description of projects are below.

Student: Bina P. Shah

> Bina is from Birmingham, Alabama, where she was a senior at the University of Alabama at Birmingham majoring in computer science. She is a member of the Golden Key International Honor Society, Phi Kappa Phi Honor Society, and is in the National Society

of Collegiate Honors. She was very involved on her campus as well. She was part of Trail Blazers, UAB's official student recruitment team for the past 2 years, as well as serving as Vice-President for the International Mentors program. She was also actively involved in the Association for Computing Machinery (ACM), where she represented UAB's C++ team at the Southeast USA Regional Programming Contest. As a community servant, she participated with the Indian Cultural Association (ICA) doing volunteer work at homeless shelters and participating in canned food drives. Bina wants to pursue graduate study in Electrical and Computer Engineering, specializing in the design and development of new software.

Student: Edwin Vargas

Edwin was a senior at Middle Tennessee State University, where he was a double major in computer science and mathematics. He is originally from Bogota, Colombia, where political and social unrest forced him to come to the United States. Edwin has been involved in martial arts for over seventeen years, specializing in Tae-Kwon-Do, and he practices with the martial arts club at MTSU. He has been teaching and competing actively in the time, and he won the national tournament in 1996 and 1997. Also at MTSU, Edwin is a member of the Hispanic Student Association and the local chapter of the Association for Computer Machinery (ACM). He has been on the Dean's list at MTSU and is a 2002-2003 recipient of the National Science Foundation CSEM scholarship. Edwin has a strong background in computer architecture and systems design and hopes to use his mathematical and computer science skills to pursue a PhD in Computer Science. He currently lives in Murfreesboro with this lovely wife.

Student: Trione Vincent

Trione was a rising senior at Fisk University double majoring in Computer Science and Computer Engineering. She is from New Orleans, Louisiana. She has spent her first three years at Fisk and will complete her engineering degree at Vanderbilt. While at Fisk, Trione is a member of the Big Sistas mentoring program and the Fisk University Pep Squad. She has received a Fisk Academic Scholarship and a scholarship from NASA. Trione is interested in research especially in the field of embedded systems and software. She wants to build her career working in the hardware aspect of computer science and engineering.

Project: Visual Tracking

Bina, Edwin, and Trione worked on a project that creates a control system for the visual tracking of objects using a PC and a remote controller camera. The objective was to create a model of the control system, develop the individual components (e.g. camera controller, object recognition module, etc), and then use the model-integrated computing tools to put together the final application.

Student: Rachael A. Dennison

Rachael was a senior at the University of Alabama at Birmingham, where she was also majoring in computer science. Her hometown is Greenville, Alabama. Rachael is very active and loves the outdoors. She enjoys fishing, hunting, swimming, snorkeling, scuba diving, reading, fossil hunting, and camping. She excels in academics at UAB by being on the Presidential Honor Roll and being a member of the Phi Kappa Phi Honor Society as well as the Golden Key International Honor Society. Also, she was nominated by the Computer Science Department for the Dean's Award. After graduation, Rachael wants to work in the field of software engineering and research into ways to make software more in tune with the physical environment that it describes. Rachael wants to develop software applications and model hardware design to handle real-time information in a safe, reliable, and accurate way.

Student: David Garcia

David was a rising senior at Vanderbilt University double-majoring in Computer Science and Mathematics. He is originally from California but now resides in Las Vegas. David was active on campus by being a member of the Society of Hispanic and Professional Engineers and the Vanderbilt Association of Hispanic Students, where he served as a board member. He also used his computer skills by working with Vanderbilt's Information Technology Services (ITS) as a support technician and helped incoming freshmen with configure their network system and troubleshooting local problems. David has also been honored as receiving High Honors on the Dean's list at Vanderbilt. David's career interests lie in the area of game development and design. He wants to pursue graduate degrees in computer science specializing in gaming software and conduct research in artificial intelligence and computer graphics. Because of his avid interest in games, he received his PlayStation Technician certification working for PlayStation as a Level 1 Support specialist last summer as part of the E3 summer gaming conference.

Student: Daniel Balasubramanian

Daniel was a rising senior at Tennessee Technological University majoring in Computer Science. He is originally from Nashville, Tennessee. Daniel is a member of several organizations on campus, including the Association for Computing Machinery (ACM) and the Jazz Band, as well as being very active in the Honors program at Tennessee Tech. He is on the chair of the Tutoring Committee and a member of the Ecology Committee and the Program Big-Sib. Academically, he has received several honors and distinctions. These include a NASA Scholarship, a Rotary Club Scholarship, a TTU Housing Scholarship, and the Earl McDonald Academic Achievement Award, which covers full

tuition at Tennessee Tech. Daniel has a real passion for learning, not only in his but also in other areas of science and mathematics. He has done extensive work in website development at Vanderbilt and at TTU. He is very interested in research, specifically in NASA's Gravity Probe B project, where he would like to combine his computer programming skills with the physical aspects that actually describes the system.

Project: LEGO Mindstorm Robot Control

Rachael, David, and Daniel worked on developing model-based control software for Lego robots. The challenge problem was to develop the models for the control software that will allow the robot to navigate in an environment, react to unforeseen objects (obstacles) and execute eaxploration tasks. They used a modeling language for creating high-level models of the controllers, develop the code for individual components, and generated the full Java code for the controller to be executed on the RCX.

Student: Michael J. Rivera-Jackson

Michael was a rising junior on a full academic scholarship at Morehouse College in Atlanta, Georgia, where he was a double major in Computer Science and Spanish. Michael is originally from Belle Chasse, Louisiana. He was very involved on campus at Morehouse, participating in the Louisiana Club, the Spanish, French, and Japanese Clubs, SGA, the Feminist Majority Leadership Alliance, and the Hip-Hop Collective. In addition to these organizations, he had time to give back to his community by volunteering at the Charles Drew Charter Elementary School, mentoring and tutoring children in reading, mathematics, and science, for which he was recognized as Mentor of the Month in November of 2002. Michael's hobbies include surfing the net, learning languages, poetry, reading, and composing. Michael is interested in internet research and wants to continue to give back to his community by aspiring to become CEO of a software company that produces learning software specifically for children to explore all different types of areas.

Student: Nickolia S. Coombs

Nickolia was a junior at North Carolina A&T State University where is a computer science major. He is originally from Jacksonville, North Carolina. He enjoys following OTC stocks, racquetball, and public speaking. On campus, he is a member of the National Society of Black Engineers, the History Club, and is a tutor for in discrete mathematics and computer science. Also, he is the fund-raising chair for the Association for Computing Machinery. He has been on the Dean's list and is a recipient of a NSBE Scholarship, the Honors 4.0 award, honors in the ACM Programming Competition, and participated in the IBM Project Breakthrough Summer internship last summer. Nickolia's research interests include understanding more about embedded software, especially in the mathematical aspect of simulation. After graduation, Nickolia wants to work in industry, but is interested in also pursuing graduate degrees in computer science and engineering.

Project: TAB Robot Control

Michael and Nickolia worked on an another robot control problem. Their robot had better sensors, and it was much smaller than the LEGO robot. They built embedded software for the robot using model-based techniques that allow the robot to solve a maze problem, as well as build a map of the maze. They used the same design tools as the other projects but in a different problem setting.

Student: Shantel Higgins

> Shantel was a rising senior at Vanderbilt University majoring in Electrical Engineering. Shantel is originally from Sugarland, Texas, a small suburb of Houston. She was the treasurer of her sorority, Delta Sigma Theta and was the chair of their First Annual Aids Awareness Walk. She was the community service chair for the Black Student Alliance at Vanderbilt as well as a volunteer for the local YMCA in Nashville. Shantel was an active member of the National Society of Black Engineers and the Society of Women Engineers, as well as a coach for an intramural women's basketball team. Shantel has received honors from the School of Engineering at Vanderbilt and she is the recipient of the Sam McCleskey Engineering Honor Scholarship. Shantel has enjoyed her time at Vanderbilt and she credits her interest and enthusiasm in research and technology from her Vanderbilt curriculum. Shantel's interest is in the field of wireless communications, sparked by her semiconductors class and integrated circuit design and fabrication class. She hopes to pursue a master's and doctoral degrees in this area.

Project: Control of Adaptive Structures

Shantel (with Efosa Ojomo, who is independently supported) worked on developing a real-time controller for a 'smart structure': a vibrating steel beam. The objective was to create a small, embedded system that detects the onset of vibrations in a beam and actively compensate for them by acting on the beam. First, they built a simulation model for the plant and the controller in Simulink/Stateflow. Once the control algorithm was determined they created an implementation of it on a PC-104 embedded processor platform. The physical implementation included a piezo element, which acted both as sensor and actuator.

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

- Bollobás, B. and O. Riordan, “Robustness and vulnerability of scale-free random graphs,” has appeared as the first paper in the first issue of *Internet Mathematics*, 2004, 1—31

- Edwards, Stephen A. and Edward A. Lee, "The Semantics and Execution of a Synchronous Block-Diagram Language," *Science of Computer Programming*, Vol. 48, no. 1, July 2003.

- Franceschetti, M., J. Bruck, and L. Schulman, "A Random Walk Model Of Wave Propagation," *IEEE Trans. on Antennas and Propagation*, 52(5), May 2004.

- Karsai, G. A. Agrawal, F. Shi, J. Sprinkle, "On the Use of Graph Transformation in the Formal Specification of Model Interpreters," *Journal of Universal Computer Science*, Volume 9, Issue 11, 2003

- Karsai, G. Maroti, M., Ledeczi, A. Gray, J., Sztipanovits, J. "Composition and Cloning in Modeling and Meta-Modeling," *IEEE Transactions on Control Systems Technology*, March 2004, pp 263-278, Volume 12, Issue: 2

- Lee, Edward A., Stephen Neuendorffer and Michael J. Wirthlin, "Actor-Oriented Design of Embedded Hardware and Software Systems," Invited paper, *Journal of Circuits, Systems, and Computers*, Vol. 12, No. 3 pp. 231-260, 2003.

- Mosterman, P. J., and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Systems, Man, and Cybernetics – A*, vol. 29, no. 6, pp. 554-565, 1999

- Mosterman, P., Sztipanovits, J., Engell, S., "Computer-Automated Multi-Paradigm Modeling in Control Systems Technology," *IEEE Transactions on Control System Technology* Vol. 12, pp. 223-234, March 2004

- Schmidt, P., I. Amundson and 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. Presented at *147th Meeting of the Acoustical Society of America*, New York, May 24-28, 2004

3.2. Conference Papers

- Abdelwahed, S., G. Karsai and G. Biswas, "Online Safety Control of a Class of Hybrid Systems." *41st IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 1988-1990

- Abdelwahed, S., J. Wu, G. Biswas, J. W. Ramirez, and E. J. Manders, "Online Hierarchical Fault-Adaptive Control for Advanced Life Support Systems," *International Conference On Environmental Systems*, Denver, CO, July 2004

- Agrawal A., Karsai G., Ledeczi A., "An End-to-End Domain-Driven Software Development Framework," *18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Anaheim, California, October 26, 2003.

- Agrawal, A., Simon, G. Karsai, G., "Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT," *Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT)* 2004. To appear in *Electronic Notes on Theoretical Computer Science*, Elsevier

- Ames, A. D. and S. Sastry, "Affine Hybrid Systems," in Hybrid Systems: Computation and Control, LNCS Vol. 2993, pg. 16-31, Springer-Verlag, 2004.

- Ammons, Glenn, David Mandelin, Rastislav Bodik, James Larus, "Debugging Temporal Specifications with Concept Analysis," ACM SIGPLAN Conference on Programming Language Design and Implementation, San Diego, CA, June 2003.

- Baldwin, Philip, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, "Modeling of Sensor Nets in Ptolemy II," In *Proc. of Information Processing in Sensor Networks*, (IPSN), April 26-27, 2004, Berkeley, CA, USA.

- Benveniste, Albert, Luca P. Carloni, Paul Caspi, and Alberto L. Sangiovanni-Vincentelli, "Heterogeneous Reactive Systems Modeling and Correct-by-Construction Deployment," *EMSOFT* 2003.

- Biswas, G., G. Simon, G. Karsai, S. Abdelwahed, N. Mahadevan, T. Szemethy, J. Ramirez, G. Péceli and T. Kovácsházy, "Self Adaptive Software for Fault Adaptive Control," *Proc. International Workshop on Self Adaptive Software*, Washington D.C., June 2003.

- Biswas, G., G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, G. Karsai, "A robust method for hybrid diagnosis of complex systems," *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS),* Washington, D. C., pp. 1125-1130, June 2003

- Bollobás, Béla, Christian Borgs, Jennifer Chayes, and Oliver Riordan, "Directed Scale-free Graphs," *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, 2003.

- Chakrabarti, Arindam, Luca de Alfaro, Thomas A. Henzinger, and Marielle Stoelinga, "Resource Interfaces," *EMSOFT* 2003.

- Chatterjee, Krishnendu, Thomas A. Henzinger, and Marcin Jurdzinski, "Games with Secure Equilibria," *Proceedings of the 19th Annual Symposium on Logic in Computer Science* (LICS), IEEE Computer Society Press, 2004.

- Chatterjee, Krishnendu, Marcin Jurdzinski, and Thomas A. Henzinger, "Quantitative Stochastic Parity Games," *Proceedings of the 15th Annual Symposium on Discrete Algorithms* (SODA), SIAM, 2004, pp. 114-123.

- Carloni, L.P. and A.L. Sangiovanni-Vincentelli, “A Formal Modeling Framework for Deploying Synchronous Designs on Distributed Architectures,” *First International Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures* (FMGALS 2003).

- de Alfaro, Luca, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Marielle Stoelinga, “Model Checking Discounted Temporal Properties,” *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (TACAS), Lecture Notes in Computer Science 2988, Springer-Verlag, 2004, pp. 77-92.

- Emerson, M. J., Sztipanovits, J. and Bapty, T., “MIC, MDA, and MOF,” *IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS* 2004

- Fields, Brian, Rastislav Bodik, Mark D. Hill, Chris J. Newburn, “Using Interaction Costs for Microarchitectural Bottleneck Analysis,” *The 36th Annual IEEE/ACM International Symposium on Microarchitecture*, San Diego, CA, December 2003.

- Franceschetti, M. O. Dousse, D. Tse, and P. Thiran, "Closing The Gap In The Capacity Of Random Wireless Networks," *IEEE International Symposium on Information Theory* (ISIT '04), Chicago, Illinois.

- Franceschetti, M., "Power Delay Profile In A Cluttered Environment," *IEEE International Conference on Communications* (ICC '04), Paris, France.

- Franceschetti, M., "Stochastic Rays: The Cluttered Environment," *International Conference on Electromagnetics in Advanced Applications* (ICEAA '03), Turin, Italy.

- Franceschetti, M., "Phase Transitions, An Engineering Perspective," *Allerton Conference on Communication Computing and Control* (Allerton '03), Monticello, Illinois.

- Franceschetti, M., "Stochastic Rays Propagation," *Allerton Conference on Communication Computing and Control* (Allerton '03), Monticello, Illinois.

- Ghosal, Arkadeb, Thomas A. Henzinger, Christoph M. Kirsch, and Marco A.A. Sanvido, “Event-Driven Programming With Logical Execution Times,” Proceedings of the Seventh International Workshop on Hybrid Systems: Computation and Control (HSCC), Lecture Notes in Computer Science 2993, Springer-Verlag, 2004, pp. 357-371.

- Henzinger , Thomas A., Christoph M. Kirsch, and Slobodan Matic, “Schedule Carrying Code,” *EMSOFT*, Philadelphia, PA October 12-15, 2003.

- Jianghai Hu, Wei Chung Wu, and Shankar Sastry, "Modeling Subtilin Production in Bacillus subtilis Using Stochastic Hybrid Systems," The *7th International Workshop on Hybrid Systems: Computation and Control*, Philadelphia, PA, vol. 2993 of Lecture Notes in Computer Science, pp. 417-431, Springer-Verlag, 2004.

- Karsai, G., "Automotive Software: A Challenge and Opportunity for Model-based Software Development," to appear in *LNCS* volume on *Automotive Software Development Workshop*, San Diego, January, 2004

- Karsai, G., Agrawal A., Ledeczi, A., "A Metamodel-Driven MDA Process and its Tools," *WISME, UML 2003 Conference*, San Francisco, CA, October 2003

- Karsai, G. Lang, A., Neema, S., "Tool Integration Patterns," *Workshop on Tool Integration in System Development, ESEC/FSE*, pp 33-38, Helsinki, Finland, September 2003

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

- Lee, Edward A., "Model-Driven Development - From Object-Oriented Design to Actor-Oriented Design," extended abstract of an invited presentation at *Workshop on Software Engineering for Embedded Systems: From Requirements to Implementation* (a.k.a. The Monterey Workshop) Chicago Sept.24, 2003.

- Neema, S., Sztipanovits, J., Karsai, G., "Constraint-Based Design-Space Exploration and Model Synthesis," *EMSOFT*, Philadelphia, PA October 12-15, 2003.

- Neuendorffer, Stephen and Edward A. Lee, "Hierarchical Reconfiguration of Dataflow Models," *Conference on Formal Methods and Models for Codesign* (MEMOCODE), June 22-25, 2004, San Diego, CA, USA.

- Pinello, C., L.P. Carloni, and A.L. Sangiovanni-Vincentelli, "Fault-Tolerant Deployment of Embedded Software for Cost-Sensitive Real-Time Feedback-Control Applications," The *Proceedings of the Conference on Design, Automation and Test in Europe* (DATE), 2004.

- Plishker, William, Kaushik Ravindran, Niraj Shah, and Kurt Keutzer, "Automated Task Allocation on Single Chip, Hardware Multithreaded, Multiprocessor Systems," *Workshop on Embedded Parallel Architectures* (WEPA-1), February, 2004.

- Schmidt, P., I. Amundson and 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. To be presented at the *147th Meeting of the Acoustical Society of America*, New York, May 24-28, 2004

- Shah, Niraj, William Plishker, Kurt Keutzer. Comparing Network Processor, "Programming Environments: A Case Study," *2004 Workshop on Productivity and Performance in High-End Computing* (P-PHEC), February, 2004.

- Sinopoli, B., L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE International Conference on Control Decision and Systems* (CDC '03), Maui, Hawaii.

- Solar-Lezama, Armando, Rastislav Bodik, "Templating Transformations for Bitstream Programs," *HPCA Workshop on Productivity and Performance in High-End Computing* (P-PHEC 2004), held in conjunction with HPCA 2004, Madrid, Spain.

- Szemethy, T., G. Karsai, "Platform modeling and model transformations for analysis," *IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS* 2004

- Sztipanovits, J., "Design Space Construction and Exploration: A Model Integrated Computing Approach," *MPSOC 2003*, Chamonix, France, June 8, 2003 (presentation in Workshop Proceedings)

- Sztipanovits, J., "Model-Integrated Computing Infrastructure for Fault Management," in *Proc. Of DX-14, 14th International Workshop on Principals of Diagnosis*, Washington DC, June 12, 2003

- Sztipanovits, J., Neema, S., Chen, K., "Model-Integrated Computing for Automotive Applications," *Automotive Software Workshop* San Diego, CA, January 2004, *LNCS*, to appear

- Xu, Min, Rastislav Bodik, Mark Hill, "A "Flight Data Recorder" for Enabling Full-system Multiprocessor Deterministic Replay," The 30th International Symposium on Computer Architecture, San Diego, CA, June 2003.

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

- Agrawal, A., Simon, G. Karsai, G., "Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT," *Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT)* 2004. To appear in *Electronic Notes on Theoretical Computer Science*, Elsevier

- Baldwin, Philip, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, "VisualSense: Visual Modeling for Wireless and Sensor Network Systems," Technical Memorandum UCB/ERL M04/08, University of California, Berkeley, CA 94720, USA, April 23, 2004.

- Brooks, Christopher Hylands and Edward A. Lee, "Ptolemy II Coding Style" Technical Memorandum UCB/ERL M03/44, University of California at Berkeley, November 24, 2003.

- Carloni, Luca, Maria Domenica DiBenedetto, Alessandro Pinto and Alberto Sangiovanni-Vincentelli, “Modeling Techniques, Programming Languages, and Design Toolsets for Hybrid Systems,” Columbus Report IST-2001-38314 WPHS.

- Cataldo, A., C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, H. Zheng "HyVisual: A Hybrid System Visual Modeler," Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, CA 94720, July 17, 2003 (earlier version, January, 2003).

- Cataldo, Adam, "Control Algorithms for Soft Walls," Master's Report, Technical Memorandum UCB/ERL M03/42, University of California, Berkeley, CA 94720, January 21, 2004.

- Cheong, Elaine and Jie Liu, "galsC: A Language for Event-Driven Embedded Systems," Technical Memorandum UCB/ERL M04/7, University of California, Berkeley, CA 94720, USA, 20 April 2004.

- Eker, Johan and Jorn W. Janneck, " CAL Language Report: Specification of the CAL actor language," Technical Memorandum No. UCB/ERL M03/48, University of California, Berkeley, CA, 94720, USA, December 1, 2003.

- Gray, J., J. Sztipanovits, T. Bapty and S. Neema, “Two-level Aspect Weaving to Support Evolution in Model-Driven Software,” in *Aspect-Oriented Programming*

- Harren, Matthew, and George Necula, "Lightweight Wrappers for Interfacing with Binary code in Ccured," In Proceedings of the 3rd International Symposium on Software Security (ISSS03), Tokyo, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II) ," Technical Memorandum UCB/ERL M03/27, University of California, Berkeley, CA USA 94720, July 16, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng, (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture) ," Technical Memorandum UCB/ERL M03/28, University of California, Berkeley, CA USA 94720, July 16, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)," TechnicalMemorandum UCB/ERL M03/29, University of California, Berkeley, CA USA 94720, July 16, 2003.

- Hylands, Christopher, Edward A. Lee, Jiu Liu, Xiaojun Liu, Stephen Neuendorffer, Haiyang Zheng, “HyVisual: A Hybrid System Visual Modeler,” Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, July 17, 2003.

- Karsai, G., Agrawal, A. “Graph Transformations in OMG’s Model-Driven Architecture,” to appear in *Lecture Notes in Computer* Science volume on *Applications of Graph Transformation with Industrial Relevance*, 2003.

- Karsai, G., S. Neema, D. Sharp, “Model-Driven Architecture for Embedded Software: A Synopsis and an Example,” To appear in *Science of Computer Programming* (Elsevier) on *Model Driven Architecture: Foundations and Applications Model Driven Architecture*.

- Kohli, Sanjeev, "Cache Aware Scheduling for Synchronous Dataflow Programs," Master's Report, Technical Memorandum UCB/ERL M04/03, University of California, Berkeley, CA 94720, February 23, 2004.

- Lee, Edward A., "Soft Walls: Frequently Asked Questions," Technical Memorandum UCB/ERL M03/31, University of California, Berkeley, CA 94720, July 21, 2003.

- Lee, Edward A., "Overview of the Ptolemy Project," Technical Memorandum No. UCB/ERL M03/25, University of California, Berkeley, CA, 94720, USA, July 2, 2003.

- Lee, Edward A. and Stephen Neuendorffer, "Actor-oriented Models for Codesign," In Sandeep Shukla and Jean-Pierre Talpin editors, *Formal Methods and Models for System Design*, Kluwer, 2004. To appear.

- Neuendorffer, Stephen and Edward A. Lee, "Hierarchical Reconfiguration of Dataflow Models," Technical Memorandum UCB/ERL M04/2, University of California, Berkeley, CA 94720, USA, January 2004.

- Neuendorffer, Stephen, "Implementation Issues in Hybrid Embedded Systems," Technical Memorandum No. UCB/ERL M03/22, University of California, Berkeley, CA, 94720, USA, June 24, 2003.

- Roberto Passerone, *Semantic Foundations for Heterogeneous Systems*, PhD dissertation, Department of EECS, University of California at Berkeley, May 2004.

- Weimer, Westley, and George Necula, "Finding and Preventing Run-Time Error Handling Mistakes", In Proceedings of the Object-Oriented Programming Systems, Languages and Applications (OOPSLA04), Vancouver, 2004.

- Yeh, James, " Image and Video Processing Libraries in Ptolemy II," Master's Report, Technical Memorandum No. UCB/ERL M03/52, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

- Zhao, Yang, " A Model of Computation with Push and Pull Processing," Master's Report, Technical Memorandum No. UCB/ERL M03/51, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

- Zhou, Ye, "Communication Systems Modeling in Ptolemy II," Master's Report, Technical Memorandum No. UCB/ERL M03/53, University of California, Berkeley, CA, 94720, USA, December 18, 2003.

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.

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.

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 [69].

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:

- HyVisual 3.0 and 4.0-alpha, beta, a block-diagram editor and simulator for continuous-time and hybrid systems. 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 is a packaged subset of Ptolemy II.

- The Giotto and xGiotto systems are a programming methodology for embedded control systems running on possibly distributed platforms. Giotto and xGiotto are programming languages that allow the implementation of deterministic and analyzable real-time systems. While Giotto restricts the programming model to be time-triggered, xGiotto allows it to be event driven. Both the languages use the logical execution time paradigm for tasks to achieve program determinism. This paradigm restricts the execution time of task to be deterministic and platform independent. Giotto software can be accessed at: http://www-cad.eecs.berkeley.edu/~giotto/.

- VisualSense 4.0-alpha, beta: a visual editor and simulator for wireless sensor network system. Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, 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 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 is a packaged subset of Ptolemy II.

- Ptolemy II 3.0.2, and 4.0-alpha, beta. Ptolemy II 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, dataflow, process networks, synchronous/reactive systems, and communicating sequential processes. Ptolemy II includes a number of support packages, such as graph, providing graph-theoretic manipulations, math, providing matrix and vector math and signal processing functions, plot, providing visual display of data, data, providing a type system, data encapsulation and an expression parser. Ptolemy II is available at http://ptolemy.eecs.berkeley.edu/ptolemyII.

- Chic 1.1, an interface compatibility checking framework for hardware and software components interacting with each other and an environment. Chic supports component-based specification of behavioral assumptions and guarantees with respect to, for example, static and/or dynamic constraints on input and output values, method call patterns, resource consumption, etc. The tool then computes the minimal restrictions that must be followed by the environment to allow the known components to functions correctly with respect to some given safety properties. If no such environment exists, the tool correctly concludes that the given set of components is not mutually consistent, or compatible. This framework thus provides a static methodology for early design-time error detection for component-based systems. Chic is available at http://www.eecs.berkeley.edu/~tah/Chic/.

- GalsC, a language and compiler for use with the TinyGALS programming model, which uses TinyOS as the underlying component model. GalsC is compatible with TinyOS 1.x and nesC 1.1.1. You can use the galsC compiler to compile all your existing TinyOS programs. TinyGALS is a globally asynchronous, locally synchronous model for programming event-driven embedded systems, especially sensor networks. At the local

level, software components communicate with each other synchronously via method calls. Components are composed to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors without explicitly passing messages. The TinyGALS programming model is structured such that code for all inter-actor communication, actor triggering mechanisms, and access to guarded global variables can be automatically generated from a high level specification. By raising concurrency concerns above the level of TinyOS components, the TinyGALS programming model allows programmers to focus on the main tasks that the application must execute. Programs developed using this task-oriented model are thread safe and easy to debug.

- Nc2momllib: This tool is an extension of the nesC compiler. nesC is "an extension to the C programming language designed to embody the structuring concepts and execution model of TinyOS. 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, described at http://webs.cs.berkeley.edu/tos/, is used, for example, on the Berkeley MICA "motes," which are small wireless sensor nodes. Nc2momllib is used to convert nesC files (.nc) into MoML files (.xml). This will create the Ptolemy II libraries of components that are used to assemble models. TinyOS provides a rich library of nesC components.

- GME is a metaprogrammable visual modeling environment that allows the metamodeling of new domain-specific modeling languages (DSML-s). The metamodeling is done using UML/OCL. Once the metamodels are compiled, they can be used to instantiate the GME tool to support the abstract syntax and static semantics of the specified DSML. GME is available via http://www.isis.vanderbilt.edu/Projects/gme/.

- GREAT is a toolsuite for building model transformation tools using graph transformation techniques. The transformations are expressed in the GREAT language, which supports the visual specification in terms of the metamodel of the input and the output of the transformations. The tool suite includes a (GME-based) modeling language, a virtual machine that directly executes the transformations, a code generator that compiles transformations into C++ code, and a debugger that allows debugging of transformation programs. GREAT is available from http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#GREAT

- SMOLES: As reported in a paper for the FSCBS 2004 workshop, we have developed a technique for modeling platforms of embedded systems and using these models to perform analysis on the models. We have designed a simple modeling language for embedded systems (SMOLES) that supports a version of the dataflow paradigm. The language has been used in the 2003 SIPHER program by undergraduates to develop small-scale embedded applications. Next we have developed a technique for transforming SMOLES models into timed automata (T/A) models that are equivalent representations for the applications. These T/A models can then be analyzed using a standard analysis tool (Uppaal) to determine the timing properties of the application. The main contribution

of this work was the generative approach to create the model of the run-time (dataflow) kernel using graph transformation techniques

4. Contributions

This section summarizes the major contributions during this reporting period.

4.1. Within Discipline

4.1.1. Hybrid Systems Theory

- We introduced a game-theoretic view of system models that is compositional, and interestingly, not a zero-sum game. We have identified and studied a special kind of Nash equilibrium for such games.

- We have collaborated with our European colleagues to conduct a systematic and detailed survey of hybrid systems modeling tools.

- We have developed a theory for composing robust models of systems where instead of traces having a hard distinction between "possible" and "impossible," the distinction is graduated, indicating a "distance" to a possible trace.

- We have developed algorithms for computing the real value of discounted properties, which are continuous values that replace discrete, brittle, Boolean-valued property satisfaction, expressed in temporal logic over state transition systems.

- We have developed a theory of affine hybrid systems.

- We have improved on the best known algorithms for finding strategies for the control of stochastic hybrid systems.

- We have constructed a toolbox using ellipsoidal methods to calculate reach sets for linear dynamic systems.

- We have developed a deterministic operational semantics for hybrid systems simulations that deals systematically with zero-time events, simultaneous events, and discontinuities in piecewise continuous signals.

- We have obtained a stability result for stochastic hybrid systems and have applied them to studying biological systems.

4.1.2. Model-Based Design

- Applying our ongoing work on metamodeling, we have developed a metamodel for the abstract syntax of the Hybrid System Interchange Format (HSIF) and have used it in developing a translator between HSIF and Simulink/Stateflow.

- We have developed 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 developed a theory and compositional framework for reasoning about causality in components that 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 developed a language and a suite of supporting tools for the specification of model transformations based on graph rewriting.

- We have developed an approach to model synthesis based on patterns specified formally as meta models.

- We have extended the principles of the Giotto language beyond periodic time-driven systems to aperiodic event-driven systems, and we have developed a language called xGiotto supporting this approach.

4.1.3. Advanced Tool Architectures

- We have defined and prototyped modularity mechanisms (classes, subclasses, inheritance, interfaces) for actor-oriented design, complementing the prevailing object-oriented methods.

- 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 improved 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 have introduced a strong type system into our embedded virtual machine, and have separated scheduling functionality into a separate virtual machine architecture called the S machine.

- We have introduced the idea of event scoping in embedded software, and have defined an extended embedded virtual machine that supports it.

- We have implemented our embedded virtual machine architectures on KURT Linux and have developed a concept demonstration prototype of code generation from Ptolemy II for this target.

- We have formulated and solved the task allocation problem for a popular, multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor.

- We have developed an interactive tool called Prospector that is an extension of Eclipse supporting navigating through complex framework documentation.

- We have developed model checking algorithms for automata where states are labeled with natural-number valued quantities rather than Booleans, expressing for example power consumption and memory usage.

- We have developed a flow-sensitive static analysis tool that identifies failures of software to deal with outstanding obligations, for example closing open files when exception conditions occur.

- We have improved the Ccured static analysis tool for C programs to deal with library functions where source code is not available.

4.1.4. Experimental Research

- We have made progress on models for conflict detection for aircraft.

- We have improved our means for modifying the control software in fly-by-wire aircraft to restrict the airspace that an aircraft will fly into. The scheme is called Soft Walls.

- We have developed a modeling environment for wireless sensor networks.

- We have fundamental results on the connectivity of large-scale sensor networks.

- We have developed methods for dealing with safety critical distributed applications that include novel models of computation to capture safety specifications and synthesis algorithms that map optimally the requirements on a redundant architecture.

- We have deployed the Metropolis platform-based design methodology and the Metropolis environment for the solution of a Picture-in-Picture subsystem for HDTV as specified by one of our industrial partners.

- We have developed new programming models for sensor networks that build on the popular TinyOS models.

- We have developed programming models for bit streaming applications that use sketching of strategies with code generation.

- We have been adapting sensor networks technology to address elder care problems.

4.2. Other Disciplines

- We developed new efficient algorithms for solving stochastic games, which have applications in other fields such as economics.

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. 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. 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.

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. Elder care, soft walls and distributed diagnosis are a few examples of this impact. In the domain of transportation systems, our research is improving safety in cars. 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.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-00225610

UNIVERSITY OF CALIFORNIA AT BERKELEY
VANDERBILT UNIVERSITY
UNIVERSITY OF MEMPHIS

MAY 31, 2005

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2004 – MAY 31, 2005

Contents

Contents 2
1. Participants 5
1.1. People 5
1.2. Partner Organizations 8
1.3. Collaborators 8
2. Activities and Findings 9
2.1. Project Activities 9
2.1.1. Hybrid Systems Theory 9
2.1.1.a. Deep Compositionality 10
Non-Zero-Sum Games as Compositional System Models *10*
Hybrid Systems Modeling Tools: a Survey *10*
2.1.1.b. Robust Hybrid Systems 11
Design and Verification of Robust System Models *11*
Affine Hybrid Systems *11*
A Homology Theory for Hybrid Systems *11*
2.1.1.c. Computational Hybrid Systems 11
Algorithms for the Control of Stochastic Systems *11*
Reach Set Calculations using Ellipsoidal Approximations *11*
A Deterministic Operational Semantics for Hybrid System Simulations *12*
2.1.1.d. Stochastic Hybrid Systems 12
Application of Stochastic Hybrid Systems to Communication Networks *13*
Application of Stochastic Hybrid Systems in Systems Biology. *13*
2.1.2. Model-Based Design 13
2.1.2.a. Composition of Domain Specific Modeling Languages 14
Metamodeling *14*
Compositional Metamodeling *15*
Integration of Metamodeling with Hybrid Systems *15*
Semantic Foundations for Heterogeneous Systems *15*
Generalized causality analysis *16*
2.1.2.b. Extensions to Distributed Models of Embedded systems 16
Compositional Theory of Heterogeneous Reactive Systems *16*
2.1.2.c. Model Transformation 17
Meta Generator Technology *17*
Pattern-Based Model Synthesis *17*
2.1.2.d. Real-Time Programming Models 18
Event-Triggered Programming *18*
2.1.3.a. Syntax and Semantics 20
Modularity Mechanisms in Actor-Oriented Design *20*
Code Generation from Actor-Oriented Models *20*
Metropolis Framework *20*
2.1.3.b. Interface Theories 21
A Component Model for Heterogeneous Systems *21*
Interface Modeling and Models of Computation *22*

2.1.3.c. Virtual Machine Architectures 22
Types for Real-Time Programs *22*
Separating Reactivity from Schedulability *22*
Implementing Event Scoping *23*
Real-Time Software using Ptolemy-II, Giotto, and the E and S Virtual Machines *23*
2.1.3.d. Components for Embedded Systems 23
Mapping Network Applications to Multiprocessor Embedded Platforms *23*
2.1.3.e Verification of Embedded Software 23
Model Checking Quantitative Properties of Systems *23*
Run-Time Error Handling *23*
Memory Safety Enforcement: *24*
2.1.4 Experimental Research 24
2.1.4.a. Embedded Control Systems 24
Automated Landing for Unmanned Aerial Vehicles (UAVs) *24*
2.1.4.b. Embedded Software for National and Homeland Security 25
Aerial Pursuit Evasion Games for Aircraft *25*
Distributed Control for Networks of Unmanned Underwater Vehicles (UUVs) *25*
Softwalls for Collision Avoidance *25*
Shooter Localization in Urban Terrain *25*
2.1.4.c. Networks of Distributed Sensors 25
VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems *25*
Large Scale Sensor Networks *26*
Programming Models for Sensor Networks *27*
Control of Communication Networks *27*
Elder Care *28*
2.1.4.d. Fault-Driven Applications 28
Distributed Diagnosis of Complex Physical Systems *28*
On-line Hierarchical Fault Adaptive Control *29*
Safety-Critical Distributed Applications *29*
2.2. Project Findings 30
2.3. Project Training and Development 70
2.4. Outreach Activities 70
2.4.1. Curriculum Development for Modern Systems Science (MSS) 71
Undergrad Course Insertion and Transfer 71
Course: Structure and Interpretation of Signals and Systems (UCB, EECS 20N) *72*
Graduate Courses 72
Course: Concurrent Models of Computation for Embedded Software (UCB, EECS 290N) *72*
Course: Hybrid Systems: Computation and Control (UCB, EECS 291E/ME 290S) *73*
Course: Theory of Automata, Formal Languages, and Computation (VU, CS 352) *73*
Course: Foundations of Hybrid and Embedded Systems (VU, CS 376) *73*
Course: Control Systems I (VU, EECE 257-01) *73*
Course: Model Integrated Computing (VU, CS 388 / EE 395) *73*
Course: Real-Time Systems (VU, EECE 353-01) *74*
Course: Automated Verification (VU, EECE 315) *74*
Course: Automated Verification (VU, EECE 375) *74*

2.4.2. SUPERB-IT Program 74
Project: Routing Protocols for Wireless Sensor Networks *75*
Project: Multi-view Configuration of Flight Dynamic Playback *75*
Project: Singular Event Detection *76*
Plans for 2005 76
2.4.3. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program 77
Project: Distributed System Implementation of the Kuvangu Running Frog's Calling Behavior *78*
Project: Maze Discovery and Chutes & Ladders *78*
Project: Tic-Tac-Toe Implementation *79*
Project: A Model-Integrated Computing Approach to Embedded Controller Design . *79*
Project: Visual Tracking and Control *79*
Plans for 2005 80
3. Publications and Products 81
3.1. Journal Publications 81
3.2. Conference Papers 82
3.3. Books, Reports, and Other One-Time Publications 85
3.4. Dissemination 86
3.5. Other Specific Product 87
4. Contributions 88
4.1. Within Discipline 88
4.1.1. Hybrid Systems Theory 88
4.1.2. Model-Based Design 89
4.1.3. Advanced Tool Architectures 89
4.1.4. Experimental Research 90
4.2. Other Disciplines 91
4.3. Human Resource Development 91
4.4. Integration of Research and Education 91
4.5. Beyond Science and Engineering 92

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, ELECTRICAL AND COMPUTER ENGINEERING)

FACULTY INVESTIGATORS:

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

POST DOCTORAL RESEARCHERS:

MASSIMO FRANCESCHETTI (UC BERKELEY)
CHRISTOPH 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 (VANDERBILT)

LUCA CARLONI (UC BERKELEY)
ADAM CATALDO (UC BERKELEY)
DENNIS CHANG (UC BERKELEY)
KAI CHEN (VANDEREBILT)
ELAINE CHEONG (UC BERKELEY)
ABHIJIT DAVARE (UC BERKELEY)
DOUG DENSMORE (UC BERKELEY)
ANDREW D. DIXON (VANDERBILT)
BRANDON EAMES (VANDERBILT)
MATTHEW J. EMERSON (VANDERBILT)
HUINING FENG (UC BERKELEY)
JOYTI GANDHE (VANDERBILT)
SUMITRA GANESH (UC BERKELEY)
CSABA GARAY (VANDERBILT)
MATTHEW HARREN (UC BERKELEY)
ETHAN JACKSON (VANDERBILT)
FARINAZ KOUSHANFAR (UC BERKELEY)
NARAYANAN KRISHNAN (UC BERKELEY)
ALEXANDER KURZHANSKIY(UC BERKELEY)
JONGHO LEE (UC BERKELEY)
XIAOJUN LIU (UC BERKELEY)
GABOR MADL (VANDERBILT)
DAVID P. MANDELIN (UC BERKELEY)
ELEFTERIOUS MATSIKOUDIS (UC BERKELEY)
MARK MCKELVIN (UC BERKELEY)
BHARATHWAJ MUTHUSWAMY (UC BERKELEY)
STEPHEN NEUENDORFFER (UC BERKELEY)
SONGHWAI OH (UC BERKELEY)
WILLIAM PLISHKER (UC BERKELEY)
KAUSHIK RAVINDRAN (UC BERKELEY)
PANNAG SANKETI (UC BERKELEY)
PETER SCHMIDT (VANDERBILT)
TIVADAR SZEMETHY (VANDERBILT)
GUANG YANG (UC BERKELEY)
HAIBO ZENG (UC BERKELEY)
YANG ZHAO (UC BERKELEY)
HAIYANG ZHENG (UC BERKELEY)
WEI ZHENG (UC BERKELEY)
GANG ZHOU (UC BERKELEY)
YE ZHOU (UC BERKELEY)

QI ZHOU (UC BERKELEY)

UNDERGRADUATE STUDENTS:

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:

CHRISTOPHER HYLANDS BROOKS (UC BERKELEY)
NATHAN JEW (UC BERKELEY)
BRADLEY A. KREBS (UC BERKELEY)
PHILLIP LOARIE (UC BERKELEY)
MARVIN MOTLEY (UC BERKELEY)
GUNNAR PROPPE (UC BERKELEY)
MARY STEWART (UC BERKELEY)

AARON WALBURG (UC BERKELEY)
BRIAN WILLIAMS (VANDERBILT)

BUSINESS ADMINISTRATORS:
SUSAN B. GARDNER (UC BERKELEY)
ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)

1.2. Partner Organizations

- University of California at Berkeley
- Vanderbilt University
- University of Memphis

1.3. Collaborators

- Albert Benveniste (IRISA INRIA, Rennes)
- 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)
- Kim 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 second Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software". This research activity is primarily organized through a Center at Berkeley CHESS (the Center for Hybrid and Embedded Systems and Software, http://chess.eecs.berkeley.edu), the Vanderbilt ISIS (Institute for Software Integrated Systems, http://www.isis.vanderbilt.edu), and the Department of Mathematical Sciences, (http://msci.memphis.edu) at 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 third year were:

1. NSF Onsite Review, November 18th, 2004, UC Berkeley. The program and the presentations are available at http://chess.eecs.berkeley.edu/conferences/04/NSFonsiteReview11-04
2. The annual Chess Review for our industrial partners, advisory board members, and friends of the project. The program and presentations are available at http://chess.eecs.berkeley.edu/conferences/04/05/index.htm
3. A weekly Chess workshop was held at Berkeley. Presentations for the workshop are available at http://chess.eecs.berkeley.edu/workshop.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, in the past year we designed and implemented 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

Non-Zero-Sum Games as Compositional System Models

The components of a complex system can be viewed as players in a game with different objectives. Each component attempts to satisfy its own specification, but the specifications of different components may be neither identical nor complementary; so the right formalization is that of a non-zero-sum games. Classically, Nash equilibria capture the notion of rationality for such games. We argue, however, that for achieving compositionality in system design, a special kind of Nash equilibrium is appropriate. 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 players' goals are not necessarily conflicting), and undiscounted. We show that if each player has an ω-regular objective expressed as a parity objective, then there exists an ϵ-Nash equilibrium, for every ε >0. However, exact Nash equilibria need not exist. We study the complexity of finding values (payoff profile) of an ε-Nash equilibrium. This work is reported in [23][24][25][26].

Hybrid Systems Modeling Tools: a Survey

Las year in a collaborative project with the European Columbus project, we evaluated a set of tools, languages and formalisms for the simulation, verification and specification of hybrid systems. In this survey we evaluated the following languages and tools: Simulink, Modelica, HyVisual, Scicos, Charon, CheckMate, Masaccio, SHIFT, HSIF, Metropolis and Hysdel. This year we continued this work with a sets of surveys by Lee on the state of embedded systems, formal versus informal methods and concurrent models of computation for embedded software. These survey results are reported in [49][50][51].

.

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. This work is reported in [25].

Affine Hybrid Systems

Affine hybrid systems are hybrid systems where the discrete domains are affine sets, and the transition maps between discrete domains are affine transformations. This definition differs from other definitions of hybrid systems that have been proposed, but the underlying ideas involved in the definition of affine hybrid systems have been seen in the literature. We give a formal framework to these ideas. Affine hybrid systems are simple, and it is this simplicity that allows us to say some useful things about them. The structure of affine hybrid systems contains a wealth of intrinsic information. Affine sets can be described in terms of matrix inequalities, and affine transformations are characterized by elements of SE(n). In the paper [10], we use the geometric information intrinsic in affine hybrid systems to develop the idea of *spatial equivalence* between an affine hybrid system **H** and an affine hybrid system **G.** More specifically we give conditions for what is known as blowing up affine hybrid systems.

A Homology Theory for Hybrid Systems

Using a categorical framework in [12] 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 [13] and it allows for the constructive methods of algebraic topology to give full characterizations of when hybrid systems do not admit of solutions on the infinite time horizon in [11].

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. This work is reported in [69].

Reach Set Calculations using Ellipsoidal Approximations

Linear dynamic systems are considered. We studied the application of the external ellipsoidal approximations of the reach sets to the internal point problem. An algorithm was developed that allows us to calculate control and initial state that bring the system to the given point at the given

time. A closed-form solution is obtained, or else, if the target point is unreachable, then the closest reachable point is found.The ellipsoidal toolbox provides the implementation of the ellipsoidal methods used for the computation of the reach sets. It includes external and internal ellipsoidal approximation algorithms as well as visualization routines. The API allows the user to run the algorithms with given precision. Most recently this work has been continued to allow for reachability approximations for hybrid systems. The work by Kurzhanskiy and Varaiya is in progress.

A Deterministic Operational Semantics for Hybrid System Simulations

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 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 [17]

2.1.1.d. Stochastic Hybrid Systems

Stochastic hybrid systems are a natural extension of the deterministic counterpart. We also develop strategies for characterizing the stability of stochastic hybrid systems in [5]. Additionally we have used stochastic hybrid systems as approximations to simulation models of deterministic hybrid systems with non-deterministic switching [1].

Application of Stochastic Hybrid Systems to Communication Networks.

In [2][3] the objective of is to use the theory of stochastic hybrid systems to introduce two original flow control schemes for wireless networks. The mathematical underpinnings lie on the recently-developed congestion control models for Transmission-Control-Protocol(TCP)-like schemes; more precisely, the model proposed by Kelly for the wired case is taken as a template, and properly extended to the more involved wireless setting. We introduce two ways to modify a part of the model; the first is through a static law, and the second via a dynamic one. In both cases, we prove the global stability of the schemes, and present a convergence rate study and a stochastic analysis.

Application of Stochastic Hybrid Systems in Systems Biology.

Last year, we applied the stochastic hybrid system to the modeling of genetic regulatory network. This modeling approach can demonstrate the inherent randomness in molecular interactions and protein production in a cell, phenomena that are observed in biological systems but are not realized adequately using conventional macroscopic modeling techniques. A stochastic hybrid system approach was used to model the genetic network regulating the biosynthesis of an antibiotic called subtilin in Bacillus subtilis. This year we have continued this work in determining the behavior of other more complex mechanisms for regulation of other signaling mechanisms and have begun the study of what may be called stochastic resonance in biological systems.

2.1.2. Model-Based Design

Model-based design focuses on the formal representation, composition, and manipulation of models during the design process. It addresses system specification, model transformation, synthesis of implementations, model analysis and validation, execution, and design evolution. The semantic frameworks in which these models are applied may be domain-specific, offering embedded system designers methods and syntaxes that are closer to their application domain. 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. See for example the work reported in [78].

Our extensive experimental work on networked embedded systems (see applications and papers [39][43][59][68] 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 [29].

2.1.2.a. Composition of Domain Specific Modeling Languages

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. See for example [44].

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 [63]) 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. See

also [37][38]. Additional work that was begun this year is on doman specific visual languages for domain model evolution (see [76]).

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 [37][38] for the new results here.

Integration of Metamodeling with Hybrid Systems

We have developed a metamodel for the abstract syntax of the Hybrid System Interchange Format (HSIF). This work, which was part of our earlier projects, was lately extended with developing a translator between HSIF and Simulink/ Stateflow models This work helped us understanding the role of model transformations for defining semantics of DSML-s via the formal specification of translators. This work has been demonstrated successfully in a joint project with Northrup Grumann to successfully land UAVs (in a flight test in the Mojave Desert in July 2004), see [77].

Semantic Foundations for Heterogeneous Systems

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.

This work is reported in the recently completed PhD dissertation of Neuendorffer [64]. See also the survey paper of Lee [51] and two key reports of Lee and Neuendorffer and Zheng [54][55].

Generalized causality analysis

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 [49], [50].

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.

We have derived a set of theorems that support effective techniques to generate automatically correct-by-construction adaptors between designs formulated using different coordination paradigms [28]. We have applied these concepts to two scenarios that are of particular relevance for the design of embedded systems: the deployment of a synchronous design over a globally-asynchronous locally-synchronous (GALS) architecture and over a loosely time-triggered architecture (LTTA). The idea followed in these applications is to abstract away from the synchronous specifications the constraints among events of different signals due to the synchronous paradigm and, then, to map the unconstrained design into a different architecture characterized by a novel set of intended behavior of the system is retained. This is achieved by relying on a formal notion of semantics-preserving transformations that we developed based on the idea of morphisms over tag sets. We have introduced GALSC a language designed for programming event driven embedded systems such as sensor networks. GALSC implements the Tiny GALS programming model and has been implemented on the Berkeley motes and is compatible with the TinyOS/nesC component library.

2.1.2.c. Model Transformation

Meta Generator Technology

We have developed a language and a suite of supporting tools for the formal, high-level, yet executable specification of model transformations (GREAT (Graph Rewriting and Transformations) The language is based on graph transformation technology, where a complex model transformation is specified in terms of sequenced graph rewriting steps consisting of rewriting rules. This high-level specification facilitates not only the compact, formal representation of what a model translator should do, but also allows formal reasoning about the transformations. The semantics of GREAT has been formally defined in previous year's work and it comes with a set of supporting tools. See also some additional contributions this year in [37][38]. The tools suite includes an interpreter for GREAT ("Meta-Programmable Transformation Tool"), a code generator for compiling GREAT rules into C++, and a debugger (coupled with the interpreter). This year we have been using this to verify distributed real time properties of embedded systems in [58].

Pattern-Based Model Synthesis

We have introduced patterns in modeling on two levels. First, our meta-model composition technology enables to define abstract metamodeling constructs such as *templates* and *hierarchy* as language design patterns and compose those with DSML . Second, using the *template* construct, we are able to build large design spaces centered around domain architecture and automatically synthesize models that satisfy user defined constraints. The graph transformation formalism also enables the introduction of reusable idioms and patterns in model transformations We plan to further explore this opportunity in our future research. A new direction in model synthesis is aspect weaving. We have experimented with the development and application of

model weaving technology for generating complex, integrating models by merging different modeling aspects according to formally represented rules. This work has not yet been published.

2.1.2.d. Real-Time Programming Models

Event-Triggered Programming

In previous work, we had developed a time-triggered language, called Giotto, based on the Logical Execution Time (LET) model. In Giotto, software tasks are invoked and terminated strictly by clock events. This achieves deterministic behavior, which is of paramount importance in safety-critical systems. Last year, we extended the LET model to non-clock events, and called the resulting language xGiotto. This language permits the invocation and termination of tasks by arbitrary events in a way that still maintains determinism. This year we applied Giotto to the control of Unmanned Aerial Vehicles. This work has been completed but it has yet to be published.

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 meta modeling, 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 [45][46].

Conceptual concurrence between our frameworks is evolving. Whereas previously GME had mastered meta modeling of abstract syntactic properties of actor-oriented models, today it is moving aggressively towards meta modeling of their semantic properties, focusing initially on the 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 meta modeling. 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 meta modeling 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.

This year a number of survey papers gave coverage to this suite of tools (see for example [34][37][51][52][54][57][62][72][76][78][81]).

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.

Actor-oriented languages, like object-oriented languages, are about modularity of software. We argue that we can adapt for actor-oriented languages many (if not all) of the innovations of OO design, including concepts such as the separation of interface from implementation, strong typing of interfaces, subtyping, classes, inheritance, and aspects. We have realized some preliminary implementations of these mechanisms in Ptolemy II and published them in [17][34].

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. See [18][19][20].

Metropolis Framework

Features and efficiency are critical factors for simulators, which are still the dominating tools for design validation. During the past year, we improved the Metropolis simulator in several ways (see [32][33]):

1. *Add quantity annotation support*. Quantities can be used to model various kinds of performance indices, such as time, and power. They can also be generalized to model scheduling policies. Both of the usages usually appear in architecture models. A simulator's job is to ensure the quantity annotation requests are made at the proper time, and quantity resolution algorithms are executed when necessary to reach a fixed point solution.

2. *Add mapped behavior simulation support*. This capability is essential to evaluate behavior-architecture mappings, which is the key idea to explore design space in platform-based design. With this feature, a user's behavioral model can run simultaneously with a particular architecture under evaluation. Then, performance of the architecture running this behavior can be determined by simulation. During simulation, not only the events in both behavior and architecture models need to be synchronized, but also values need to be passed between. Both requirements present challenges to efficient simulation.

3. *Improve simulation efficiency*. In today's design, both behavior models and architecture models can be huge; combining them only makes the scale problem worse. We applied several techniques to improve the Metropolis simulator's performance. Simulation results show orders of magnitude performance improvement. The optimization techniques include named event reduction, a medium-centric simulation algorithm, mapped behavior handling with equivalent classes, efficient hierarchy traversal and an interleaving-concurrency-based simplification.

4. The current implementation of Metropolis simulator is based on the SystemC simulation kernel, which is an interleaving concurrent single kernel process environment. We detached the Metropolis simulator from this foundation and realized the parallelism with the pthreads library, which gives kernel-level processes (true concurrent) on some SMP machines. The preliminary experiments show sub-linear speed up on SMPs. Experiments show sub-linear speed up on the SMPs.

2.1.3.b. Interface Theories

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.

The component model also makes it easier to study certain formal properties of the components. For example, we can classify discrete event components as either reactive or proactive. A DE component is reactive if all output events and state changes are triggered by input events and proactive otherwise. From the output and next state functions of a DE component, we can derive the relations among the time stamps of its input and output signals, and use these relations to analyze whether a DE composite component is reactive. Current work includes defining a behavioral type system based on pattern matching as the component programming model. See the papers [32][60][61][33].

Interface Modeling and Models of Computation

One research effort on interface modeling was centered around the fundamental principles of interface theories and the evaluation of their pragmatic impact on component-based designs. We have constructed a general experimentation framework within Ptolemy II, and integrated the Checker for Interface Compatibility (Chic) tool into Ptolemy II. We have experimented with the various supported formalisms and evaluated their applicability to the wide spectrum of computational models supported in Ptolemy II. The overall outcome stressed the inherent subjectivity of interface theories, and suggested a more model-of-computation oriented approach in their development.

An extended effort has been made towards the specification of behavioral types at the dynamic interaction level through interface automata[15]. We focused on augmenting the interface automata theory in order to handle explicitly more elaborate forms of concurrency. Based on a variant of interface automata we were able to develop an assume guarantee reasoning for mutual exclusion in open systems. In the process of generalizing our results, we revealed the fundamental limitations of the interface automata theory associated with the inherent computational capacity of finite automata.

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. A report has been submitted for publication on the use of embedded machine code for time triggered controllers for UAVs.

Separating Reactivity from Schedulability

We implemented two virtual machines, one to react to environment events (the E or Embedded machine), and the other to react to CPU events (the S or Scheduling Machine), and their interaction. This architecture allows great flexibility (portability, extensibility, and composibility) in the implementation of reactive software.

Implementing Event Scoping

We extended the E (Embedded) Machine, a virtual machine with reactive real-time behavior, to accommodate the event scoping mechanism of xGiotto. Publication is still pending.

Real-Time Software using Ptolemy-II, Giotto, and the E and S Virtual Machines

We have created a software infrastructure for designing hard real-time systems using Ptolemy II, Giotto, and an implementation of the E and S machines on KURT Linux (from University of Kansas). We use the Giotto domain within Ptolemy II, which offers a graphical syntax for Giotto models. Systems with periodic tasks are specified with a timing resolution of milliseconds. The first stage of our implementation takes a graphical model as an input and generates C code, E code, and S code. The second stage is an implementation of the Embedded Machine interpreter, which reads in the E Code and interprets it to release the tasks for execution as per the timing requirements stated. The released tasks can either be assigned to a standard scheduler such as EDF, or the scheduling can be preformed by our implementation of the Scheduling Machine, an interpreter for S code. See also [47].

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 paper [66].

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. A report on this work has been submitted for publication

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[18][19][20]. Among the systems that we have debugged with our tool is the Ptolemy software.

Memory Safety Enforcement:

In previous work we have developed Ccured, a static analysis tool for C programs. CCured discovers for each pointer what kind of run-time checks, if any, are necessary in order to ensure memory safe execution. On average, Ccured discovers that 80% of the pointers used in a typical C program do not require any run-time checks. However, CCured has serious difficulties for programs that use libraries whose source is not available. For those libraries, CCured must not only make conservative assumptions about how they manipulate pointers, but must also refrain from changing the run-time representation of pointers and objects they point to. These difficulties arise in all static analysis and instrumentation systems. We have developed a solution based on user-defined polymorphic wrapper functions. These wrappers act both as a proxy for the missing library source code for the purpose of the static analysis, and also as the basis for generating code that changes the representation of pointers at the library boundary, to allow the co-existence of Ccured-controlled pointers and standard backwards-compatible pointers for the library objects. See [62] and [42].

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 Northrup Grumann to devise and actually fly (on a surrogate UAV, a Boeing T-35 with special avionics to allow it to be autonomous) algorithms for automated landing of UAVs with special contingency maneuvers for waving off landing. This work is reported in [73]

2.1.4.b. Embedded Software for National and Homeland Security

Aerial Pursuit Evasion Games for Aircraft

We have used techniques from stochastic hybrid systems along with templates of aerial combat techniques to build model predictive controllers (a form of approximate Hamilton Jacobi approximation controllers for safe operation) for enabling a UAV to participate in air to air combat. This work in partnership with Boeing Phantom Works was flown as a contest between a T-35 surrogate UAV and a human flown F-15E in the Mojave desert in July 2004. 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. See[74].

Distributed Control for Networks of Unmanned Underwater Vehicles (UUVs)

A decentralized control scheme for large packs of UUVs with communication constraints has been proposed and tested at our water tank facility at Richmond Field Station. See [36].

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. See also [4].

Shooter Localization in Urban Terrain

The world's best localization algorithms for sensor networks have been developed this year at Vanderbilt and have been used to efficiently localize shooters in an urban environment (see [59]). This work is being transitioned by Raytheon Systems, Inc.

2.1.4.c. Networks of Distributed Sensors

VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems

Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. VisualSense is a software framework designed to support a component-based construction of such 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.

Large Scale Sensor Networks

We looked at the problem of routing in large scale sensor networks, where only local connectivity information is used at each sensor node to decide the next-hop destination. The problem is studied in the context of small-world networks, providing a stochastic model that bridges between the well known 'six degrees of separation' property in social science and network routing protocols. Results are summarized in and have also been presented in different seminars and workshops. Some of this work began at Caltech and some under the DARPA NEST project, but it has been refined and revised under this project and is being carried forward under this project. Some of the references on this topic due to Bollobas and co-workers are not yet published and will be cited in the next report.

Another important issue in sensor networks regards how global connectivity is achieved in the network. Links are inherently unreliable and probabilistic connectivity models based on random connection models have been developed. In some recent work with Franceschetti, we obtain basic percolation results for different random connection models that account for the unreliability of the wireless channel.

The unreliability of the wireless channel is also the motivation for the work by Sinopoli, Franceschetti, Sastry et al (to be submitted to a special issue of the IEEE Proceedings), with the difference that in this case the effect on the control feedback loop is explicitly taken into account. The estimation problem is central for control systems (e.g. in pursuit evasion games with autonomous robots) and is based on the automatic refinement of the estimation as new observation data are collected in real time. When some of these data are lost due to the unreliability of the link the performance of such estimator obviously starts degrading. We explicitly characterize such degradation giving analytic bounds and showing a sharp transition to instability as the probability of link failure exceeds a given threshold. This is a substantial and first generalization to the centralized LQG paradigm of traditional control theory.

The data collection and distribution problems in sensor networks are studied in using a discrete mathematical model that allows to obtain basic limits on the time delay performance of different distribution and retrieval algorithms.

In [68] we addressed the basic problem of wireless signal propagation. Characterizing the physics of propagation is one of the key problems in wireless, as fundamental information theory limits strongly depend on the accuracy of the channel model. At the same time, this is a very complex task, due to complexity of the propagating environment that does not allow to solve Maxwell equations explicitly. Hence, often stochastic models based on few tunable parameters that can be analytically treated are employed. We proposed a stochastic model based on the theory of random walks and we obtained expressions for the path loss and power delay profile of a transmitted signal. The highlight of the research this year is the world's best localization algorithms for sensor networks developed by Vanderbilt (see for example [39]) and used in a number of applications such as shooter localization, see [59].

Distributed Control in Smart Structures

The control of Smart Structures presents a unique challenge for distributed control systems. The challenge is unique for two reasons: 1) the physics of Smart Structures are such that all nodes in a distributed sensing/control network will experience strongly connected dynamics and (2) the bandwidth of Smart Structures tends to be very high requiring high sampling rates and fast, efficient inter-node communications. In the past year we have begun experiments in the distributed control of Smart Structures experimental platforms that were constructed in the previous year. As noted, the primary challenges in these experiments center on (1) the design of distributed control systems that accommodate strongly connected inter-node dynamics and (2) the development of control systems and distributed network infrastructure that permit the relatively high sampling and communication rates needed. We have achieved some success in the distributed control development. However, the issue of sufficiently fast inter-node communication remains a bottle neck in performance. See [40][41].

Programming Models for Sensor Networks

We have created galsC [http://galsc.sourceforge.net][28], a language and compiler designed for use with the TinyGALS programming model, which uses TinyOS as the underlying component model. TinyGALS is a globally asynchronous, locally synchronous model for programming event-driven embedded systems, especially sensor networks. At the local level, software components communicate with each other synchronously via method calls. Components are composed to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors without explicitly passing messages. The TinyGALS programming model is structured such that code for all inter-actor communication, actor triggering mechanisms, and access to guarded global variables can be automatically generated from a high level specification. By raising concurrency concerns above the level of TinyOS components, the TinyGALS programming model allows programmers to focus on the main tasks that the application must execute. Programs developed using this task-oriented model are thread safe and easy to debug.

The original TinyGALS code generation toolset was designed to be compatible with software components written for TinyOS 0.6.1. The new implementation is designed to be compatible with TinyOS 1.x, which uses the nesC programming language. Our new language, galsC, extends the nesC language, which allows for better code generation and static analysis of programs. We have also redesigned the TinyGUYS mechanism to have better scoping. Having a well-structured concurrency model at the application level greatly reduces the risk of concurrency errors, such as deadlock and race conditions. These features are especially important in severely memory-constrained and safety-critical systems.

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, see [2][3][4], [30].

Elder Care

We have been experimenting with three axis accelerometer attached to a person and studying the sensitivity and robustness of these measurements for activities: Change from sitting to standing, Free fall, Walking, Running. We also experimented with two different locations where the accelerometer is attached to the human body: on the chest and on the waist. The data is preliminary but it suggests that the data analysis will have to be customized. We have performed some recent deployments of the technology in old age homes in Sonoma, Aarhus, Denmark and in Tampere, Finland. Experimental data is being gathered and we expect this to be a very important source of new ideas for high confidence medical devices and systems

Networks of Distributed Sensors

Distributed acoustic sensing has received a lot of attention in sensor network work. The possible applications include self-localization of sensors, target identification and tracking as well as environmental monitoring. We have developed a distributed acoustic sensing experimental platform based on PC-104 stack hardware (the same used in the Smart Structures control experiments, see [40]) and begun testing and evaluation. The primary objective is to develop sensing and sensor fusion algorithms that are distributed in nature (i.e. the computations are carried out in a parallel fashion among the sensor nodes) and without the support of a centralized controller or processor. Initial experiments have demonstrated that accuracies comparable to more sophisticated centralized solvers are possible (see [80]). We have also applied the results to the distributed control of thermoelectric coolers in [43].

2.1.4.d. Fault-Driven Applications

Distributed Diagnosis of Complex Physical Systems

The size and complexity of present day systems motivates the need for developing distributed fault diagnosis algorithms. This work extends the TRANSCEND qualitative framework for fault diagnosis in continuous dynamic systems, to develop a methodology that partitions the set of possible fault candidates in a physical system to independent sets of faults given a set of measurements. Separate diagnosers that do not interact with each other can be constructed for each set of independent faults while maintaining complete diagnosability of the system. Our approach is to partition the set of faults into subsets in such a way that we can construct independent diagnosers for each subset. Two diagnosers are independent if they do not have to share information in establishing unique diagnosis results that are globally valid. We establish this by ensuring that the two fault subsets corresponding to the two diagnosers do not require the same set of measurements to achieve complete diagnosability. Complete diagnosability is the ability to uniquely isolate every fault candidate in the system given a set of measurements. Although such system decomposition will not always be possible, our approach can be the first step for designing distributed diagnosers. The implication of this approach is that a large computationally expensive diagnosis task is decomposed into a set of smaller tasks that can be

performed independently, thus reducing the overall complexity of online diagnosis. See for example [6].

On-line Hierarchical Fault Adaptive Control

This work extends previous work we have done on online approaches for the safety control of a general class of hybrid systems. This year we have focused on a hierarchical online fault-adaptive control approach for complex systems made up of a number of interacting subsystems. To avoid complexity in the models and online analysis, diagnosis and fault-adaptive control is achieved by local units. To maintain overall performance, the problem of resource management for contending concurrent subsystems has to be addressed. We have developed a control structure, where predefined set-point specifications for system operation are used to derive optimizing utility functions for the subsystem controllers. We apply this approach in situations where a fault occurs in a system, and once the fault is isolated and identified, the controllers use the updated system model to derive new set point specifications and utility functions for the faulty system. See for example the work in [7]

We have successfully demonstrated the use of this system for a NASA application that involves components of the Advanced Life Support Systems for long-term manned missions. This is reported in [16].

Safety-Critical Distributed Applications

We developed a design methodology and the supporting tools to address feedback control problems with fault-tolerant requirements (e.g. automotive safety-critical applications).

The flow lets the designer specify independently:
- the algorithmic solution
- the distributed execution platform
- the fault behavior

The three aspects of the design are represented using respectively
- a flavor of synchronous dataflow called fault-tolerant dataflow (FTDF)
- a bipartite graph (channels and electronic control units) with performance annotations
- a relation between failure patterns (subsets of the architecture graph that may fail in a same iteration) and the corresponding subset of the algorithm that must be guaranteed

Based on these three aspects of the specification, an automatic synthesis tool introduces redundancy in the algorithms and schedules the FTDF actors on the distributed architecture, so that the fault behavior is met. In doing so, the scheduling tool aims at minimizing latency (the critical path from sensors to actuators). Finally some verification tools analyze the solution to extract timing and to verify replica determinism and fault behavior. The work is reported in [60]

We have tested the entire flow on a drive-by-wire example from BMW and a steer-by-wire example from General Motors. The experiments support the value of the methodology and suggested directions for further improvement.

2.2. Project Findings

Abstracts for key publications representing project findings during this reporting period, are provided here. These are listed alphabetically by first author. 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 Stochastic Approximation for Hybrid Systems

A. Abate, A. Ames, S. Sastry, In Proc. American Control Conference, (in publication), Portland, June 2005.

Abstract: 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.

[2] New Congestion Control Schemes over Wireless Networks: Stability Analysis

Alessandro Abate, Minghua Chen, Avideh Zakhor, Sankar Sastry, submitted to IFAC 05.

Abstract: The objective of this work is to introduce two original flow control schemes for wireless networks. The mathematical underpinnings lie on the recently-developed congestion control models for Transmission-Control-Protocol(TCP)-like schemes; more precisely, the model proposed by Kelly for the wired case is taken as a template, and properly extended to the more involved wireless setting. We introduce two ways to modify a part of the model; the first is through a static law, and the second via a dynamic one. In both cases, we prove the global stability of the schemes, and present a convergence rate study and a stochastic analysis.

[3] New Congestion Control Schemes over Wireless Networks: Delay Sensitivity Analys is and Simulations

A. Abate, M. Chen, S. Sastry. In Proc. International Federation of Automatic Control World Congress, (in publication), Prague, July 2005.

Abstract: This paper proposes two new congestion control schemes for packet switched wireless networks. Starting from the seminal work of Kelly (Kelly *et al.*, Dec 1999), we consider the decentralized flow control model for a TCP-like scheme and extend it to the wireless scenario. Motivated by the presence of channel errors, we introduce updates in the part of the model representing the number of connections the user establishes with the network; this assumption has important physical interpretation. Specifically, we propose two updates: the ¯rst is static, while the second evolves dynamically. The global stability of both schemes has been proved; also, a stochastic stability study and the rate of convergence of the two algorithms have been investigated. This paper focuses on the delay sensitivity of both schemes. A stability condition on the parameters of the system is introduced and proved. Moreover, some deeper insight on the structure of the oscillations of the system is attained. To support the theoretical results, simulations are provided.

[4] Robust Model Predictive Control through Adjustable Variables: An Application to Path Planning

A. Abate, L. El Ghaoui, In Proc. International Conference on Decision and Control, Atlantis, Bahamas, December 2004.

Abstract: Robustness in *Model Predictive Control* (MPC) is the main focus of this work. After a definition of the conceptual framework and of the problem's setting, we will analyze how a technique developed for studying robustness in *Convex Optimization* can be applied to address the problem of robustness in the MPC case. Therefore, exploiting this relationship between Control and Optimization, we will tackle robustness issues for the first setting through methods developed in the second framework. Proofs for our results are included. As an application of this Robust MPC result, we shall consider a Path Planning problem and discuss some simulations thereabout.

[5] A Stability Criterion for Stochastic Hybrid Systems

A. Abate, L. Shi, S. Simic, S. Sastry, In Proc. International Symposium of Mathematical Theory of Networks and Systems, Leuven, July 2004.

Abstract: This paper investigates the notion of stability for Stochastic Hybrid Systems. The uncertainty is introduced in the discrete jumps between the domains, as if we had an underlying Markov Chain. The jumps happen every fixed time T; moreover, a result is given for the case of probabilistic dwelling times inside each domain. Unlike the more classical Hybrid Sys-stems setting, the guards here are time-related, rather than space-related. We shall focus on vector fields describing input-less dynamical systems.Clearly, the uncertainty intrinsic to the model forces to introduce a fairly new definition of stability, which can be related to the classical Lyapunov one though. Proofs and simulations for our results are provided, as well as a motivational example from finance.

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

S. Abdelwahed, J. Wu, G. Biswas, E. J. Manders, paper no. 2005-01-2965, International Conference on Environmental Systems and European Symposium on Space Environmental Control Systems, (to appear), Rome Italy, July 11—14, 2005.

Abstract: 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.

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

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

Abstract: 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).

[8] Semantic Translation of Simulink/Stateflow models to Hybrid Automata using Graph Transformations

A. Agrawal, Gy. Simon, G. Karsai, Electronic Notes in Theoretical Computer Science, In Proc. Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2004), Volume 109, pp. 43-56.

Abstract: Embedded systems are often modeled using Matlab's Simulink and Stateflow (MSS), to simulate plant and controller behavior but these models lack support for formal verification. On the other hand verification techniques and tools do exist for models based on the notion of Hybrid Automata (HA) but there are no tools that can convert Simulink/Stateflow models into their semantically equivalent Hybrid Automata models. This paper describes a translation algorithm that converts a well-defined subset of the MSS modeling language into an equivalent hybrid automata. The translation has been

specified and implemented using a metamodel-based graph transformation tool. The translation process allows semantic interoperability between the industry-standard MSS tools and the new verification tools developed in the research community.

[9] Reusable Idioms and Patterns in Graph Transformation Languages

A. Agrawal, A. Vizhanyo, Z. Kalmar, F. Shi, A. Narayanan, G. Karsai, International Workshop on Graph-Based Tools, In Proc. 2004 International Conference on Graph Transformations, Rome, Italy, October, 2004.

Abstract: 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.

[10] Blowing Up Affine Hybrid Systems

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

Abstract: In this paper we construct the "blow up" of an affine hybrid system H, i.e., a new affine hybrid system Bl(H) in which H is embedded, that does not exhibit Zeno behavior. We show the existence of a bijection U between periodic orbits and equilibrium points of H and Bl(H) that preserves stability; we refer to this property as P-stability equivalence.

[11] Characterization of Zeno Behavior in Hybrid Systems using Homological Methods

A. D. Ames, S. Sastry, 24th American Control Conference 2005 (ACC'05), (in publication), Portland, OR, 2005.

Abstract: It is possible to associate to a hybrid system a single topological space--its underlying topological space. Simultaneously, every hybrid system has a graph as its indexing object--its underlying graph. Here we discuss the relationship between the underlying topological space of a hybrid system, its underlying graph and Zeno behavior. When each domain is contractible and the reset maps are homotopic to the identity map, the homology of the underlying topological space is isomorphic to the homology of the underlying graph; the nonexistence of Zeno is implied when the first homology is trivial. Moreover, the first homology is trivial when the null space of the incidence matrix is trivial. The result is an easy way to verify the nonexistence of Zeno behavior.

[12] A Homology Theory for Hybrid Systems: Hybrid Homology

A. D. Ames, S. Sastry, 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.

Abstract: By transferring the theory of hybrid systems to a categorical framework, it is possible to develop a homology theory for hybrid systems: hybrid homology. This is achieved by considering the underlying ``space'' of a hybrid system---its hybrid space or H-space. The homotopy colimit can be applied to this H-space to obtain a single topological space; the hybrid homology of an H-space is the homology of this space. The result is a spectral sequence converging to the hybrid homology of an H-space, providing a concrete way to compute this homology. Moreover, the hybrid homology of the H-space underlying a hybrid system gives useful information about the behavior of this system: the vanishing of the first hybrid homology of this H-space---when it is contractible and finite---implies that this hybrid system is not Zeno.

[13] Sufficient Conditions for the Existence of Zeno Behavior

A. D. Ames, S. Sastry, 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), (submitted for publication), Seville, Spain, Dec., 12--15, 2005.

Abstract: In this paper, sufficient conditions for the existence of Zeno behavior in a class of hybrid systems is 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 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.

[14] A Decentralized Approach to Sound Source Localization with Sensor Networks

I. Amundson, P. Schmidt, K. D. Frampton, Presented at the 2004 ASME International Mechanical Engineering Conference and Exposition, Anaheim CA, November 2004.

Abstract: A sound source localization system has been developed based on a decentralized sensor network. Decentralization permits all nodes in a network to handle their own processing and decision-making, and as a result, reduce network congestion and the need for a centralized processor. The system consists of an array of battery operated COTS Ethernet-ready embedded systems with an attached microphone circuit. The localization solution requires groups of at least four nodes to be active within the array to return an acceptable two-dimensional result. Sensor nodes, positioned randomly

over a ten square meter area, recorded detection times of impulsive sources with microsecond resolution. In order to achieve a scalable system, nodes were organized in groups of from 4 to 10 nodes. Grouping was determined by the selecting the nodes farthest apart from each other. A designated leader of each group analyzed the sound source arrival times and calculated the sound source location based on time-differences of arrival. Experimental results show that this approach to sound source localization can achieve accuracies of about 30 cm . Perhaps more importantly though, it is accomplished in a decentralized manner which can lead to a more flexible, scalable distributed sensor network.

[15] Web Service Interfaces

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

Abstract: We present a language for specifying web service interfaces. A web service interface puts three kinds of constraints on the users of the service.

First, the interface specifies the methods that can be called by a client, together with types of input and output parameters; these are called signature constraints. Second, the interface may specify propositional constraints on method calls and output values that may occur in a web service conversation; these are called consistency constraints. Third, the interface may specify temporal constraints on the ordering of method calls; these are called protocol constraints. The interfaces can be used to check, first, if two or more web services are compatible, and second, if a web service A can be safely substituted for a web service B. The algorithm for compatibility checking verifies that two or more interfaces fulfill each others' constraints. The algorithm for substitutivity checking verifies that service A demands fewer and fulfills more constraints than service B.

[16] Online Model-Based Diagnosis to Support Autonomous Operation of an Advanced Life Support System

G. Biswas, E.J. Manders, J.W. Ramirez, N. Mahadevan, S. Abdelwahed, Habitation: International Journal of Human Support Research, vol. 10, no. 1, pp. 21-38, 2004.

Abstract: This article describes methods for online model-based diagnosis of subsystems of the advanced life support system (ALS). The diagnosis methodology is tailored to detect, isolate, and identify faults in components of the system quickly so that fault-adaptive control techniques can be applied to maintain system operation without interruption. We describe the components of our hybrid modeling scheme and the diagnosis methodology, and then demonstrate the effectiveness of this methodology by building a detailed model of the reverse osmosis (RO) system of the water recovery system (WRS) of the ALS. This model is validated with real data collected from an experimental testbed at NASA JSC. A number of diagnosis experiments run on simulated faulty data are presented and the results are discussed.

[17] HyVisual: A Hybrid System Visual Modeler

C. Brooks, A. Cataldo, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, H. Zheng, Technical Memorandum UCB/ERL M04/18/, University of California, Berkeley, June 28, 2004.

Abstract: 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.

[18] 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 Memorandum UCB/ERL M04/27/, University of California, Berkeley, July 29, 2004.

Abstract: 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.

[19] 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 Memorandum UCB/ERL M04/16/, University of California, Berkeley, June 24, 2004.

Abstract: 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.

[20] Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)

C. Brooks, E. A. Lee, X. Liu, S. Neuendorffer, Y. Zhao, H. Zheng (eds.), Technical Memorandum UCB/ERL M04/17/, University of California, Berkeley, June 24, 2004.

Abstract: 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 usual 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.

[21] Discrete-Event Systems: Generalizing Metric Spaces and Fixed Point Semantics

A. Cataldo, E. A. Lee, X. Liu, E. Matsikoudis, H. Zheng, 16th International Conference on Concurrecy Theory (CONCUR 2005), (submitted for publication), San Francisco, CA, August 2005.

Abstract: This paper studies the semantics of discrete-event systems as a concurrent model of computation. The classical approach, which is based on metric spaces, does not handle well multiplicities of simultaneous events, yet such simultaneity is a common property of discrete-event models and modeling languages. (Consider, for example, delta time in VHDL.) In this paper, we develop a semantics using an extended notion of time. We give a generalization of metric spaces that we call tetric spaces. (A tetric functions like a metric, but its value is an element of a totally-ordered monoid rather than an element of the non-negative reals.) A straightforward generalization of the Banach fixed point theorem to tetric spaces supports the definition of a fixed-point semantics and generalizations of well-known sufficient conditions for avoidance of Zeno conditions.

[22] Verifying Quantitative Properties Using Bound Functions

A. Chakrabarti, K. Chatterjee, T. A. Henzinger, O. Kupferman and R. Majumdar, In Proc. 13th Advanced Research Working Conference on Correct Hardware Design and Verification Methods (CHARME 2005), Saarbrucken, Germany, October 3—6, 2005.

Abstract: In the boolean framework of model-based specification and verification, systems are graphs where each state is labeled with boolean propositions, and properties are languages where each trace has a boolean value (i.e., a trace either satisfies a property or it does not). We define and study a quantitative generalization of this traditional setting, where 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 quantitative 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 or 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 a quantitative-bound automaton, which consists of (1) a finite automaton with integer registers, and (2) a bound function f that maps each system K to an integer f(K). While a traditional automaton accepts or rejects a given trace, a quantitative automaton maps each trace to an integer. The bound function f(K) defines an upper bound on register values which depends on the system K. We show that every quantitative-bound automaton defines a dynamic program that provides model-checking and game-solving algorithms. Along with the linear-time, automaton-based view of quantitative verification, we present a corresponding branching-time view based on a quantitative-bound mu-calculus. We study the relationship, expressive power, and complexity of both views.

[23] Two-player Nonzero-sum ω-regular Games

Krishnendu Chatterjee, In Proc. of 16th International Conference on Concurrency Theory, (submitted for publication), San Francisco, CA, August 23—26, 2005.

Abstract: 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 players' goals are not necessarily conflicting), and undiscounted. We show that if each player has an ω-regular objective expressed as a parity objective, then there exists an ϵ-Nash equilibrium, for every ε >0. However, exact Nash equilibria need not exist. We study the complexity of finding values (payoff profile) of an ε-Nash equilibrium. We show that the values of an ε-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 ε-Nash equilibrium values of nonzero-sum concurrent games with reachability objectives. As a consequence we establish that values of an ε-Nash equilibrium can be approximated in FNP (functional NP), and hence in EXPTIME.

[24] The Complexity of Stochastic Rabin and Streett Games

K. Chatterjee, L. de Alfaro, T. A. Henzinger, 32nd International Colloquium of Automata, Languages and Programming (ICALP 05), (submitted for publication), Lisboa, Portugal, July 11--15, 2005.

Abstract: The theory of graph games with ω-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 {\em qualitative\/} problem asks for the set of states from which a player can win with probability~1(\emph{almost-sure winning}); the {\em quantitative\/} problem asks for the maximal probability of winning (\emph{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 ω-regular condition, optimal winning strategies are no more complex than almost-sure winning strategies.

[25] Trading Memory for Randomness

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

Abstract: Strategies in repeated games can be classified as to whether or not they use memory and/or randomization. the deterministic and probabilistic varieties. We characterize when memory and/or randomization are required for winning with respect to various classes of ω-regular objectives, noting particularly when the use of memory can be traded for the use of randomization. In particular, we show that Markov decision processes allow randomized memoryless optimal strategies for all M\"uller objectives. Furthermore, we show that 2-player probabilistic graph games allow randomized memoryless strategies for winning with probability~1 those M\"uller objectives which are upward-closed. Upward-closure means that if a set α of infinitely repeating vertices is winning, then all supersets of α are also winning.

[26] Mean-Payoff Parity Games

K. Chatterjee, T. A. Henzinger, M. Jurdzinski, 20th Annual Symposium of Logics in Computer Science (LICS 05), (submitted for publication), Chicago, IL, June 26—29, 2005

Abstract: Games played on graphs may have qualitative objectives, such as the satisfaction of an ω-regular property, or quantitative objectives, such as the optimization of a real-valued 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 {\it parity} condition and the quantitative component is a{\it 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.

[27] Counter-example Guided Planning

K. Chatterjee, T. A. Henzinger, R. Jhala, R. Majumdar, 21st International Conference in Uncertainty in Artifical Intelligence (UAI 05), (submitted for publication), University of Edinburgh, Edinburgh, Scotland, July 26—29, 2005.

Abstract: 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 {\em abstractions} of the model. The difficulty, of course, 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 {\em counterexample-guided abstraction refinement} has been successful to construct useful but parsimonious abstractions {\em automatically}.We extend this paradigm, for the first time, to {\em probabilistic} models (namely, $\twohalf$ 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.

[28] galsC: A Language for Event-Driven Embedded Systems

E. Cheong, J. Liu, Presented at Design, Automation and Test in Europe (DATE), Munich, Germany, March 7--11, 2005.

Abstract: We introduce galsC, a language designed for programming event-driven embedded systems such as sensor networks. galsC implements the TinyGALS programming model. At the local level, software components are linked via synchronous method calls to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors via parameters without explicitly passing messages. The galsC compiler extends the nesC compiler, which allows for better type checking and code generation. Having a well-structured concurrency model at the application level greatly reduces the risk of concurrency errors, such as deadlock and race conditions. The galsC language is implemented on the Berkeley motes and is compatible with the TinyOS/nesC component library. We use a multi-hop wireless sensor network as an example to illustrate the effectiveness of the language.

[29] Toward a Semantic Anchoring Infrastructure for Domain Specific Modeling Languages

K. Chen, J. Sztipanovits, S. Neema, M. Emerson, S. Abdelwahed, Embedded Systems Software Conference (EMSOFT 2005), (submitted for publication), Jersey City, NJ, September 18—22, 2005.

Abstract: 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.

[30] New Congestion Control Schemes Over Wireless Networks: Stability Analysis

M. Chen, A. Abate, S. Sastry. In Proc. International Federation of Automatic Control World Congress, (in publication), Prague, July 2005.

Abstract: This paper proposes two new congestion control schemes for packet switched wireless networks. Starting from the seminal work of Kelly (Kelly *et al.*, Dec 1999), we consider the decentralized flow control model for a TCP-like scheme and extend it to the wireless scenario. Motivated by the presence of channel errors, we introduce updates in the part of the model representing the number of connections the user establishes with the network; this assumption has important physical interpretation. Specifically, we propose two updates: the ¯rst is static, while the second evolves dynamically. The global stability of both schemes has been proved; also, a stochastic stability study and the rate of convergence of the two algorithms have been investigated. This paper focuses on the delay sensitivity of both schemes. A stability condition on the parameters of the system is introduced and proved. Moreover, some deeper insight on the structure of the oscillations of the system is attained. To support the theoretical results, simulations are provided.

[31] Stability and Delay Considerations for Flow Control Over Wireless Networks

M.Chen, A. Abate, A. Zakhor, S. Sastry, UCB ERL Tech Report No M05/14, Berkeley, CA, 2005.

Abstract: In this paper we develop a general framework for the problem of flow control over wireless networks, evaluate the existing approaches within that framework, and propose new ones. Significant progress has been made on the mathematical modeling of flow control for the wired Internet, among which Kelly's contribution is widely accepted as a standard framework. We extend Kelly's flow control framework to the wireless scenario, where the wireless link is assumed to have a fixed link capacity and a packet loss rate caused by the physical channel errors. In this framework, the problem of flow

control over wireless can be formulated as a convex optimization problem with noisy feedback. We then propose two new solutions to the problem achieving optimal performance by only modifying the application layer. The global stability and the delay sensitivity of the schemes are investigated, and verified by numerical results. Our work advocates the use of multiple connections for flow, or congestion control, over wireless.

[32] Simulation Based Deadlock Analysis for System Level Designs

X. Chen, A. Davare, H. Hsieh, A. Sangiovanni-Vincentelli, Y. Watanabe, ACM/IEEE Design Automation Conference, (submitted for publication), Anaheim, CA, June 2005.

Abstract: 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.

[33] The Best of Both Worlds: The Efficient Asynchronous Implementation of Synchronous Specifications

A. Davare, K. Lwin, A. Kondratyev, A. Sangiovanni-Vincentelli. Presented at ACM/IEEE Design Automation Conference, San Diego, CA, June 7 --11, 2004.

Abstract: The desynchronization approach combines a traditional synchronous specification style with a robust asynchronous implementation model. The main contribution of this paper is the description of two optimizations that decrease the overhead of desynchronization. First, we investigate the use of clustering to vary the granularity of desynchronization. Second, by applying temporal analysis on a formal execution model of the desynchronized design, we uncover significant amounts of timing slack. These methods are successfully applied to industrial RTL designs.

[34] Overview of the Ptolemy Project

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, Technical Memorandum UCB/ERL M01/11, EECS, University of California, Berkeley, March 6, 2001.

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

J. M. Eklund, J. Sprinkle, S. S. Sastry, American Control Conference (ACC) 2005, (In Publication), Portland, OR, Jun., 8-10, 2005.

Abstract: 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.

[36] Template Based Planning and Distributed Control for Networks of Unmanned Underwater Vehicles

J. M. Eklund, J. Sprinkle, S. S. Sastry, 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), (submitted for publication), Seville, Spain, Dec., 12-15, 2005.

Abstract: A decentralized control scheme for large packs of unmanned underwater vehicles (UUV) is proposed and investigated. This scheme is based on shared knowledge of a template, which includes operational plans, modes of operation, contingencies including the ability to adapt individual plans within the template to changing operational conditions, and the protocols for disseminating individual state, network and command information between UUVs. This template-based control enables complex and cooperative functionality by the network within the bounds of severe communications limitations and provides for a highly scalable solution for distributed control. Simulation results of medium-sized packs of UUVs are presented and the road ahead to physical implementation, experimentation and deployment is described.

[37] A MOF-Based Metamodeling Environment

M. Emerson, J. Sztipanovits, T. Bapty, Journal of Universal Computer Science, vol. 10, No. 10, pp. 1357-1382, October, 2004.

Abstract: The Meta Object Facility (MOF) forms one of the core standards of the Object Management Group's Model Driven Architecture. It has several use-cases, including as a repository service for storing abstract models used in distributed object-oriented software development, a development environment for generating CORBA IDL, and a metamodeling language for the rapid specification, construction, and management of domain-specific technology-neutral modeling languages. This paper will focus on the use of MOF as a metamodeling language and describe our latest work on changing the MIC metamodeling environment from UML/OCL to MOF. We have implemented a functional graphical metamodeling environment based on the MOF v1.4 standard using GME and GReAT. This implementation serves as a testament to the power of formally well-defined metamodeling and metamodel-based model transformation approaches. Furthermore, our

work gave us an opportunity to evaluate sevaral important features of MOF v1.4 as a metamodeling language: (a) Completeness of MOF v1.4 for defining the abstract syntax for complex (multiple aspect) DSML-s , (b) The Package concept for composing and reusing metamodels and (c) Facilities for modeling the mapping between the abstract and concrete syntax of DSML-s.

[38] Implementing a MOF-Based Metamodeling Environment Using Graph Transformations

M. Emerson, J. Sztipanovits, In Proc. 4th Workshop on Domain-Specific Modeling, pp. 83-92, 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Vancouver, Canada, October 2004.

Abstract: Versatile model-based design demands languages and tools which are suitable for the creation, manipulation, transformation, and composition of domain-specific modeling languages and domain models. The Meta Object Facility (MOF) forms the cornerstone of the OMG's Model Driven Architecture (MDA) as the standard metamodeling language for the specification of domain-specific languages. We have implemented MOF v1.4 as an alternative metamodeling language for the Generic Modeling Environment (GME), the flagship tool of Model Integrated Computing (MIC). Our implementation utilizes model-to-model transformations specified with the Graph Rewriting and Transformation toolsuite (GReAT) to translate between MOF and the UML-based GME metamodeling language. The technique described by this paper illustrates the role graph transformations can play in interfacing MIC technology to new and evolving modeling standards.

[39] Acoustic Self-Localization in a Distributed Sensor Network

K. D. Frampton, IEEE Sensors Journal, (in publication), 2005.

Abstract: The purpose of this work is to present a technique for determining the locations of nodes in a distributed sensor network. This technique is based on the Time Difference of Arrival (TDOA) of acoustic signals. In this scheme, several sound sources of known locations transmit while each node in the sensor network records the wave front time-of-arrival. Data from the nodes are transmitted to a central processor and the nonlinear TDOA equations are solved. Computational simulation results are presented in order to quantify the solution behavior and its sensitivity to likely error sources. Experimental self-localization results are also presented in order to demonstrate the potential for this approach in solving the challenging self-localization problem.

[40] Distributed Group-Based Vibration Control with a Networked Embedded System

K. D. Frampton, Journal of Intelligent Materials Systems and Structures, Vol. 14, pp. 307--314, 2005.

Abstract: The purpose of this work is to demonstrate the performance of a distributed vibration control system based on a networked embedded system. The platform from which control is affected consists of a network of computational elements called nodes.

Each node possesses its own computational capability, sensor, actuator and the ability to communicate with other nodes via a wired or wireless network. The primary focus of this work is to demonstrate the use of existing group management middleware concepts to enable vibration control with such a distributed network. Group management middleware is distributed software that provides for the establishment and maintenance of groups of distributed nodes and that provides for the network communication within such groups. The reason for developing distributed control based on group concepts is that communication of real-time sensor and actuator data among all system nodes would not be possible due to bandwidth constraints. Group management middleware provides for inter-node communications among subsets of nodes in an efficient and scalable manner. The objective of demonstrating the effectiveness of such grouping for distributed control is met by designing distributed feedback compensators that take advantage of node groups in order to affect their control. Two types of node groups are considered: groups based on physical proximity and groups based on modal sensitivity. The global control objective is to minimize the vibrational response of a rectangular plate in specific modes while minimizing spillover to out-of-bandwidth modes. Results of this investigation demonstrate that such a distributed control system can achieve vibration attenuations comparable to that of a centralized controller. The importance of efficient use of network communications bandwidth is also discussed with regard to the control architectures considered.

[41] Vibroacoustic Control with a Distributed Sensor Network

K. D. Frampton, Presented at Applications of Graph Transformations with Industrial Relevance (AGTIVE04), Williamsburg, VA, September, 2004.

Abstract: The purpose of this work is to demonstrate the ability of a distributed control system, based on a networked embedded system, to reduce acoustic radiation from a vibrating structure. The platform from which control is affected consists of a network of computational elements called nodes. Each node possesses its own computational capability, sensor, actuator and the ability to communicate with other nodes via a wired or wireless network. The primary focus of this work is to employ existing group management middleware concepts to enable vibration control with such a distributed network. Group management middleware is distributed software that provides for the establishment and maintenance of groups of distributed nodes and that provides for the network communication among such groups. This objective is met by designing distributed feedback compensators that take advantage of node groups in order to affect their control. Two types of node groups are considered: groups based on physical proximity and groups based on modal sensitivity. The global control objective is to minimize the vibrational response of a rectangular plate in specific modes while minimizing spillover to out-of-bandwidth modes. Results of this investigation demonstrate that such a distributed control system can achieve vibration attenuations comparable to that of a centralized controller. The importance of efficient use of network communications bandwidth is also discussed with regard to the control architectures considered.

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

M. Harren, G. C. Necula, 12th International Static Analysis Symposium (SAS '05), (in publication), London, UK, September 7-9, 2005.

Abstract: 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.

[43] Distributed Control to Improve Performance of Thermoelectric Coolers

R. D. Harvey, D. G. Walker, K. D. Frampton, Presented at 2004 ASME International Mechanical Engineering Conference and Exposition, Anaheim CA, November 2004.

Abstract: Thermoelectric coolers (TECs) have become more popular in chip cooling applications. However, due to material properties, the scope of TEC applicability is limited. The primary reason for this limitation stems from the poor efficiency of the TEC. This low efficiency causes increased heat production resulting in a very narrow band in which the TEC is effective. Since TECs are cooling units composed of numerous individual cooling elements, this band can be expanded by implementing distributed control of the individual cooler components. Distributed control is a system for allowing each element to be powered depending on the localized heat load. Distributed control would allow for increased cooling in hot spots while minimizing excess heat generated by the TEC in areas where it is not needed. The preliminary results suggest that this type of control may be feasible, and would result in a significant increase in the TEC effectiveness. The current work provides a closer look at the increased effectiveness and improves the previous models. The current model considers lateral heat conduction in the chip, as well as variable control of the individual cooling elements proportional to heat load. By modeling different scenarios for heat distributions and exploring the application of the individual cooling units, wider applicability of the TEC for computer chip cooling can be achieved.

[44] Concerns on Separation: Modeling Separation of Concerns in the Semantics of Embedded Systems

E. Jackson, J. Sztipanovits, In Proc. Embedded Software Systems Conference (EMSOFT'2005), (in publication), Jersey City, NJ, September 18—22, 2005.

Abstract: 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.

[45] Graph Transformations in OMG's Model-Driven Architecture

G. Karsai, A. Agrawal, In Proc. Applications of Graph Transformations with Industrial Relevance (AGTIVE 2003), LNCS 2062. pp. 243-259, Charlottesville, VA, September 29—October 1, 2003.

Abstract: The Model-Driven Architecture (MDA) vision of the Object Management Group offers a unique opportunity for introducing Graph Transformation (GT) technology to the software industry. The paper proposes a domain-specific refinement of MDA, and describes a practical manifestation of MDA called Model-Integrated Computing (MIC). MIC extends MDA towards domain-specific modeling languages, and it is well supported by various generic tools that include model transformation tools based on graph transformations. The MIC tools are metaprogrammable, i.e. they can be tailored for specific domains using metamodels that include metamodels of transformations. The paper describes the development process and the supporting tools of MIC, and it raises a number of issues for future research on GT in MDA.

[46] Design Patterns for Open Tool Integration

G. Karsai, A. Lang, S. Neema, Journal of Software and System Modeling, vol 4., no. 1, DOI: 10.1007/s10270-004-0073-y, 2004.

Abstract: Design tool integration is a highly relevant area of software engineering, which can greatly improve the efficiency of development processes. Design patterns have been widely recognized as important contributors to the success of software systems. This paper describes and compares two large-grain, architectural design patterns that solve specific design tool integration problems. Both patterns have been implemented and used in real-life engineering processes.

[47] Real-Time Systems Design in Ptolemy II: A Time-Triggered Approach

V. Krishnan, Master's Report, Technical Memorandum UCB/ERL M04/22/, University of California, Berkeley, July 12, 2004.

Abstract: In this report is described a software infrastructure to enable users to design hard real-time systems from Ptolemy II [1]. The Giotto [2] domain within the Ptolemy II

design environment is made use of to model systems which are then compiled and executed on KURT-Linux [3], a real time flavor of Linux.

The first stage of the software takes a graphical model as an input to generate intermediate code in the C language. This intermediate code consists of the task-code to be executed, as well as a representation of their timing requirements.

The second stage, called the Embedded Machine [5] reads in the timing information and interprets it to release the tasks for execution as per the stated requirements. The released tasks can either be assigned to a standard scheduler such as EDF, or to a scheduling interpreter called the Scheduling machine, or S Machine.

The S Machine was developed to gain fine grained control over the scheduling of tasks. The S Machine requires as input scheduling information that specifies a time line for the tasks involved thus giving the designer maximum flexibility over task scheduling, and consequently greater resource utilization. The E&S Machines when compiled along with the generated task and timing code for the KURT-Linux platform forms an executable that delivers predictable real-time performance. The benefit this approach offers is that the real-time tasks can run along with ordinary Linux tasks without the timing properties of the real-time tasks being affected.

An audio application was designed to illustrate the effectiveness of this tool-flow, which achieved a timing uncertainty of less than 130 microseconds in its task execution times.

[48] Engineering Education: A Focus on System, in Advances in Control, Communication Networks, and Transportation Systems: In Honor of Pravin Varaiya

E. A. Lee, E.H. Abed (Ed.), Systems and Control: Foundations and Applications Series, Birkhauser, Boston, 2005.

Abstract: Engineers have a major advantage over scientists. For the most part, the systems we analyze are of our own devising. It has not always been so. Not long ago, the principle objective of engineering was to coax physical materials to do our bidding by leveraging their intrinsic physical properties. The discipline was of "applied science." Today, a great deal of engineering is about coaxing abstractions that we have invented. The abstractions provided by microprocessors, programming languages, operating systems, and computer networks are only loosely linked to the underlying physics of electronics.

[49] Absolutely Positively On Time: What Would It Take?

E. A. Lee, Editorial, February 19, 2005, Available at http://ptolemy.eecs.berkeley.edu/publications/papers/05/EmbeddedSoftwareColumn/

Editorial, March 8, 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. Although synchronous digital logic delivers precise timing determinacy, advances in computer architecture have made it difficult or impossible to estimate 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. In this column, 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."

Twentieth century computer science has taught us that everything that can be computed can be specified by a Turing machine. "Computation" is accomplished by a terminating sequence of state transformations. This core abstraction underlies the design of nearly all computers, programming languages, and operating systems in use today. But unfortunately, this core abstraction does not fit embedded software very well.

This core abstraction fits reasonably well if embedded software is simply "software on small computers." In this view, embedded software differs from other software only in its resource limitations (small memory, small data word sizes, and relatively slow clocks). In this view, the "embedded software problem" is an optimization problem. Solutions emphasize efficiency; engineers write software at a very low level (in assembly code or C), avoid operating systems with a rich suite of services, and use specialized computer architectures such as programmable DSPs and network processors that provide hardware support for common operations. These solutions have defined the practice of embedded software design and development for the last 25 years or so.

Of course, thanks to the semiconductor industry's ability to follow Moore's law, the resource limitations of 25 years ago should have almost entirely evaporated today. Why then has embedded software design and development changed so little? It may be that extreme competitive pressure in products based on embedded software, such as consumer electronics, rewards only the most efficient solutions. This argument is questionable, however. There are many examples where functionality has proven more important than efficiency. It is arguable that resource limitations are not the only defining factor for embedded software, and may not even be the principal factor.

There are clues that embedded software differs from other software in more fundamental ways. If we examine carefully why engineers write embedded software in assembly code or C, we discover that efficiency is not the only concern, and may not even be the main concern. The reasons may include, for example, the need to count cycles in a critical inner loop, not to make it fast, but rather to make it predictable. No widely used programming language integrates a way to specify timing requirements or constraints.

Instead, the abstractions they offer are about scalability (inheritance, dynamic binding, polymorphism, memory management), and, if anything, further obscure timing (consider the impact of garbage collection on timing). Counting cycles, of course, becomes extremely difficult on modern processor architectures, where memory hierarchy (caches), dynamic dispatch, and speculative execution make it nearly impossible to tell how long it will take to execute a particular piece of code. Worse, execution time is context dependent, which leads to unmanageable variability. Still worse, programming languages are almost always Turing complete, and as a consequence, execution time is undecidable in general. Embedded software designers must choose alternative processor architectures such as programmable DSPs, and must use disciplined programming techniques (e.g. avoiding recursion) to get predictable timing.

Another reason engineers stick to low-level programming is that embedded software typically has to interact with hardware that is specialized to the application. In conventional software, interaction with hardware is the domain of the operating system. Device drivers are not typically part of an application program, and are not typically created by application designers. But in the embedded software context, generic hardware interfaces are rarer. The fact is that creating interfaces to hardware is not something that higher level languages support. For example, although concurrency is not uncommon in modern programming languages (consider threads in Java), no widely used programming language includes in its semantics the notion of interrupts. Yet the concept is not difficult, and it can be built into programming languages (consider for example nesC and TinyOS, which are widely used for programming sensor networks).

It becomes apparent that the avoidance of so many recent improvements in computation is not due to ignorance of those improvements. It is due to a mismatch of the core abstractions and the technologies built on those core abstractions. In embedded software, time matters. In the 20th century abstractions of computing, time is irrelevant. In embedded software, concurrency and interaction with hardware are intrinsic, since embedded software engages the physical world in non-trivial ways (more than keyboards and screens). The most influential 20th century computing abstractions speak only weakly about concurrency, if at all. Even the core 20th century notion of "computable" is at odds with the requirements of embedded software. In this notion, useful computation terminates, but termination is undecidable. In embedded software, termination is failure, and yet to get predictable timing, subcomputations must decidably terminate.

Embedded systems are integrations of software and hardware where the software reacts to sensor data and/or issues commands to actuators. The physical system is an integral part of the design and the software must be conceptualized to operate in concert with that physical system. Physical systems are intrinsically concurrent and temporal. Actions and reactions happen simultaneously and over time, and the metric properties of time are an essential part of the behavior of the system. Prevailing software methods abstract away time, replacing it with ordering. In imperative languages such as C, C++, and Java, the order of actions is defined by the program, but not their timing. This prevailing imperative abstraction is overlaid with another, that of threads or processes, typically provided by the operating system, but occasionally by the language (as in Java).

The lack of timing in the core abstraction is a flaw, from the perspective of embedded software, and threads as a concurrency model are a poor match for embedded systems. They are mainly focused on providing an illusion of parallelism in fundamentally sequential models, and they work well only for modest levels of concurrency or for highly decoupled systems that are sharing resources, where best-effort scheduling policies are sufficient. Indeed, several recent innovative embedded software frameworks, such as Simulink (from The MathWorks), nesC and TinyOS (from Berkeley), and Lustre/SCADE (from Esterel Technologies) are concurrent programming languages with no threads or processes in the programmer's model.

Embedded software systems are generally held to a much higher reliability standard than general purpose software. Often, failures in the software can be life threatening (e.g., in avionics and military systems). The prevailing concurrency model in general purpose software that is based on threads does not achieve adequate reliability. In this prevailing model, interaction between threads is extremely difficult for humans to understand. Although it is arguable that concurrent computation is inherently complex, threads make it far more complex because between any two atomic operations (a concept that is rarely well defined), any part of the state of the system can change. The basic techniques for controlling this interaction use semaphores and mutual exclusion locks, methods that date back to the 1960s. Many uses of these techniques lead to deadlock or livelock. In general-purpose computing, this is inconvenient, and typically forces a restart of the program (or even a reboot of the machine). However, in embedded software, such errors can be far more than inconvenient. Moreover, software is often written without sufficient use of these interlock mechanisms, resulting in race conditions that yield nondeterministic program behavior. In practice, errors due to misuse (or no use) of semaphores and mutual exclusion locks are extremely difficult to detect by testing. Code can be exercised for years before a design flaw appears. Static analysis techniques can help (e.g. Sun Microsystems' LockLint), but these methods are often thwarted by conservative approximations and/or false positives, and they are not widely used in practice.

It can be argued that the unreliability of multi-threaded programs is due at least in part to inadequate software engineering processes. For example, better code reviews, better specifications, better compliance testing, and better planning of the development process can help solve the problems. It is certainly true that these techniques can help. However, programs that use threads can be extremely difficult for programmers to understand. If a program is incomprehensible, then no amount of process improvement will make it reliable. Formal methods can help detect flaws in threaded programs, and in the process can improve the understanding that a designer has of the behavior of a complex program. But if the basic mechanisms fundamentally lead to programs that are difficult to understand, then these improvements will fall short of delivering reliable software. Incomprehensible software will always be unreliable software.

Prevailing industrial practice in embedded software relies on bench testing for concurrency and timing properties. This has worked reasonably well, because programs are small, and because the software gets encased in a box with no outside connectivity that can alter the behavior of the software. However, applications today demand that

embedded systems be feature-rich and networked, so bench testing and encasing become inadequate. In a networked environment, it becomes impossible to test the software under all possible conditions, because the environment is not known. Moreover, general-purpose networking techniques themselves make program behavior much more unpredictable.

What would it take to achieve concurrent and networked embedded software that was absolutely positively on time (say, to the precision and reliability of digital logic)? Unfortunately, everything would have to change. The core abstractions of computing need to be modified to embrace time. Computer architectures need to be changed to deliver precisely timed behaviors. Networking techniques need to be changed to provide time concurrence. Programming languages have to change to embrace time and concurrency in their core semantics. Operating systems have to change to rely less on priorities to (indirectly) specify timing requirements. Software engineering methods need to change to specify and analyze the temporal dynamics of software. And the traditional boundary between the operating system and the programming language needs to be rethought. What is needed is nearly a reinvention of computer science.

Fortunately, there is quite a bit to draw on. To name a few examples, architecture techniques such as software-managed caches promise to deliver much of the benefit of memory hierarchy without the timing unpredictability. Operating systems such as TinyOS provide simple ways to create thin wrappers around hardware, and with nesC, alter the OS/language boundary. Programming languages such as Lustre/SCADE provide understandable and analyzable concurrency. Embedded software languages such as Simulink provide time in their semantics. Network time synchronization methods such as IEEE 1588 provide time concurrence at resolutions (tens of nanoseconds) far finer than any processor or software architectures can deal with today. The time is ripe to pull these techniques together and build the 21st Century (Embedded) Computer Science.

Thanks to helpful comments from Elaine Cheong and Douglas Niehaus.

[50] Balance between Formal and Informal Methods, Engineering and Artistry, Evolution and Rebuild

E. A. Lee, Technical Memorandum UCB/ERL M04/19 /, University of California, Berkeley, July 4, 2004.

Abstract: This paper is the result of a workshop entitled "Software Reliability for FCS" that was organized by the Army Research Office, held on May 18-19, 2004, and hosted by: Institute for Software Integrated Systems (ISIS), Vanderbilt University. I was given the charge of leading one of four topic areas, and was assigned the title. This is my summary of the results of the workshop on this topic.

It may well be that established approaches to software engineering will not be sufficient to avert a software disaster in FCS and similarly ambitious, software-intensive efforts. This topic examines the tension between informal methods, particularly those that focus on the human, creative process of software engineering and the management of that

process, and formal methods, specifically those that rely on mathematically rooted systems theories and semantic frameworks. It is arguable that, as these approaches are construed today by their respective (largely disjoint) research communities, neither offers much hope of delivering reliable FCS software. Although certainly these communities have something to offer, the difficulties may be more deeply rooted than either approach can address. In this workshop, we took an aggressive stand that there are problems in software that are intrinsically unsolvable with today's software technology. This stand asserts that no amount of process will fix the problems because the problems are not with the process, and that today's formal techniques cannot solve the problem as long as they remain focused on formalizing today's software technologies. A sea change in the underlying software technology could lead to more effective informal and formal methods. What form could that take?

[51] Concurrent Models of Computation for Embedded Software

E. A. Lee, Technical Memorandum, University of California, Berkeley, UCB/ERL M05/2/, January 4, 2005.

Abstract: This document collects the lecture notes that I used when teaching EECS 290n in the Fall of 2004. This course is an advanced graduate course with a nominal title of Advanced Topics in Systems Theory. This instance of the course studies 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 are considered, including real-time operating systems, process networks, synchronous languages (such as used in SCADE, Esterel, and Statecharts), timed models (such as used in Simulink, Giotto, VHDL, and Verilog), and dataflow models (such as a used in Labview and SPW). The course combines an experimental approach with a study of formal semantics. The objective is to develop a deep understanding of the wealth of alternative approaches to managing concurrency and time in software.

The experimental portion of the course uses Ptolemy II as the software laboratory. The formal semantics portion of the course builds on the mathematics of partially ordered sets, particularly as applied to prefix orders and Scott orders. It develops a framework for models of computation for concurrent systems that uses partially ordered tags associated with events. Discrete-event models, synchronous/reactive languages, dataflow models, and process networks are studied in this context. Basic issues of computability, boundedness, determinacy, liveness, and the modeling of time are studied. Classes of functions over partial orders, including continuous, monotonic, stable, and sequential functions are considered, as are semantics based on fixed-point theorems.

[52] What are the Key Challenges in Embedded Software?

E. A. Lee, Guest Editorial in System Design Frontier, Shanghai Hometown Microsystems Inc., Volume 2, Number 1, January 2005.

Abstract: Embedded software has traditionally been thought of as "software on small computers." In this traditional view, the principal problem is resource limitations (small memory, small data word sizes, and relatively slow clocks). Solutions emphasize efficiency; software is written at a very low level (in assembly code or C), operating systems with a rich suite of services are avoided, and specialized computer architectures such as programmable DSPs and network processors are developed to provide hardware support for common operations. These solutions have defined the practice of embedded software design and development for the last 25 years or so.

Of course, thanks to the semiconductor industry's ability to follow Moore's law, the resource limitations of 25 years ago should have almost entirely evaporated today. Why then has embedded software design and development changed so little? It may be that extreme competitive pressure in products based on embedded software, such as consumer electronics, rewards only the most efficient solutions. This argument is questionable, however, since there are many examples where functionality has proven more important than efficiency. We will argue that resource limitations are not the only defining factor for embedded software, and may not even be the principal factor.

Resource limitations are an issue to some degree with almost all software. So generic improvements in software engineering should, in theory, also help with embedded software. There are several hints, however, that embedded software is different in fundamental ways. For one, object-oriented techniques such as inheritance, dynamic binding, and polymorphism are rarely used in practice with embedded software development. In another example, processors used for embedded systems often avoid the memory hierarchy techniques that are used in general purpose processors to deliver large virtual memory spaces and faster execution using caches. In a third example, automated memory management, with allocation, deallocation, and garbage collection, are largely avoided in embedded software. To be fair, there are some successful applications of these technologies in embedded software, such as the use of Java in cell phones, but their application remains limited and is largely providing services that are actually more akin to general-purpose software applications (such as database services in cell phones).

Embedded systems are integrations of software and hardware where the software reacts to sensor data and issues commands to actuators. The physical system is an integral part of the design and the software must be conceptualized to operate in concert with that physical system. Physical systems are intrinsically concurrent and temporal. Actions and reactions happen simultaneously and over time, and the metric properties of time are an essential part of the behavior of the system. Prevailing software methods abstract away time, replacing it with ordering. In imperative languages such as C, C++, and Java, the order of actions is defined by the program, but not their timing. This prevailing imperative abstraction is overlaid with another, that of threads or processes, typically provided by the operating system, but occasionally by the language (as in Java).

The lack of timing in the core abstraction is a flaw, from the perspective of embedded software, and threads as a concurrency model are a poor match to embedded systems. They are mainly focused on providing an illusion of concurrency in fundamentally sequential models, and they work well only for modest levels of concurrency or for highly decoupled systems that are sharing resources, where best-effort scheduling policies are sufficient. Indeed, several recent innovative embedded software frameworks, such as Simulink (from the MathWorks), TinyOS (from Berkeley), and SCADE (from Esterel Technologies) have no threads or processes.

Embedded software systems are generally held to a much higher reliability standard than general purpose software. Often, failures in the software can be life threatening (e.g., in avionics and military systems). The prevailing concurrency model based on threads does not achieve adequate reliability. In this prevailing model, interaction between threads is extremely difficult for humans to understand. The basic techniques for controlling this interaction use semaphores and mutual exclusion locks, methods that date back to the 1960s. These techniques often lead to deadlock or livelock conditions, where all or part of a program cannot continue executing. In general-purpose computing, this is inconvenient, and typically forces a restart of the program (or even a reboot of the machine). However, in embedded software, such errors can be far more than inconvenient. Moreover, software is often written without sufficient use of these interlock mechanisms, resulting in race conditions that yield nondeterministic program behavior.

In practice, errors due to misuse (or no use) of semaphores and mutual exclusion locks are extremely difficult to detect by testing. Code can be exercised for years before a design flaw appears. Static analysis techniques can help (e.g. Sun Microsystems' LockLint), but these methods are often thwarted by conservative approximations and/or false positives, and they are not widely used in practice.

It can be argued that the unreliability of multi-threaded programs is due at least in part to inadequate software engineering processes. For example, better code reviews, better specifications, better compliance testing, and better planning of the development process can help solve the problems. It is certainly true that these techniques can help. However, programs that use threads can be extremely difficult for programmers to understand. If a program is incomprehensible, then no amount of process improvement will make it reliable. Formal methods can help detect flaws in threaded programs, and in the process can improve the understanding that a designer has of the behavior of a complex program. But if the basic mechanisms fundamentally lead to programs that are difficult to understand, then these improvements will fall short of delivering reliable software.

The key challenge in embedded software is to invent (or apply) abstractions that yield more understandable programs that are both concurrent and timed. These abstractions will be very different from those widely used for the design and development of general-purpose software.

[53] Classes and Subclasses in Actor-Oriented Design

E. A. Lee, S. Neuendorffer, invited paper, Conference on Formal Methods and Models for Codesign/ (MEMOCODE), San Diego, CA, USA, June 22-25, 2004.

Abstract: Actor-oriented languages provide a component composition methodology that emphasizes concurrency. The interfaces to actors are *parameters* and *ports* (vs. members and methods in object-oriented languages). Actors interact with one another through their ports via a messaging schema that can follow any of several concurrent semantics (vs. procedure calls, with prevail in OO languages). Domain-specific actor-oriented languages and frameworks are common (e.g. Simulink, LabVIEW, and many others). However, they lack many of the modularity and abstraction mechanisms that programmers have become accustomed to in OO languages, such as classes, inheritance, interfaces, and polymorphism. This extended abstract shows the form that such mechanisms might take in AO languages. A prototype of these mechanisms realized in Ptolemy II is described.

[54] Concurrent Models of Computation for Embedded Software

E. A. Lee, S. Neuendorffer, Technical Memorandum UCB/ERL M04/26/, University of California, Berkeley, July 22, 2004.

Abstract: The prevailing abstractions for software are better suited to the traditional problem of computation, namely transformation of data, than to the problems of embedded software. These abstractions have weak notions of concurrency and the passage of time, which are key elements of embedded software. Innovations such as nesC/TinyOS (developed for programming very small programmable sensor nodes called motes), Click (created to support the design of software-based network routers), Simulink with Real-Time Workshop (created for embedded control software), and Lustre/SCADE (created for safety-critical embedded software) offer abstractions that address some of these issues and differ significantly from the prevailing abstractions in software engineering. This paper surveys some of the abstractions that have been explored.

[55] Operational Semantics of Hybrid Systems

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

Abstract: This paper discusses an interpretation of hybrid systems as executable models. A specification of a hybrid system for this purpose can be viewed as a program in a domain-specific programming language. We describe the semantics of HyVisual, which is such a domain-specific programming language. The semantic properties of such a language affect our ability to understand, execute, and analyze a model. We discuss several semantic issues that come in defining such a programming language, such as the interpretation of discontinuities in continuous-time signals, and the interpretation of discrete-event signals in hybrid systems, and the consequences of numerical ODE solver techniques. We describe the solution in HyVisual by giving its operational semantics.

[56] Scientific Workflow Management and the KEPLER System

B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, Y. Zhao, Concurrency & Computation: Practice & Experience, (in publication), draft version, March 2005.

Abstract: Many scientific disciplines are now data and information driven, and new scientific knowledge is often gained by scientists putting together data analysis and knowledge discovery "pipelines." A related trend is that more and more scientific communities realize the benefits of sharing their data and computational services, and are thus contributing to a distributed data and computational and community infrastrucutre (a.k.a. "the Grid"). However, this infrasctructure is only a means to an end and scientists ideally should be bothered little with its existence. The goal is for scientists to focus on development and use of what we call *scientific workflows*. These are networks of analytical steps that may involve, e.g., database access and querying steps, data analysis and mining steps, and many other steps including computationally intensive jobs on high performance cluster computers. In this paper we describe characteristics of and requirements for scientific workflows as identified in a number of our application projects. We then elaborate on KEPLER, a particular scientific workflow system, currently under development across a number of scientific data management projects. We describe some key features of KEPLER and its underlying PTOLEMY II system, planned extensions, and areas of future research. KEPLER is a community-driven, open source project, and we always welcome related projects and new contributors to join.

[57] Automatic Verification of Component-based Real-time CORBA Applications

G. Madl, S. Abdelwahed, G. Karsai, In Proc. 25th IEEE International Real-Time Systems Symposium (RTSS'04), Lisbon, Portugal, Dec. 2004, pp. 231—240.

Abstract: Distributed real-time embedded (DRE) systems often need to satisfy various time, resource and faulttolerance constraints. To manage the complexity of scheduling these systems many methods use Rate Monotonic Scheduling assuming a time-triggered architecture. This paper presents a method that captures the reactive behavior of complex time- and event-driven systems, can provide simulation runs and can provide exact characterization of timed properties of component-based DRE applications that use the publisher/subscriber communication pattern. We demonstrate our approach on real-time CORBA avionics applications.

[58] Verifying Distributed Real-Time Properties of Embedded Systems via Graph Transformations and Model Checking

G. Madl, S. Abdelwahed, D., Real-Time Systems Journal, (in publication), 2005.

Abstract: quality of service (QoS) needs of distributed real-time embedded (DRE) systems. However, component middleware also introduces challenges for DRE system developers, such as evaluating the predictability of DRE system behavior and choosing the right design alternatives before committing to a specific platform. Model-based technologies help address these issues by enabling design-time analysis and providing the means to automate the development, configuration, and integration of component-based

DRE systems. This paper provides three contributions to research on model-based design and analysis of component-based DRE systems. First, we apply model checking techniques to DRE design models using model transformations to verify key QoS properties of component-based DRE systems developed using Real-time CORBA. Second, we implemented a property-preserving model-transformation method and used it to define formal semantics to component-based modeling languages. Third, we develop a formal description of the Boeing Bold Stroke architecture from which abstract behavioral models can be constructed and verified. Our results show that model-based techniques enable design-time analysis of timed properties and can be applied to effectively predict, simulate , and verify the event-driven behavior of component-based DRE systems.

[59] Shooter Localization in Urban Terrain

M. Maroti, G. Simon, A. Ledeczi, J. Sztipanovits, IEEE Computer, pp. 60-61, August, 2004.

Abstract: The paper describes PinPtr, an acoustic sensor network-based shooter localization system. Instead of using a few expensive acoustic sensors, a low-cost ad-hoc acoustic sensor network measures both the muzzle blast and shock wave to accurately determine the location of the shooter and the trajectory of the bullet. The basic idea is simple: using the arrival times of the acoustic events at different sensor locations, the shooter position is calculated using the speed of sound and the location of the sensors. The robust sensor fusion algorithm, which is running on the base station, is based on a search on a hyper-surface defined by a consistency function The consistency function, which provides the number of sensor measurements consistent with hypothetical shooter positions and shot time, automatically classifies measurements and eliminates those, which are erroneous or the result of multipath effects. The highly redundant ad-hoc sensor field ensures that enough good measurements are left to determine the shooter's position. The global maximum of the surface, corresponding to the shooter position, is guaranteed to be found by a fast search algorithm.

[60] Fault Tolerant Data Flow Modeling Using the Generic Modeling Environment

M. L. McKelvin, Jr, J. Sprinkle, C. Pinello, A. Sangiovanni-Vincentelli, 12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, Greenbelt, Maryland, Apr. 4--5, 2005, pp. 229–235.

Abstract: Designing embedded software for safety-critical, real-time feedback control applications is a complex and error prone task. Fault tolerance is an important aspect of safety. In general, fault tolerance is achieved by duplicating hardware components, a solution that is often more expensive than needed. In particular applications, such as automotive electronics, a subset of the functionalities has to be guaranteed while others are not crucial to the safety of the operation of the vehicle. In this case, we must make sure that this subset is operational under the potential faults of the architecture. A model of computation called Fault-Tolerant Data Flow (FTDF) was recently introduced to describe at the highest level of abstraction of the design the fault tolerance requirements on the functionality of the system. Then, the problem of implementing the system efficiently on a platform consists of finding a mapping of the FTDF model on the

components of the platform. A complete design flow for this kind of application requires a user-friendly graphical interface to capture the functionality of the systems with the FTDF model, algorithms for choosing an architecture optimally, (possibly automatic) code generation for the parts of the system to be implemented in software and verification tools. In this paper, we use the Generic Modeling Environment (GME) developed at Vanderbilt University to design a graphical design capture system and to provide the infrastructure for automatic code generation. The design flow is embedded into the Metropolis environment developed at the University of California at Berkeley to provide the necessary verification and analysis framework.

[61] A Visual Language for Describing Instruction Sets and Generating Decoders

T. Meyerowitz, J. Sprinkle, A. Sangiovanni-Vincentelli, 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Vancouver, BC, Oct., 25, 2004, pp. 23–32.

Abstract: We detail the syntax and semantics of ISA_ML, a visual modeling language for describing Instruction Set Architectures of microprocessors, and an accompanying tool that takes a description in the language and generates decoders from it in the form of a disassembler and a micro-architectural trace interfacer. The language and tool were built using the Generic Modeling Environment (GME), and leverage the concepts of meta-modeling to increase productivity and to provide extensive error checking to the modeler. Using this tool, we were able to construct a model of significant subsets of the MIPS, ARM, and PowerPC instruction sets each in 8 hours or less. This language can be retargeted for other purposes, such as generating synthesizable instruction decoders.

[62] CCured: Type-Safe Retrofitting of Legacy Software

G. C. Necula, J. Condit, M. Harren, S. McPeak, W. Weimer, ACM Transactions on Programming Languages and Systems, vol. 27, no. 3, May, 2005.

Abstract: This article describes CCured, a program transformation system that adds type safety guarantees to existing C programs. CCured attempts to verify statically that memory errors cannot occur, and it inserts run-time checks where static verification is insufficient. CCured extends C's type system by separating pointer types according to their usage, and it uses a surprisingly simple type inference algorithm that is able to infer the appropriate pointer kinds for existing C programs. CCured uses physical subtyping to recognize and verify a large number of type casts at compile time. Additional type casts are verified using run-time type information. CCured uses two instrumentation schemes, one that is optimized for performance and one in which metadata is stored in a separate data structure whose shape mirrors that of the original user data. This latter scheme allows instrumented programs to invoke external functions directly on the program's data without the use of a wrapper function. We have used CCured on real-world security-critical network daemons to produce instrumented versions without memory-safety vulnerabilities, and we have found several bugs in these programs. The instrumented code is efficient enough to be used in day-to-day operations.

[63] Model-Integrated Computing for Heterogeneous Systems

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

Abstract: Modern embedded and networked embedded system applications are demanding very high performance from systems with minimal resources. These applications must also be flexible to operate in a rapidly changing environment. High performance with limited resources needs application-specific architectures, while flexibility requires adaptation capabilities. Design of these systems creates unique challenges, since the traditional decomposition of the design space to hardware and software components and to functional and non-functional requirements do not give acceptable performance. Model-Integrated Computing (MIC) is an emerging design technology, which integrates all essential aspects of system design in a general, but highly customizable framework. This paper provides an overview of MIC and shows its application in the design of reconfigurable processing system.

[64] Actor-Oriented Metaprogramming

S. Neuendorffer, PhD Thesis, University of California, Berkeley, December 21, 2004.

Abstract: Robust design of concurrent systems is important in many areas of engineering, from embedded systems to scientific computing. Designing such systems using dataflow-oriented models can expose large amounts of concurrency to system implementation. Utilizing this concurrency effectively enables distributed execution and increased throughput, or reduced power usage at the same throughput. Code generation can then be used to automatically transform the design into an implementation, allowing design refactoring at the dataflow level and reduced design time over hand implementation.

This thesis focuses particularly on the benefits and disadvantages that arise when constructing models from generic, parameterized, dataflow-oriented components called *actors*. A designer can easily reuse actors in different models with different parameter values, data types, and interaction semantics. Additionally, during execution of a model actors can be reconfigured by changing their connections or assigning new parameter values. This form of reconfiguration can conveniently represent adaptive systems, systems with multiple operating modes, systems without fixed structure, and systems that control other systems. Ptolemy II is a Java-based design environment that supports the construction and execution of hierarchical, reconfigurable models using actors.

Unfortunately, allowing unconstrained reconfiguration of actors can sometimes cause problems. If a model is reconfigured, it may no longer accurately represent the system being modeled. Reconfiguration may prevent the application of static scheduling analysis to improve execution performance. In systems with data type parameters, reconfiguration may prevent static analysis of data types, eliminating an important form of error

detection. In such cases, it is therefore useful to limit which parameters or structures in a model can be reconfigured, or when during execution reconfiguration can occur.

This thesis describes a reconfiguration analysis that determines when reconfiguration occurs in a hierarchical model. Given appropriate formulated constraints, the analysis can alert a designer to potential design problems. The analysis is based on a mathematical framework for approximately describing periodic points in the behavior of a model. This framework has a lattice structure that reflects the hierarchical structure of actors in a model. Because of the lattice structure of the framework, this analysis can be performed efficiently. Models of two different systems are presented where this analysis helps verify that reconfiguration does not violate the assumptions of the model.

Run-time reconfiguration of actors not only presents difficulties for a system modeler, but can also impede efficient system implementation. In order to support run-time reconfiguration of actors in Java, Ptolemy II introduces extra levels of indirection into many operations. The overhead from this indirection is incurred in all models, even if a particular model does not use reconfiguration.

In order to remove the indirection overhead, we have developed a system called Copernicus which transforms a Ptolemy II model into self-contained Java code. In performing this transformation the Java code for each actor is specialized to its usage in a particular model. As a result, indirection overhead only remains in the generated code if it is required by reconfiguration in the model. The specialization is guided by various types of static analysis, including data type analysis and analysis of reconfiguration. In certain cases, the generated code runs 100 times faster and with almost no memory allocation, compared to the same model running in a Ptolemy II simulation. For small examples, performance close to handwritten Java code has been achieved.

[65] Modeling Real-World Control Systems: Beyond Hybrid Systems

S. Neuendorffer, In Proc. Winter Simulation Conference (WSC), Washington, DC, USA, December 5--8, 2004.

Abstract: Hybrid system modeling refers to the construction of system models combining both continuous and discrete dynamics. These models can greatly reduce the complexity of a physical system model by abstracting some of the continuous dynamics of the system into discrete dynamics. Hybrid system models are also useful for describing the interaction between physical processes and computational processes, such as in a digital feedback control system. Unfortunately, hybrid system models poorly capture common software architecture design patterns, such as threads, mobile code, safety, and hardware interfaces. Dealing effectively with these practical software issues is crucial when designing real-world systems. This paper presents a model of a complex control system that combines continuous-state physical system models with rich discrete-state software models in a disciplined fashion. We show how expressive modeling using multiple semantics can be used to address the design difficulties in such a system.

[66] Automated Task Allocation for Network Processors

W. Plishker, K. Ravindran, N. Shah, K. Keutzer. In Proc. Network System Design Conference, October, 2004, pp. 235-245.

Abstract: Network processors have great potential to combine high performance with increased flexibility. These multiprocessor systems consist of programmable elements, dedicated logic, and specialized memory and interconnection networks. However, the architectural complexity of the systems makes programming difficult. Programmers must be able to productively implement high performance applications for network processors to succeed. Ideally, designers describe applications in a domain specific language (DSL). DSLs expedite the development process by providing component libraries, communication and computation semantics, visualization tools, and test suites for an application domain. An integral aspect of mapping applications described in a DSL to network processors is allocating computational tasks to processing elements. We formulate this task allocation problem for a popular network processor, the Intel IXP1200. 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 DiffServ.

[67] Designing Distributed Diagnosers for Complex Physical Systems

I. Roychoudhury, G. Biswas, X. Koutsoukos and S. Abdelwahed, Intl. Workshop on Principles of Diagnosis (DX-05), (in publication) Monterey, CA, June 2005.

Abstract: Online diagnosis methods require large computationally expensive diagnosis tasks to be decomposed into sets of smaller tasks so that time and space complexity constraints are not violated. This paper defines the distributed diagnosis problem in the Transcend qualitative diagnosis framework, and then develops heuristic algorithms for generating a set of local diagnosers that solve the global diagnosis problem without a coordinator. Two versions of the algorithm are discussed. The time complexity and optimality of these algorithms are compared and validated through experimental results.

[68] A Distributed Algorithm for Acoustic Localization Using a Distributed Sensor Network

P. Schmidt, I. Amundson, K.D. Frampton, Journal of the Acoustical Society of America, Vol. 115, No. 5, Pt. 2, pp. 2578, 2004.

Abstract: An acoustic source localization algorithm has been developed for use with large scale sensor networks using a decentralized computing approach. This algorithm, based on a time delay of arrival (TDOA) method, uses information from the minimum number of sensors necessary for an exactly determined solution. Since the algorithm is designed to run on computational devices with limited memory and speed, the complexity of the computations has been intentionally limited. The sensor network consists of an array of battery operated COTS Ethernet ready embedded systems with an integrated microphone as a sensor. All solutions are calculated as distinct values, and the same TDOA method used for solution is applied for ranking the accuracy of an individual solution. Repeated for all combinations of sensor nodes, solutions with accuracy

equivalent to complex array calculations are obtainable. Effects of sensor placement uncertainty and multipath propagation are quantified and analyzed, and a comparison to results obtained in the field with a large array and a centralized computing capability using a complex, memory intensive algorithm is included.

[69] Optimal Control for a class of Stochastic Hybrid Systems

L. Shi, A. Abate, S. Sastry, In Proc. International Conference on Decision and Control, Atlantis, Bahamas, December 2004.

Abstract: In this paper, an optimal control problem over a "hybrid Markov Chain" (hMC) is studied. A hMC can be thought of as a traditional MC with continuous time dynamics pertaining to each node; from a different perspective, it can be regarded as a class of hybrid system with random discrete switches induced by an *embedded MC*. As a consequence of this setting, the index to be maximized, which depends on the dynamics, is the expected value of a non deterministic cost function. After obtaining a closed form for the objective function, we gradually suggest how to device a computationally tractable algorithm to get to the optimal value. Furthermore, the complexity and rate of convergence of the algorithm is analyzed. Proofs and simulations of our results are provided; moreover, an applicative and motivating example is introduced.

[70] Generative Components for Hybrid Systems Tools

J. Sprinkle, 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

Abstract: Generative techniques, while normally associated with programming languages or code generation, may also be used to produce non-executable artifacts (e.g., configuration or toolchain artifacts). Coupled with domain-specific modeling, generative techniques provide a way to consolidate toolchains and complex domains into a relatively compact space, by providing generators that produce artifacts in the appropriate semantic domain. This paper describes the motivation and usage of one such environment, in the domain of hybrid systems, as well as discussion and goals for future research.

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

J. Sprinkle, A. D. Ames, S. S. Sastry, 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05), (submitted for publication), Seville, Spain, Dec., 12--15, 2005.

Abstract: 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.

[72] A Paradigm for Teaching Modeling Environment Design

J. Sprinkle, J. Davis, G. Nordstrom, 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.

Abstract: Model-Integrated Computing (MIC) is a generic term for the practice of coupling models of a complex system with the execution of that system. Although MIC has been in use for years by experts who learn its techniques in an ad hoc manner, it is only recently that system modeling-as a science-has begun to be taught as a subject. This paper presents a paradigm in use in a course designed specifically to teach the design of domain-specific modeling environments. The work describes how this paradigm grows throughout the course to complement the expanding nomenclature, mapping technologies, visitor and object-oriented technologies, and design decisions encountered by the students.

[73] Deciding to Land a UAV Safely in Real Time

J. Sprinkle, J. M. Eklund, S. S. Sastry, In Proc. American Control Conference (ACC) 2005, (in publication), Portland, OR, Jun., 8--10, 2005.

Abstract: The difficulty of autonomous free-flight of a fixedwing 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 will give 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.

[74] Encoding Aerial Pursuit/Evasion Games with Fixed Wing Aircraft into a Nonlinear Model Predictive Tracking Controller

J. Sprinkle, J. M. Eklund, H. J. Kim, S. S. Sastry, IEEE Conference on Decision and Control, pp. 2609--2614 , Dec., 2004.

Abstract: Unmanned Aerial Vehicles (UAVs) have shown themselves to be highly capable in intelligence gathering, as well as a possible future deployment platform for munitions. Currently UAVs are supervised or piloted remotely, meaning that their

behavior is not autonomous throughout the flight. For uncontested missions this is a viable method; however, if confronted by an adversary, UAVs may be required to execute maneuvers faster than a remote pilot could perform them in order to evade being targeted. In this paper we give a description of a non-linear model predictive controller in which evasive maneuvers in three dimensions are encoded for a fixed wing UAV for the purposes of this pursuit/evasion game.

[75] Toward Design Parameterization Support for Model Predictive Control

J. Sprinkle, J. M. Eklund, S. S. Sastry, IEEE 4th International Conference on Intelligent Systems Design and Application, IEEE, IEEE Press, Budapest, Hungary, Aug., 26--28, 2004.

Abstract: Research into the autonomous behavior of Unmanned Aerial Vehicles (UAVs) requires concise and dependable specification techniques in order to provide behavioral descriptions for the controllers of these aircraft. Practical issues with autonomous aircraft involve the safety and reliability of the controller (e.g., guarantee of stability), as well as verification of the high-level intention of the autonomous behavior. Since most behaviors are implemented orthogonally to their high-level specification, improvements in the ability to rapidly specify the models that govern the behavior are certainly welcome. This paper describes the framework for decreasing the abstraction required to specify the behavior of an autonomous controller implemented through model predictive control.

[76] A Domain-Specific Visual Language for Domain Model Evolution

J. Sprinkle, G. Karsai, J. Vis. Lang. and Comp., vol. 15, no. 3-4, pp. 291-307, Jun., 2004.

Abstract: Domain-specific visual languages (DSVLs) are concise and useful tools that allow the rapid development of the behavior and/or structure of applications in well-defined domains. These languages are typically developed specifically for a domain, and have a strong cohesion to the domain concepts, which often appear as primitives in the language. The strong cohesion between DSVL language primitives and the domain is a benefit for development by domain experts, but can be a drawback when the domain evolves---even when that evolution appears insignificant. This paper presents a domain-specific visual language developed expressly for the evolution of domain-specific visual languages, and uses concepts from graph-rewriting to specify and carry out the transformation of the models built using the original DSVL.

[77] Using the Hybrid Systems Interchange Format to Input Design Models to Verification & Validation Tools

J. Sprinkle, O. Shakernia, R. Miller, S. S. Sastry, IEEE Aerospace Conference, Big Sky, MT, Mar., 2005.

Abstract: The domain of hybrid systems lacks the set of mature tools which can reliably apply verification and validation (V&V) techniques for all kinds of systems. Currently, no single tool supports analysis, simulation, verification, validation, and code synthesis of controllers for hybrid systems. As such, it becomes necessary to depend on several tools

for analysis of different aspects of the system. This paper , describes the utilization of the definition of a system in more than one toolsuite, while reducing the required effort for the same engineers that design the controllers to interface to the V&V toolsuites.

[78] Platform Modeling and Model Transformations for Analysis

T. Szemethy, G. Karsai, Journal of Universal Computer Science, vol. 10, no. 10, pp 1383-1406, 2004.

Abstract: The model-based approach to the development of embedded systems relies on the use of explicit models in the design process. If these models faithfully represent the components of the system with respect to their properties as well as their interactions, then they can be used to predict the dynamic behavior of the system under construction. In this paper we argue for modeling the execution platform that facilitates the component interactions, and show how models of the application and the knowledge of the platform can be used to translate system configurations into another abstract formalism (timed automata, in our case) that allows system verification through model checking.

[79] 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, T. J. Koo, X. Koutsoukos, D. Schmidt, ACM Transactions on Embedded Systems, (in publication), 2005.

Abstract: 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.

[80] Experiments on the Decentralized Vibration Control with Networked Embedded Systems

T. Tao, K.D. Frampton, Presented at the 2004 ASME International Mechanical Engineering Conference and Exposition, Anaheim CA, November 2004.

Abstract: The early promise of centralized active control technologies to improve the performance of large scale, complex systems has not been realized largely due to the

inability of centralized control systems to "scale up"; that is, the inability to continue to perform well when the number of sensors and actuators becomes large. Now, recent advances in Micro-electro-mechanical systems (MEMS), microprocessor developments and the breakthroughs in embedded systems technologies, decentralized control systems may see these promises through. A networked embedded system consists of many nodes that possess limited computational capability, sensors, actuators and the ability to communicate with each other over a network. The aim of this decentralized control system is to control the vibration of a structure by using such an embedded system backbone. The key attributes of such control architectures are that it be scalable and that it be effective within the constraints of embedded systems. Toward this end, the decentralized vibration control of a simply supported beam has been implemented experimentally. The experiments demonstrate that the reduction of the system vibration is realized with the decentralized control strategy while meeting the embedded system constraints, such as a minimum of inter-node sensor data communication, robustness to delays in sensor data and scalability.

[81] Proceedings of the 4th OOPSLA Workshop on Domain-Specific Modeling (DSM'04)

J. Tolvanen, J. Sprinkle, M. Rossi, eds., Jyvavaskyla, Finland, 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), University of Jyvavaskyla, Oct., 2004.

Abstract: 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 fourth workshop on Domain-Specific Modeling, following the encouraging experiences from the earlier workshops at past OOPSLA conferences (Tampa 2001, Seattle 2002 and Anaheim 2003). During the time the DSM workshops have been organized, interest in domain-specific modeling languages, metamodeling and supporting tools has increased greatly.

[82] Towards Generation of Efficient Transformations

A. Vizhanyo, A. Agrawal, F. Shi, Generative Programming and Component Engineering, 3rd International Conference, October, 2004, LNCS 3286, pp 298-316, 2004.

Abstract: In this paper we discuss efficiency related constructs of a graph rewriting language, called Graph Rewriting and Transformation (GReAT), and introduce a code generator tool, which together provide a programming framework for the specification and efficient realization of graph rewriting systems. We argue that the performance problems frequently associated with the implementation of the transformation can be significantly reduced by partial evaluation and adopting language constructs that allow algorithmic optimizations.

[83] Distributed Control of Thermoelectric Coolers

D. G. Walker, K. D. Frampton, R.D. Harvey, 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.

Abstract: Thermoelectric refrigeration has been studied for use in electronics cooling applications. Because of their low efficiency, a significant amount of additional heat is produced that must be removed from the device by passive means. However, even well-designed passive heat removal systems are faced with physical limitations and can not dissipate additional energy. Therefore, thermoelectric coolers often are not a viable alternative to chip cooling. Distributed control refers to a concept where individual devices operate based on their local condition and that of their nearest neighbors. With this approach a large number of sensors and actuators can be bundled with minimal compute power and communication. In the case of electronics cooling, a thermoelectric cooler can be constructed as many separate coolers, each with its own thermocouple and minimal control system. With this architecture, only those regions that generate heat will be cooled thermoelectrically reducing the impact on the heat removal system and increasing the viability of thermoelectric coolers. Through analytic TEC heat models, the present work seeks to evaluate the possibility of using distributed controlled thermoelectric coolers for microelectronics cooling applications. Results indicate that TEC coolers can provide a 2-fold increase in efficiency when distributed control is used for nonuniformly heated chips.

[84] The Design and Application of Structured Types in Ptolemy

Y. Xiong, E.A. Lee, X. Liu, Y. Zhao, L.C. Zhong, IEEE Int. Conf. on Granular Computing (Grc 2005), (in publication), Beijing, China, July 25-27, 2005.

Abstract: 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.

[85] Dynamic Dataflow Modeling in Ptolemy II

G. Zhou, Master's Report, Technical Memorandum No. UCB/ERL M05/2/, University of California, Berkeley, December 21, 2004.

Abstract: Dataflow process networks are a special case of Kahn process networks (PN). In dataflow process networks, each process consists of repeated firings of a dataflow actor, which defines a quantum of computation. Using this quantum avoids the complexities and context switching overhead of process suspension and resumption incurred in most implementations of Kahn process networks. Instead of context switching, dataflow process networks are executed by scheduling the actor firings. This scheduling can be done at compile time for synchronous dataflow (SDF) which is a particularly restricted case with the extremely useful property that deadlock and boundedness are decidable. However, for the most general dataflow, the scheduling has to be done at run time and questions about deadlock and boundedness cannot be statically answered. This report describes and implements a dynamic dataflow (DDF) scheduling algorithm under Ptolemy II framework based on original work in Ptolemy Classic. The design of the algorithm is guided by several criteria that have implications in practical implementation. We compared the performance of SDF, DDF and PN. We also discussed composing DDF with other models of computation (MoC). Due to Turing-completeness of DDF, it is not easy to define a meaningful iteration for a DDF submodel when it is embedded inside another MoC. We provide a suite of mechanisms that will facilitate this process. We give several application examples to show how conditionals, data-dependent iterations, recursion and other dynamic constructs can be modeled in the DDF domain.

2.3. Project Training and Development

As part of setting up CHESS (the new UCB Center for Hybrid and Embedded Software Systems), we have created a CHESS Software Lab, which is focused on supporting the creation of publication-quality software supporting embedded systems design. The lab is physically 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.

Course: Structure and Interpretation of Signals and Systems (UCB, EECS 20N)

Instructor: Prof. Edward A. Lee
Prof. Pravin Varaiya

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.

Graduate Courses

Several graduate courses were taught in the past year 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)

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)

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: Theory of Automata, Formal Languages, and Computation (VU, CS 352)

Instructor: Prof. Xenofon Koutsoukos

Structures, automata and formal language theory, process modeling with finite state automata, supervisory control theory, controllability and supervision, supervisory control under partial observation, modular and hierarchical supervisory control, supervisory control of real-time systems, fault diagnosis of discrete event systems, and modular diagnosis approaches.

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: Dr. James Davis
Dr. Greg Nordstrom

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 increase diversity in the graduate school pipeline by affirming students' motivation for graduate study and strengthening their qualifications.

SUPERB-IT participants spent eight weeks at UC Berkeley during the summer of 2004 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 three SUPERB-IT students in 2004, and organized projects (described below in the abstracts from their papers) in hybrid systems theory, wireless sensor networks, and aerial vehicle simulation. 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 2004 were Basil Etefia, Rafael Garcia, and Elizabeth Fatusin. The three 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. Three 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/2004/superb.htm.

Project: Routing Protocols for Wireless Sensor Networks

Student: Basil Etefia—*Loyola Marymount University*

> This paper presents an improvement on the implementation of information routing capabilities in ad hoc wireless sensor networks. Improving the protocols used by each sensor node can increase the network's localization and power conservation abilities. Furthermore, using a more efficient algorithm will improve the overall effectiveness of the entire network in terms of power efficiency. Using novel and creative schemes to generate shortest paths for information routing from source to destination nodes, we have implemented an approach to limit the inefficiencies of routing protocols used by sensor networks for information transfer.

Project: Multi-view Configuration of Flight Dynamic Playback

Student: Elizabeth (Beth) Fatusin—*Cornell University, via Ohlone Community College*

> Unmanned Aerial Vehicles (UAVs) have proven to be highly capable in reconnaissance and intelligence gathering, as well as a possible future operation platform for weapons.

Currently UAVs are supervised or piloted remotely, meaning that their behavior is not autonomous throughout the flight. Consequently, instructions are being passed to the UAVs commanding the maneuvers to execute and perform to pursue their adversary or to evade being targeted. In recent years, the security level of these aircrafts has become a critical issue because these aircraft are used for combat missions and other secure operations. For this reason and many others, designing a dual display of the pursuer/evader of UAVs using a flight simulator and displaying in a singular window, appears to be a good idea. In this paper, we give a description of how the data is being collected and transformed into friendly data for viewing its actual maneuvers as a three dimension weapon using a flight simulator and why it is important and safe to do so.

Project: Singular Event Detection

Student: Rafael S. García—*University of Puerto Rico at Mayagüez*

The main objective in the study of event location is to find when the solution to an Ordinary Differential Equation (ODE) crosses a switching surface. Due to the fact that it is almost always impossible to find the exact solution to an arbitrary ODE, the event must be located with a numerical approximation to the exact solution. Since approximate solutions are by nature inexact, the question is whether the real solution displays the same qualitative behavior as its numerical approximation. This raises the need for a number that gauges the validity of this approximation, i.e., a condition number. The purpose of this research is to show how numerical approximations of the solutions of simple systems–linear systems–with simple switching surfaces–hyperplanes–can lead to incorrect event detection. Motivated by these examples, a candidate condition number will be proposed in an attempt to quantify this failure.

Plans for 2005

The SUPERB program is dedicated to providing undergraduate students with the opportunities and experiences of research. 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 Chess/ITR philosophy 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 2005 SUPERB program a set of projects which intersect in some areas, and 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 everyone in the future. Our goal is to use this as a benefit for the student over the summer, where they will learn how to think in terms of multiple goals at once, and achieve results that can be interesting to experts in more than one field. The projects broadly fit into the areas of hybrid systems, sensor networks, computer vision, VLSI design, and simulation.

Six graduate student mentors have been identified to facilitate the process, and the operation is being coordinated and directed by Dr. Jonathan Sprinkle (a postdoc under the supervision of Professor Shankar Sastry). Although the students are working together and interacting extensively, each will be responsible for a single project's full completion. The students and mentors have already been paired for what promises to be an exciting summer.

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/fountain/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 2004 for seven 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 five 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.

Project: Distributed System Implementation of the Kuvangu Running Frog's Calling Behavior

Students: Praveen Mudindi—*Alabama A&M University*
Efosa Ojomo—*Vanderbilt University*

Male frogs attract their female mates over great distances by having collective synchronized chirps. These collective chirps increase the intensity and the possibility of being heard at great distances. Initially, the frogs chirp at different rates and out of phase, but after a while, they all synchronize their chirps to those of the frog(s) with the highest chirp intensity, producing a much louder and unified chirp. The objective of this project is to mimic this behavior using a distributed, real-time, embedded (DRE) system architecture. This platform has made it possible to write a generic program that will perform different functions on different systems. One main goal of the project is to get acquainted with embedded systems and sensor networks. Sensor networks sense their environments and react according to the program specifications they have been given.

Project: Maze Discovery and Chutes & Ladders

Students: Jameson Porter—*Tennessee Tech University*
Mary Hilliard—*University of Tennessee, Chattanooga*

As the need for embedded systems increase in society, capabilities for system development must expand to better define the constraints and physical aspects of the system. By using Model Integrated Computing (MIC), engineers can better understand the embedded system as a whole as well as easily evolve the system according to its environment. The goal of both the original and additional project is to gain a better

understanding of embedded systems and model integrating computing techniques by implementing a maze mapping and Chutes & Ladders playing Lego robot, respectively.

Project: Tic-Tac-Toe Implementation

Students: Shirley (Xue) Li—*Massachusetts Institute of Technology*
Sinithro (Miguel) Taveras—*University of Florida*

From medical procedures to space explorations, embedded, autonomous computing is becoming vital in many safety-critical applications. At the same time, many industries are looking to use the model-integrated approach for the analysis of complex systems. As a part of the SIPHER program, our initial objective was to familiarize ourselves with GME and its underlying principles. For the first half of the summer, using the SMOLES paradigm, our group worked on developing control algorithms for a LEGO Mindstorm robot. The specific task included implementing low-level, reactive control algorithms, and higher-level, planning activities. For the second half of the summer, our group's general objective was to further explore the capabilities of Mindstorm robots. Our particular task involved implementing a Tic-Tac-Toe gaming algorithm and perfecting navigational movements.

Project: A Model-Integrated Computing Approach to Embedded Controller Design

Students: Christopher Beers—*Vanderbilt University*

The MACS Lab at Vanderbilt University has a three tank fluid system, which is used as a test bed system for modeling, monitoring, control, estimation, and fault detection/adaptation of a hybrid system. The sensors and actuators in the system are operated by a Network Capable Application Processor (NCAP) and a Smart Transducer Interface Module (STIM). The NCAP has a small web server, and all commands are sent as through a HTTP/CGI interface to the STIM, which either reads a physical measurement or sends out an actuator signal to the system. An initial implementation of a controller was written as platform dependent C code. The control software was handwritten and required detailed knowledge of the actual NCAP configuration. The first step for this summer project was to use the ACE framework to write a wrapper that adheres to the IEEE 1451.2 standard for the NCAP communication. This was completed by mid-summer. The next step was to implement the control software using the OCP (Open Control Platform) and modify the FACT (Fault-Adaptive Control Technology) architecture so that a controller could be built using multiple finite state machines. This also required making changes to the graphical interfaces of GME. This simplifies the controller creation process, and allows users to easily create and run experiments on the three-tank system.

Project: Visual Tracking and Control

Students: John Williamson, III—*North Carolina A&T University*
Trevor Brown—*Middle Tennessee State University*

The first part was to visually track a robot going through a maze using a web camera to capture the images. In order to achieve this we had to start off with the basics, and that was to represent the relationship between the personal computer (PC) and camera using

the Generic Modeling Environment (GME). Once we had a good visual model of the camera and PC's components, the data flow between the two using input and output ports, and timers to start the data flow, we interpreted it and produced our main C++ files. We then hand-coded supporting functions, resulting in the achievement of having a PC controlled camera tracking a moving robot, keeping the target in the middle of our display screen. The second half of our project also involved tracking a moving robot. This time, instead of having a digital camera doing the tracking, we constructed a laser holding robot to do the tracking. With the now non-moving camera positioned to view our entire maze, we gathered information from its images to find the X and Y coordinates of the moving robot and our laser. With this information, Java and C++ code coupled with the Lejos firmware on the RCX, we had a system that controlled our built robot to move and keep the laser pointed on the moving robot captured by the web camera.

Plans for 2005

Similarly to the SUPERB program, the SIPHER is also 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 have chosen a set of research projects that for the 2005 SIPHER program that reflect this goal.

The selected research projects/topics are as follows:

* *Visually guided, remotely controlled robots.* Students in this project will use LEGO robots equipped with short-range communication devices, and a visual tracking system with a desktop workstation. The goal is to help an operator to track and control robots via a camera.
* *Process control applications.* Students in this project will learn about hybrid modeling of a small industrial plant and will develop fault-adaptive controllers for it.
* *Sensor networks for monitoring and control.* Students here will learn about networks of cameras that will be used to track moving objects.
* *Embedded controllers.* Students here will work on active vibration control problems.
* *Model-based tools for low-end microcontrollers.* Students will do research on how to build model-based software development tools for 8-bit microcontrollers.

Five graduate student mentors have been identified to facilitate the process, and the operation is being coordinated and directed by Mr. Efosa Ojomo (a former SIPHER student, now a graduate engineer). The students will be working in small teams, under the tutelage of the graduate students. The students and mentors have already been paired for what promises to be an exciting summer.

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 (Toshiba plant, Saturn plant, and NASA Marshall Space Flight Center).

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

- 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

- 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

- 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.

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.

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.

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 4) 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 (8/25/04) 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, r1.3.3, can be downloaded from: http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#GREAT

- HyVisual 5.0 is a hybrid system visual modeler, was released in March of 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 5.0-alpha is available at http://ptolemy.eecs.berkeley.edu/hyvisual/index.htm

- Metropolis (1.02) consists of an infrastructure, a tool set, and design methodologies for various application domains. The infrastructure provides a mechanism such that heterogeneous components of a system can be represented uniformly and tools for formal methods can be applied naturally. Metropolis 1.02 is available at http://chess.eecs.berkeley.edu/chess/forum/6.html

- Ptolemy II 5.0 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, dataflow, 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 of data, math, providing matrix and vector math and signal processing functions, and graph, providing graph-theoretic manipulations. The Ptolemy II V.5 beta release is available at http://ptolemy.eecs.berkeley.edu/ptolemyII/ptII5.0/index.htm

- 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). V. 2.21 can be downloaded from http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#UDM

- VisualSense is a visual editor and simulator for wireless sensor network systems. Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, 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 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-4.0.1 is available as part of the Ptolemy II 4.0.1 release at http://ptolemy.eecs.berkeley.edu/ptolemyII/ptII4.0/index.htm.

4. Contributions

This section summarizes the major contributions during this reporting period.

4.1. Within Discipline

4.1.1. Hybrid Systems Theory

- We have defined an operational semantics for hybrid systems which was an invited plenary speech at the *Hybrid Systems: Computation and Control* conference in 2005, and poised to influence 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 developd a theory of a functorial view of hybrid systems which enables elegant characterization of Zeno and other qualitative properties of hybrid systems.

- We have developed an introductory theory on a condition number for hybrid system simulations.

- We have improved on the best known algorithms for finding strategies for the control of stochastic hybrid systems. These have been applied to applications as diverse as air to air combat for UAVs, coordinated operations of UUVs and communication networks.

- We have constructed 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 have begun the implementation of a broad initiative to support toolchains in hybrid systems under semantic anchoring and model transformations.

4.1.2. Model-Based Design

- Applying our ongoing work on metamodeling, we have continued development on semantic anchoring for model-based development.

- 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.

4.1.3. Advanced Tool Architectures

- We have extended modularity mechanisms (classes, subclasses, inheritance, interfaces) for actor-oriented design, complementing the prevailing object-oriented methods to also include types and subtypes for modeling-time template based design.

- 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 have developed a modeling environment for processor instruction set decoding and used it to simulate the MIPS, PowerPC, and ARM processors.

- We have continued to investigate interests in fault-tolerant systems by developing new modeling languages which simulate and trace faults in a system.

4.1.4. Experimental Research

- We have used our work on safe set calculations to successfully land UAVs (a joint project with Northrup Grumann) in flight tests

- We have used our results in model predictive control to develop air to air combat strategies for a UAV to successfully defeat a manned aircraft (joint project with Boeing). The UAV successfully targeted the manned aircraft (of much higher speed and performance and piloted by an expert test pilot) several times.

- 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 in very high profile shooter localization schemes.

- We have shown application results on the connectivity of large-scale sensor networks.

- We have developed new programming models for sensor networks that build on the popular TinyOS models.

- We have developed programming models for bit streaming applications that use sketching of strategies with code generation.

- We have continued adapting sensor networks and wireless technology to address elder care diagnosis and monitoring issues.

- We have used control theory and software based on the Time-Triggered Architecture and Giotto language to control Unmanned Aerial Rotorcraft, and also integrated Giotto with Metropolis.

4.2. Other Disciplines

- We developed new efficient algorithms for solving stochastic games, which have applications in other fields such as economics.

- We contributed to scientific interdisciplinary information sharing through collaboration and major contribution to the framework of the Kepler Scientific Workflow project.

- We continued to contribute to the simulation and understanding of biological processes, such as the piliation of *E. coli*, and the production of the *B. subtilis* antibiotics. We have now moved on to other mechanisms in other organisms.

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. 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.

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. Elder care, soft walls and distributed diagnosis are a few examples of this impact. In the domain of transportation systems, our research is improving safety in cars. 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, transportation, cybersecurity, and transportation.

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 ith system, and Ω 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 μ. 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.

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

June 19, 2007

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2006 – MAY 31, 2007

Contents

Contents 2
1. Participants 5
1.1. People 5
1.2. Partner Organizations: 7
1.3. Collaborators: 7
2. Activities and Findings 10
2.1. Project Activities 10
2.1.1. Hybrid Systems Theory 10
2.1.1.a. Deep Compositionality 11
Compositional Specification of Behavioral Semantics 11
Composing Different Models of Computation in Ptolemy II and Kepler 11
Hybrid Geometric Reduction of Hybrid Systems 12
2.1.1.b. Robust Hybrid Systems 12
Fault Tree Analysis 12
2.1.1.c. Computational Hybrid Systems 12
Concurrent Reachability Games 12
Zeno Conditions 12
2.1.1.d. Stochastic Hybrid Systems 13
Reachability analysis for controlled discrete time stochastic hybrid systems 13
Bounding Error for Stochastic Approximations 13
2.1.2. Hybrid Systems – Components for Embedded Systems 13
2.1.2.a Building Efficient Simulations from Hybrid Bond Graph Models 13
2.1.2.b Concurrency formalisms 15
Code Generation Frameworks 15
Feedback in Strictly Causal Systems 15
2.1.2.c Application to system level design 16
An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures 16
Synthesis of task and message activation models in real-time distributed automotive systems 16
Optimizing end-to-end latencies by adaptation of the activation events in distributed automotive systems 17
2.1.3. Model-based Design 17
2.1.3.a Composition of Domain Specific Modeling Languages 17
2.1.3.b Model Transformations 19
2.1.4 Experimental Research 20
2.1.4.a. Embedded Control Systems 21
Decentralized control of unmanned underwater vehicles 21
2.1.4.b. Embedded Software for National and Homeland Security 21
Autonomous Ground Vehicles 21
Real-time Computer Vision 21
2.1.4.c. Networks of Distributed Sensors 22
VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems 22

Viptos: a Programming Models for Sensor Networks *22*
Control of Communication Networks *23*
2.2. Project Findings 23
3. Outreach 56
3.1. Project Training and Development 56
3.2. Outreach Activities 56
3.2.1. Curriculum Development for Modern Systems Science (MSS) 56
Undergrad Course Insertion and Transfer 57
Course: Structure and Interpretation of Signals and Systems (UCB, EECS 20N) http://ptolemy.eecs.berkeley.edu/eecs20/ *57*
Graduate Courses 58
Course: Autonomous Systems: Algorithms and Implementation (UCB, EECS 290n) Instructor: Dr. Jonathan Sprinkle, Prof. S. Shankar Sastry *58*
Course: Embedded System Design: Models, Validation, and Synthesis (UCB EE249) *58*
Course: Foundations of Hybrid and Embedded Systems (VU, CS 376) *58*
Course: Model Integrated Computing (VU, CS 388 / EE 395) *59*
Course: Real-Time Systems (VU, EECE 353-01) *59*
Course: Automated Verification (VU, EECE 315) *59*
Course: Automated Verification (VU, EECE 375) *59*
3.2.2. SUPERB-IT Program 60
Project: Highway Traffic Flow Analysis and Control *61*
Project: Tool for probabilistic safety verification of stochastic hybrid systems *61*
Project: Autopilot for an Ultra-Light, Flying Wing *62*
Project: Multihop Routing Simulation of TinyOS-Based Wireless Sensor Networks in Viptos *62*
3.2.3. Summer Internship Program in Hybrid and Embedded Software Research (SIPHER) Program 63
Project: Radio Controlled Car Controller *64*
Project: Hybrid System Modeling / Fault Diagnosis *64*
Project: Controlling Lego Robots Using a Synchronous-Reactive Model of Computation *64*
Project: Exploring with Lego Robots *65*
4. Publications and Products 66
4.1. Journal Publications 66
4.2. Conference Papers 66
4.3. Books, Reports, and Other One-Time Publications 73
4.4. Dissemination 75
4.4.1. Software Maturation 75
Industry Technology Transition 76
4.4.2. Working Groups and Standards 76
4.4.3. "Foundations" The 2006-2007 Chess seminar series 76
4.4.4. Workshops and Invited Talks 78
6th OOPSLA Workshop on Domain-Specific Modeling *78*
2007 International Symposium on Code Generation and Optimization (CGO) *79*
Real-Time and Embedded Technology and Applications Symposium (RTAS) *79*

6th ACM & IEEE Conference on Embedded Software (EMSOFT'06) *79*
4.4.5. General Dissemination 79
4.5. Other Specific Product 79
5. Contributions 82
5.1. Within Discipline 82
5.1.1. Hybrid Systems Theory 82
5.1.2. Model-Based Design 83
5.1.3. Advanced Tool Architectures 84
5.1.4. Experimental Research 84
5.2. Other Disciplines 85
5.3. Human Resource Development 85
5.4. Integration of Research and Education 86
5.5. Beyond Science and Engineering 86

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:

AHMAD BAHAI (UC BERKELEY, EECS)
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:

AKOS LEDECZI (VANDERBILT)
JONATHAN SPRINKLE (UC BERKELEY)

GRADUATE STUDENTS:

ALESSANDRO ABATE (UC BERKELEY)
AARON AMES (UC BERKELEY)
SAURABH AMIN (UC BERKELEY)
DANIEL BALASUBRAMANIAN (VANDERBILT)
ARINDAM CHAKRABARTI (UC BERKELEY)

DENNIS CHANG (UC BERKELEY)
KRISHNENDU CHATTERJEE (UC BERKELEY)
ELAINE CHEONG (UC BERKELEY)
ABHIJIT DAVARE (UC BERKELEY)
DOUGLAS DENSMORE (UC BERKELEY)
MATTHEW EMERSON (VANDERBILT)
AARONG FLEETWOOD (VANDERBILT)
THOMAS HUINING FENG (UC BERKELEY)
SUMITRA GANESH (UC BERKELEY)
ARKEDEB GHOSAL (UC BERKELEY)
GRAHAM HEMMINGWAY (VANDERBILT)
ETHAN JACKSON (VANDERBILT)
FARINAZ KOUSHANFAR (UC BERKELEY)
ALEXANDER KURZHANSKIY (UC BERKELEY)
CHRISTINA LEE (VANDERBILT)
JONGHO LEE (UC BERKELEY)
DAVID P. MANDELIN (UC BERKELEY)
SLOBODAN MATIC (UC BERKELEY)
ELEFTHERIOS MATSIKOUDIS (UC BERKELEY)
TREVOR MEYEROWITZ (UC BERKELEY)
BHARATHWAJ MUTHUSWARMY (UC BERKELEY)
TAKASHI NAGATA (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)
TRIPTI SAXENA (VANDERBILT)
RYAN T THIBODEAUX (VANDERBILT)
GUOGIANG WANG (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:

ZULHIMI AHMAD (VANDERBILT)
ANGELINE BROWN (VANDERBILT)
LOUISE COLLINS (VANDERBILT)
DOMINIQUE DUNCAN (UC BERKELEY/UNIVERSITY OF CHICAGO)

TARRELL EZELL (VANDERBILT)
DANIELLE JONES (VANDERBILT)
CHARLES LEFONT (VANDERBILT)
KENNETH MCPHERSON (VANDERBILT)
NANDITA MITRA (UC BERKELEY/RUTGERS)
NASHLIE SEPHUS (UC BERKELEY/MISSISSIPPI STATE)
HEATHER TAYLOR (UC BERKELEY/UNIVERSITY OF VERMONT)
STEPHANIE WRIGHT (VANDERBILT)

TECHNICAL STAFF, PROGRAMMERS:

GYORGY BALOGH (VANDERBILT)
CHRISTOPHER BROOKS (UC BERKELEY)
PHILLIP LOARIE (UC BERKELEY)
DAN STEWARD (VANDERBILT)
MARY STEWART (UC BERKELEY)
BRIAN WILLIAMS (UC BERKELEY)

BUSINESS ADMINISTRATORS:

ROBERT BOXIE (VANDERBILT, SIPHER COORDINATOR)
MICHELE M. CODD (VANDERBILT)
TRACEY RICHARDS (UC BERKELEY)

1.2. Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY
VANDERBILT
MEMPHIS

1.3. Collaborators:

NORIYASU ADACHI (TOYOTA TECHNICAL CENTER)
LUCA DE ALFARO (UC SANTA CRUZ)
HUGO A. ANDRADE (NATIONAL INSTRUMENTS)
J. D. AXELROD (STANFORD)
ILKAY ALTINTAS (SDSC)
AARON AMES (CALTECH)
SHAMIK BANDYOPADHYAY (UC BERKELEY)
MARIA DOMENICA DI BENEDETTO (UNIVERSITA' DELL'AQUILA)
CHAD BERKLEY (UC SANTA BARBARA)
ALBERT BENVENISTE (INRIA)
DIRK BEYER (SIMON FRASER UNIVERSITY)
MARK BIGGIN (LBNL)

THOMAS BRIHAYE (LSV-CNRS & ENS DE CACHAN, FRANCE)
JEFF BURCH (AGILENT)
BENOIT CAILLAUD (INRIA)
LUCA CARLONI (COLUMBIA UNIVERSITY)
MINGHUA CHEN (MICROSOFT RESEARCH)
ADAM DONLIN (XILINX RESEARCH LABS)
LAURENT DOYEN (EPFL)
STEPHEN EDWARDS (COLUMBIA UNIVERSITY)
JOHN EIDSEN (AGILENT)
MIKE EISEN (LBNL)
J. MIKAEL EKLUND (UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY)
MARCO FAELLA (UC SANTA CRUZ)
ANTOON GODERIS (UNIVERSITY OF MANCHESTER)
JEFF GRAY (UNIVERSITY OF ALABAMA (BIRMINGHAM)
ROBERT D. GREGG (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
ESTEN GROTLI (UNIVERSITY OF TRONDHEIM)
PAOLO GIUSTO (GM RESEARCH)
BRUCE HAMILTON (AGILENT)
DAN HIGGINS (UC SANTA BARBARA)
DANIEL IERCAN (POLITECHNICA U. OF TIMISOARA)
ALESSANDRO D'INNOCENZO (UNIVERSITA' DELL'AQUILA)
STAN JEFFERSON (AGILENT)
EFRET JAEGER (SDSC)
MATTHEW JONES (UC SANTA BARBARA)
WOOYOUNG JUNG (DGIST)
TOMOYUKI KAGA (TOYOTA TECHNICAL CENTER)
SRI KANAJAN (GM RESEARCH)
EUNJOO KIM (DGIST)
CHRISTOPH KIRSCH (UNIVERSITY OF SALZBURG)
DOMINIK LANGEN (INFINEON TECHNOLOGIES)
ELIZABETH LATRONICO (BOSCH)
MAN-KIT LEUNG (UC BERKELEY)
JIE LIU (MICROSOFT RESEARCH)
XIAOJUN LIU (SUN MICROSYSTEMS)
JOHN LYGEROS (ETH ZURICH)
BERTRAM LUDASCHER (UC DAVIS)
YI MA (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
THOMAS MANDL (BOSCH)
RUPAK MAJUMDAR (UC LOS ANGELES)
JOSEPH MAKIN (UC BERKELEY)
RADU MARCULESCU (CMU)
RICK MCGEER (HP LABS)
ANDREW MIHAL (UC BERKELEY)
MARK MILLER (HP LABS)
MARCO DI NATALE (GENERAL MOTORS RESEARCH)

STEPHEN NEUENDORFFER (XILINX)
MARIA PARANDINI (POLITECNICO DI MILANO)
ROBERT PASSERONE (UNIVERSITY OF TRENTO ITALY)
CLAUDIO PINELLO (CADENCE BERKELEY LABS)
NIR PITERMAN (EPFL)
GIORDANO POLA (UNIVERSITA' DELL'AQUILA)
MARIA PRANDINI (POLITECNICO DI MILANO)
JEAN-FRANCOIS RASKIN (UNIVERSITE LIBRE DE BRUXELLES)
SANJAY REKHI (CYPRESS SEMICONDUCTOR)
MATTI ROSSI (UNIVERSITY OF JYVASKALA)
MIRKO SAUERMANN (INFINEON TECHNOLOGIES)
NADATHUR SATISH (UC BERKELEY)
JOSEPH SIFAKIS (VERIMAG GRENOBLE)
MYOUNGKYU SOHN (DGIST)
MARIELLE STOELINGA (UNIVERSITY OF TWENTE)
EELCO SCHOLTE (UNITED TECHNOLOGIES RESEARCH CENTER)
CAROLYN TALCOTT (SRI INTERNATIONAL)
ASHISH TIWARI (SRI INTERNATIONAL)
J.-P. TOLVANEN (META CASE)
STAVROS TRIPAKIS (CADENCE BERKELEY LABS)
BEN UPCROFT (UNIVERSITY OF SYDNEY)
RANDALL URBANCE (GM RESEARCH)
GUANQIANG WANG (EECS (UC BERKELEY)
LYNN WANG (UC BERKELEY)
ERIC D.B. WENDEL (SENSIS CORPORATION)
JOHN WRIGHT (UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN)
JOSEPH WYSOCKI (INDEPENDENT CONSULTANT (MALIBU (CALIFORNIA)
YEON-MO YANG (DGIST)
GUANG YANG (UC BERKELEY)
HAKAN YAZAREL (TOYOTA TECHNICAL CENTER)
AVIDEH ZAKOR (UC BERKELEY)
QI ZHU (UC BERKELEY)

2. Activities and Findings

2.1. Project Activities

This is the fifth 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 fifth year were:

- CHESS NSF ITR Fourth Year Site Visit, October 4, 2006, Alexandria, VA. The program and presentations are available at http://chess.eecs.berkeley.edu/conferences/06/FallReview/index.htm
- CHESS Winter Meeting, February 14, 2007, UC Berkeley. The program and the presentations are available at http://chess.eecs.berkeley.edu/conferences/07/WinterReview/program.htm
- 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://www.eecs.berkeley.edu/BEARS/2007/index.html
- A weekly Chess workshop was held at Berkeley. The speakers and topics are listed in Section 4.4.3, 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. Our work in composition of domain-specific modeling languages has supplemented the understanding we have gained in previous years on real-time and functional properties. We also continued our work on 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. We have also been continuing our work in fault-tree analysis, where we generate threes based on dataflow graphs.

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. Our work in game-based reasoning continues to inform how memoryless operations can be used to achieve some measure of optimality while still remaining within objectives of reachability.

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

Compositional Specification of Behavioral Semantics
An emerging common trend in model-based design of embedded software and systems is the adoption of Domain-Specific Modeling Languages (DSMLs). While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of DSML semantics is still a hard problem. 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 (ASM) framework. In [21] we extend the semantic anchoring framework to heterogeneous behaviors by developing a method for the composition of semantic units. Semantic unit composition reduces the required effort from DSML designers and improves the quality of the specification.

Composing Different Models of Computation in Ptolemy II and Kepler
A model of computation (MoC) is a formal abstraction of execution in a computer. There is a need for composing MoCs in e-science. Kepler, which is based on Ptolemy II, is a scientific workflow environment that allows for MoC composition. This paper explains how MoCs are combined in Kepler and Ptolemy II and analyzes which combinations of MoCs are currently possible and useful. It demonstrates the approach by combining MoCs involving dataflow and finite state machines. The resulting classification should be relevant to other workflow environments wishing to combine multiple MoCs. Keywords: Model of computation, scientific workflow, Kepler, Ptolemy II [28].

Hybrid Geometric Reduction of Hybrid Systems
The reduction of mechanical systems with symmetries plays a fundamental role in understanding the many important and interesting properties of these systems. We have been working to generalize this result to a hybrid setting—a formidable obstacle to which is the copious mathematical framework needed to perform reduction. Such a generalization of reduction, therefore, requires a new method for viewing "hybrid objects," one that allows classical mathematical objects and morphisms between these objects to be easily "hybridized." The framework in which we have been working to carry out this generalization is that of category theory, and specifically through the notion of a hybrid object over a category (see [54]). Additional work is included in [76], where we show application to bipedal walkers. Techniques in Routhian reduction are presented in [82], and in [83] we discuss more uses for category theory to provide results in this area.

2.1.1.b. Robust Hybrid Systems

Fault Tree Analysis
In [70] we introduce a model-based approach to heterogeneous system design that enables the automatic generation of fault trees for analyzing system reliability properties. This approach extends our previous work that addressed the generation of fault trees from a dataflow model. In this new context, heterogeneous systems are composed of interacting discrete-time components, such as an electronic feedback controller, and continuous-time components, such as a plant. More recent work in computer-aided fault-tree generation methods is based on functional models of the system to produce a system fault tree automatically. Yet, most of these approaches were not applied to heterogeneous systems. Furthermore, these approaches continued to rely on intuition to create fault trees. Since in this approach fault tree generation is disjoint from the system modeling, consistency problems may arise when the structure and behavior of the system model is not accurately reflected. Our approach is different since we use a model of the system specified as a set of mathematical equations to derive the system fault modes and ultimately produce fault trees for heterogeneous systems.

2.1.1.c. Computational Hybrid Systems

Concurrent Reachability Games
A concurrent reachability game is a two-player game played on a graph: at each state, the players simultaneously and independently select moves; the two moves determine jointly a probability distribution over the successor states. The objective for player 1 consists in reaching a set of target states; the objective for player 2 is to prevent this, so that the game is zero-sum. Our contributions are two-fold. In [63], we discuss that for all epsilon>0, memoryless epsilon-optimal strategies exist. We also present a strategy-improvement (a.k.a. policy-iteration) algorithm for concurrent games with reachability objectives. Other work in [65] studies Nash Equilibria for memoryless strategies. This work, along with [67][62][63], addresses techniques and results for these applications.

Zeno Conditions
In [80], and in more detail in [91], 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.

Zeno systems are also discussed in [92], with respect to the stability properties of a class of Zeno equilibria.

2.1.1.d. Stochastic Hybrid Systems

Reachability analysis for controlled discrete time stochastic hybrid systems
A model for discrete time stochastic hybrid systems whose evolution can be influenced by some control input is proposed in this paper. With reference to the introduced class of systems, a methodology for probabilistic reachability analysis is developed that is relevant to 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 characterizing through some optimal cost function the set of initial conditions for the system such that safety is guaranteed with sufficiently high probability. The proposed methodology is applied to the problem of regulating the average temperature in a room by a thermostat controlling a heater (see [52]).

Bounding Error for Stochastic Approximations
The work in [85] 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.

2.1.2. Hybrid Systems – Components for Embedded Systems

2.1.2.a Building Efficient Simulations from Hybrid Bond Graph Models

Modern engineering systems are complex and made up of a large number of interacting components with nonlinear hybrid behaviors. This makes the building of accurate and computationally efficient simulation models a very challenging task. Recently, researchers and practitioners have adopted component- and actor-oriented frameworks for systematic construction of large models of complex systems. Such frameworks consist of mathematical

models for specifying individual component behavior and formal models of computation for defining component interactions. Simulation models are derived by representing the component behavior models as computational blocks, and the interaction models define the syntax and semantics of the connections between the blocks.

We have adopted the Hybrid Bond Graph (HBG) paradigm, an extension of the Bond Graph (BG) modeling language, for component-based modeling of embedded systems. It is a domain-independent topological modeling language that captures interactions among the different processes that make up the system and can be very effective in parameterized component-based modeling of hybrid systems. The challenge we face is in translating these models to computationally efficient simulation models.

The causal structure inherent in BG models provides the basis for conversion of BGs to efficient computational models. For HBGs, mode changes imply dynamic changes in the causal structure, and this alters the computational model during execution. We have developed a method for efficient simulation of HBG models by converting them to block diagram models, extending the procedure for BGs. Run-time changes are handled by reconfiguring the data flow paths within the blocks of the model. We demonstrate the technique by creating a computational model of an electrical power system in Matlab Simulink.

Our goal is to build efficient simulation models from HBG representations. The block diagram (BD) formalism is a widely used graphical, computational scheme for describing simulation models of continuous and hybrid systems (e.g., Ptolemy, Modelica, and Simulink, among others). We adopt the BD modeling paradigm, and develop a methodology for transforming HBGs to BDs [26][27].

We encounter two primary challenges in generating simulation models from HBG representations.

Challenge 1: Avoid pre-enumeration of model configurations. Consider a HBG model with m components and assume that each component has n_i switching junctions, where $i = 1, 2, \ldots m$. The HBG model, then, defines

$$\sum_{i=1}^{m} 2^{n_i}$$

n_i different system modes (or model configurations). When that number is large, it is infeasible to pre-enumerate all the model configurations before running the simulation. Therefore, mode changes, and reconfiguration of the BD model, have to be performed during run-time. Mode changes, implemented as junction switching, produce changes in the HBG model topology, which implies that the connections between blocks in the BD model may change dynamically during the simulations.

Challenge 2: Avoid algebraic loops. In component-based modeling, the underlying mathematical model is usually a set of differential-algebraic equations (DAEs). The DAE models may include algebraic loops. A system of equations with algebraic loops has a fixed point solution if the conditions for a unique solution are satisfied. However, generating the solution may become computationally expensive if the fixed point method needs many iterations to converge to a solution when algebraic loops are present. The order of equation evaluation, then, becomes very important.

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. This procedure is described in Section 2.
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, our uses physical system modeling semantics as defined by BGs and HBGs to impose semantic structure on hybrid computational models in Simulink. Other elegant computational approaches, such as Ptolemy and HyVisual possess these semantics in a mathematical framework, but 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.

2.1.2.b Concurrency formalisms

Code Generation Frameworks

Embedded software requires concurrency formalisms other than threads and mutexes used in traditional programming languages like C. Actor-oriented design presents a high level abstraction for composing concurrent components. However, high level abstraction often introduces overhead and results in slower system. In [30], we address the problem of generating efficient implementation for the systems with such a high level description. We use partial evaluation as an optimized compilation technique for actor-oriented models. We use a helper-based mechanism, which results in flexible and extensible code generation framework. The end result is that the benefit offered by high level abstraction comes with (almost) no performance penalty. The code generation framework has been released in open source form as part of Ptolemy II.

Feedback in Strictly Causal Systems

In [46], we ask whether strictly causal components form well defined systems when arranged in feedback configurations. The standard interpretation for such configurations induces a fixed-point constraint on the function modeling the component involved. We bring together ideas from the areas of set theory, order theory, and theory of generalized ultrametric spaces to establish the existence of a unique fixed point for any such function constructively, what has resisted more traditional approaches and has been a much coveted albeit elusive goal.

2.1.2.c Application to system level design

System-level design (SLD) is considered by many as the next frontier in electronic design automation (EDA). SLD means many things to different people since there is no wide agreement on a definition of the term. Academia, designers, and EDA experts have taken different avenues to attack the problem, for the most part springing from the basis of traditional EDA and trying to raise the level of abstraction at which integrated circuit designs are captured, analyzed, and synthesized from. However, my opinion is that this is just the tip of the iceberg of a much bigger problem that is common to all system industry. In particular, I believe that notwithstanding the obvious differences in the vertical industrial segments (for example, consumer, automotive, computing, and communication); there is a common underlying basis that can be explored. This basis may yield a novel EDA industry and even a novel engineering field that could bring substantial productivity gains not only to the semiconductor industry but to all system industries including industrial and automotive, communication and computing, avionics and building automation, space and agriculture, and health and security, in short, a real technical renaissance (more information provided in [41]).

An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures

One of the many challenges facing electronic system architects is how to provide a cost estimate related to design decisions over the entire life-cycle and product line of the architecture. Various cost modeling techniques may be used to perform this estimation. However, the estimation is often done in an ad-hoc manner, based on specific design scenarios or business assumptions. This situation may yield an unfair comparison of architectural alternatives due to the limited scope of the evaluation. A preferred estimation method would involve rigorous cost modeling based on architectural design cost drivers similar to those used in the manufacturing (e.g. process-based technical cost modeling) or in the enterprise software domain (e.g. COCOMO). This paper describes an initial study of a cost model associated with automotive electronic system architecture. The model's intended use is to evaluate system cost drivers in response to various architectural decisions (e.g. choosing a communication bus topology or mapping a function to hardware). The primary cost driver categories explored are design and development, part fabrication, assembly and in-service costs. The preliminary version of this cost model focuses on describing the key influences on cost, but not the entire mathematical model. The paper presents the cost model with the help of influence diagrams and illustrates the use of the cost modeling methodology through an automotive case study – a steer-by-wire system. As future work, we propose to build a cost model and supporting methodology that accounts for architecture evolution to address the issue of evolving architecture requirements as well as when and where to employ new technology in the architecture (see [32]).

Synthesis of task and message activation models in real-time distributed automotive systems

Modern automotive architectures support the execution of distributed safety- and time-critical functions on a complex networked system with several buses and tens of ECUs. Schedulability theory allows the analysis of the worst case end-to-end latencies and the evaluation of the possible architecture configurations options with respect to timing constraints. We present an optimization framework, based on an ILP formulation of the problem, to select the communication and synchronization model that exploits the trade-offs between the purely periodic and the precedence constrained data-driven activation models to meet the latency and

jitter requirements of the application. We demonstrated its effectiveness by optimizing complex real-life General Motors architecture (see [36]).

Optimizing end-to-end latencies by adaptation of the activation events in distributed automotive systems

Schedulability theory provides support for the analysis of the worst case latencies in distributed computations when the architecture of the system is known and the communication and synchronization mechanisms have been defined. In the design of complex automotive systems, however, a great benefit of schedulability analysis may come from its use as an aid in the exploration of the software architecture configurations that can best support the target application. We present an optimization algorithm that leverages the trade-offs between the purely periodic and the data-driven activation models to meet the latency requirements of distributed vehicle functions. We demonstrate its effectiveness on a complex automotive architecture (see [37]).

2.1.3. Model-based Design

Model-based design of embedded systems is predicated on the notion that models play an essential role in the entire life-cycle of systems from specification, design, development, verification, integration to upgrade and maintenance. Our approach refines the overall concept of "model-driven development" (advocated for example by the Model-Driven Architecture (MDA) of the Object Management Group, www.omg.org/mda) by emphasizing two novel elements: (1) the use of domain-specific modeling languages, and (2) the integration of formal analysis tools, verification techniques and model transformations in the development process.

Domain-specific modeling languages (DSMLs). Our approach to the development of embedded systems is centered on analyzable models that capture the developer's design intent. Domain specificity means that modeling languages are tailored to the needs of the application domain. The success of this approach has been demonstrated by existing tools like Simulink/Stateflow and Matrix-X, which are widely used in the aerospace/automotive industry for flight control/vetronics software development. In spite of their successes, these tools exhibit two serious shortcomings: (1) It is impossible to reason formally about and verify the code generated by the tools since the semantics of their modeling languages are not precisely defined, and (2) It is very hard to integrate the generated code into a larger system which contains parts that use other models of computation.

We have continued our research on a technology that supports the precise definition of the abstract syntax and well-formedness rules for DSMLs through the use of meta-modeling and meta-programmable tools [11]. We have also extended our model transformation tool suite and deepened its theoretical foundation.

2.1.3.a Composition of Domain Specific Modeling Languages

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. We focused our efforts on the semantic foundation of modeling languages and their extension to sensor network application domain.

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

1. We have developed new theory for defining the structural semantics of domain specific modeling language and developed tools for composing and comparing metamodels [12][13][14][15]. Despite the fundamental role of structure in the model-based approach, it remained largely unformalized. To address this and other issues, we developed a mathematical formulation of structure, giving a precise tool-independent definition that is amendable to formal analysis. This work also provides a formal understanding of model transformations and metamodeling, yielding a complete picture of the structural basis of model-based design. In this sense, we broaden the term structural semantics to include the semantics of model transformation and metamodeling.

2. We have developed the second release of a semantic anchoring tool suite [19][25] 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. Our attention turned to the formulation of semantic units (as opposed to the technique for their formal specification). We continued experimenting with building alternative derivations for "systems of semantic units" that logically bring together behavioral and interaction categories. This work closely related to Berkeley's work on abstract semantics but approaches the problem from a very different angle.

3. We have continued developing the foundations for the compositional specification of behavioral semantics in the semantic anchoring framework [17][18][21]. In the semantic anchoring infrastructure, we define a finite set of semantic units, which capture the semantics of basic behavioral and interaction categories. If the semantics of a DSML can be directly anchored to one of these basic categories, its semantics can be defined by simply specifying the model transformation rules between the metamodel of the DSML and the Abstract Data Model of the semantic unit [18]. However, in heterogeneous systems, the semantics is not always fully captured by a predefined semantic unit. If the semantics is specified from scratch (which is the typical solution if it is done at all) it is not only expensive but we loose the advantages of anchoring the semantics to (a set of) common and well-established semantic units. This is not only loosing reusability of previous efforts, but has negative consequences on our ability to relate semantics of DSMLs to each other and to guide language designers to use well understood and safe behavioral and interaction semantic "building blocks" as well.
   Our proposed solution is to define semantics for heterogeneous DSMLs is the composition of semantic units. If the composed semantics specifies a behavior which is frequently used in system design (for example, the composition of SDF interaction semantics with FSM behavioral semantics defines semantics for modeling signal processing systems), the resulting semantics can be considered a *derived semantic unit*, which is built on *primary semantic units*, and could be offered up as one of the set of semantic units for future anchoring efforts. Note that *primary semantic units* refer to the semantic units that capture the semantics of the *basic behavioral categories*, such as FSM, TA and HA. The composition approach we describe in the rest of the paper is

strongly influenced by Gossler and Sifakis framework for composition by clearly separating behavior and interaction.

4. We continued our efforts in migrating research results into stable tool suites and presenting their use for the broader community [11][23][24]. Some of our complex applications turned our attention to the challenges of managing large models. This requires the development of a variety of techniques for the decomposition and recomposition of large model databases. The unique challenge is the reconstruction of model consistency after model recomposition.
5. We have extended model-based design approaches to developing sensor network applications on SOA platforms [8][10][22].

2.1.3.b Model Transformations

In the model-based design area, we have further developed and refined our model transformation tool suite. Based on practical feedback from users who have used the model transformation language GReAT, we have developed a suite of new capabilities, listed below. The updated language and tools are available from the ESCHER website (http://escher.isis.vanderbilt.edu).

- Templatized transformation rules. This is a construct, similar to the 'template functions' of C++ or 'higher-order functions' of ML and Haskell, where the type of specific rule elements is not specified at rule definition time, rather when the rule is used in a context. 'Template rules' are almost identical to normal rules, except the type of pattern variables (i.e. classes) is not bound. When a 'template rule' is used in a larger model transformation, then the designer has the opportunity to 'bind' the type of the pattern variable to a concrete type. This improves readability and the reusability of transformation rules.
- 'Blockification' tool. When constructing complex transformations, we have observed that it is a very common operation when the user groups together a certain sequence of rewriting rules into a block which is then re-used on a higher level. When done manually, this involves a lot of tedious editing operations that are error-prone. We have developed an interactive tool that allows selecting a group of rules and forms a higher-level block from them that is inserted into their place. This tool is a 'modeler's aid' for constructing complex transformations. The tool checks several consistency properties and executes the 'blockification' only if the result does not violate the semantics of the original transformation program.
- 'Next rule connector' tool. Similarly to the tool explained above, this tool supports the rapid, sequential 'wiring' of two transformation rules. The designer has to select an 'upstream' and 'downstream' rule, and the tool connects them up using the ordering of the ports of the rules. The tool performs a semantic check on the compatibility between the rules, and establishes the connections only if they are legal.
- 'Group' operator. One common operation in transforming designs (using graph transformation techniques) is the deletion, copying, or moving entire subgraphs. While this could be done with elementary operations of node/edge addition and removal, their modeling is rather cumbersome. To support such operations in a convenient manner, we have introduced a new type of rewriting rule into the transformation language: the 'group' rule. The group rule matches the host graph and generates (potentially multiple) matches for the pattern nodes and edges. Now, the designer can select specific pattern

nodes and edges to form 'groups' (i.e. sub graphs) from the matched elements, which are treated as a unit, a 'cluster'. These clusters can then be deleted/copied/moved as a unit in the action part of the rule. The detailed design of this operator has been published in [1]. We have found a number of application examples for the new operator; these have been published in [2]. Both the GREAT interpreter and the code generator support now this new operator.

We have used the GReAT tool suite 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 uses a transformational approach to 'platform modeling'. In another, NASA-sponsored project we have used the language to develop a Simulink and Stateflow code generator that compiles (restricted) models into C code and another, intermediate language that can then be subjected to static code analysis to prove safety-critical properties of the code (e.g. no buffer overflows). During the year, we have continued making improvements to the GReAT and UDM packages.

Another related effort focused on platform modeling and the connection between modeling languages and analysis tools.

In the platform modeling area, we developed 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. The results of this work have been reported in [4] and [6] , the latter being a PhD thesis.

One of the most difficult problems in model transformations is to ensure their correctness. For complex, model-based design tool chains, it is essential that the model transformations connecting the different tools preserve semantics of the models they translate. We have started working on this problem and one approach, based on verifying the correctness of specific instances of transformations has been developed for specific set of examples. Early results have been published in [5] and [7]. These techniques can be considered as light-weight formal methods for producing an 'independent certificate' for each execution of the transformation that proves that the models transformed preserve some property of interest (e.g. reachability in finite transition systems).

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

Decentralized control of unmanned underwater vehicles
A decentralized control scheme for large packs of unmanned underwater vehicles (UUV) using reachability computation and model predictive control is proposed and investigated. This scheme fits within a broader template based method for organizing and controlling large groups of UUVs and addressed one important mode of operation for these vehicles, specifically decentralized, coordinated group pursuit and tracking of intruders through the UUV network. The reachability computation and model predictive control schemes are presented along with simulation results with medium-sized packs of UUVs. Hardware implementation is also described for five UUVs (see [31]).

2.1.4.b. Embedded Software for National and Homeland Security

Autonomous Ground Vehicles
Ground vehicle research, although more stable than airborne vehicles, presents the subtle problem of providing intelligent behavior which operates in real-time, executes safely, and yet provides a “smooth” reaction to stimuli—i.e., the software behavior is somewhat humanized. Our application is the DARPA Urban Challenge, and we are using the ground vehicle testbed to show the performance of various algorithms and advancements in theory of switched systems, model-predictive control, parameter identification, time-triggered distributed components, and computer vision, as well as how these components work in real-time with one another. Additional concerns which this project addresses are distributed software testing, component-based design, and vehicle/sensor health monitoring. We are proving many of the theories and algorithms which have been developed in the last four years of the ITR.

Real-time Computer Vision
The goal of this task is to detect moving objects using a stereo camera system mounted on a car. This information will be used to detect, and estimate the trajectories of (possibly) mobile obstacles such as other vehicles. Note that, in this scenario, we wish to detect both objects in close proximity to our vehicle (for example, the vehicle in front of us), and also objects that are farther away (for example, oncoming vehicles), so we cannot make assumptions about whether height errors at adjacent pixels appear to be close to planar. Our solution must also be able to run in real time. In this application, we are interested in determining 3D motion in the scene, rather than the more-studied case of 2D motion in the image. In particular, we want to find the motion of objects in the scene relative to each other; if we have vehicle state data available, or if we can assume that the majority of the scene is not moving, we can determine which motion-segmented

region corresponds to the background and remove its perceived motion from the perceived motion of the other objects, obtaining the motion of the other objects in a global frame.

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 continued to explore 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 (see [77][86]).

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 4 below, including publications representing findings that were reported in the previous annual report.

[1] D. Balasubramanian, A. Narayanan, S. Neema, F. Shi, R. Thibodeaux, G. Karsai: A Subgraph Operator for Graph Transformation Languages, Proceedings of 6th International Workshop on Graph Transformation and Visual Modeling Techniques, Braga, Portugal.
In practical applications of graph transformation techniques to model transformations one often has the need for copying, deleting, or moving entire subgraphs that match a certain graph pattern. While this can be done using elementary node and edge operations, the transformation is rather cumbersome to write. To simplify the transformation, we have recently developed a novel approach that allows selecting subgraphs from the matched portion of the host graph, applying a filter condition to the selection, and performing a delete, move, or copy operation on the filtered result in the context of a transformation rule. The approach has been implemented in the GReAT language and tested on examples that show the practical efficacy of the technique. The paper describes the technique in detail and illustrates its use on a real-life example.

[2] D. Balasubramanian, A. Narayanan, S. Neema, B. Ness, F. Shi, R. Thibodeaux, and G. Karsai: “Applying a Grouping Operator in Model Transformations”, submitted to 3rd International Workshop on Graph and Model Transformation (GraMoT). (Submitted.)
The usability of model transformation languages depends on the level of abstractions one can work with in rules to perform complex operations on models. Recently, we have

introduced a novel operator for our model transformation language GReAT that allows the concise specification of complex model (graph) rewriting operations. In this paper we show how the new operator can be used to implement non-trivial model manipulations with fewer and simpler rules, while maintaining efficiency. The examples were motivated by problems encountered in real-life model transformations.

[3] A. Agrawal, Gabor Karsai, Sandeep Neema, Feng Shi, Attila Vizhanyo: "The Design of a Language for Model Transformations", Journal on Software and System Modeling, pp 261-288, Volume 5, Number 3 / September, 2006.
Model-driven development of software systems envisions transformations applied in various stages of the development process. Similarly, the use of domain-specific languages also necessitates transformations that map domain-specific constructs into the constructs of an underlying programming language. Thus, in these cases, the writing of transformation tools becomes a first-class activity of the software engineer. This paper introduces a language that was designed to support implementing highly efficient transformation programs that perform model-to-model or model-to-code translations. The language uses the concepts of graph transformations and metamodeling, and is supported by a suite of tools that allow the rapid prototyping and realization of transformation tools.

[4] T. Szemethy, Gabor Karsai: "PML: a Language for Platform Modeling," Electronic Communications of the EASST, Volume 4, 2006: Graph and Model Transformation 2006.
Modeling the computational platforms is necessary to analyze the execution characteristics of systems developed using a model-based approach. In this paper, we introduce a novel platform modeling language: PML that is based on (a) transformational concepts borrowed from graph transformation languages, and (b) generative concepts from platform modeling, like 'kernel skeleton'. PML relies on higher-level, compact constructs that represent a special case of model transformations, and which are then used to specify platform semantics. The paper also illustrates how PML constructs can be compiled into lower-level constructs of more traditional model transformational languages, such as GReAT.

[5] A. Narayanan, Gabor Karsai: "Using Semantic Anchoring to Verify Behavior Preservation in Graph Transformations," Electronic Communications of the EASST, Volume 4, 2006: Graph and Model Transformation 2006.
Graph transformation is often used to transform domain models from one domain specific language (DSML) to another. In some cases, the DSMLs are based on a formalism that has many implementation variants, such as Statecharts. For instance, it could be necessary to transform iLogix Statechart models into Matlab Stateflow models. The preservation of behavior of the models is crucial in such transformations. Bisimulation has previously been demonstrated as an approach to verifying behavior preservation, and semantic anchoring is an approach to specifying the dynamic semantics of DSMLs. We propose a method to verify behavior preservation, using bisimulation in conjunction with semantic anchoring. We will consider two hypothetical variants of the Statecharts formalism, and specify the operational semantics of each variant by semantic anchoring, using Abstract State Machines as a common semantic framework. We then establish

bisimulation properties to verify if the behavior models of the source and target Statechart models are equivalent for a particular execution of the transformation.

[6] T. Szemethy: Domain-Specific Models, Model Analysis, Model Transformations. PhD thesis, Vanderbilt University, 2006.
This dissertation proposes a novel approach, applicable in the design-time analysis and verification of computer-based systems. The proposed approach, platform modeling, constructs analysis models capturing the system's behavior on a particular implementation platform. The approach is discussed in the context of Model-Integrated Computing, which is a development methodology leveraging on the use of domain-specific modeling languages and advanced model transformation techniques. During development, platform-independent design models are refined into platform-specific models using model transformation. The main contribution of this work is the enhancement of this process with the automatic generation of analysis models. These analysis models assign platform-specific semantics to the design model. This assignment is done through a transformational approach, using graph transformation. The dissertation presents a case study using a conventional graph transformation tool. Then, a new graph transformation language designed specifically for such transformations is proposed, and its advantages are demonstrated.

[7] G. Karsai, A. Narayanan: On the Correctness of Model Transformations in the Development of Embedded Systems, Proceedings of the 2006 Monterey Workshop (submitted), Paris, France, Oct. 2006.
Model based techniques have become very popular in the development of software for embedded systems, with a variety of tools for design, simulation and analysis of model based systems being available (such as Matlab's Simulink, the model checking tool NuSMV, etc.). Model transformations usually play a critical role in such model based development approaches. While the available tools are geared to verify properties about individual models, the correctness of model transformations is generally not verified. However, errors in the transformation could present serious problems. Proving a property for a certain source model becomes irrelevant if an erroneous transformation produces an incorrect target model. One way to provide assurance about a transformation would be to prove that it preserves certain properties of the source model (such as reachability) in the target model. In this paper, we present some general approaches to providing such assurances about model transformations. We will present some case studies where these techniques can be applied.

[8] Isaac Amundson, Manish Kushwaha, Xenofon Koutsoukos, Sandeep Neema, and Janos Sztipanovits. "OASiS: A Service-Oriented Middleware for Pervasive Ambient-Aware Sensor Networks", *Pervasive and Mobile Computing Journal on Middleware for Pervasive Computing.* (Submitted October 2006.)
Heterogeneous sensor networks consisting of networked devices embedded into the physical world have a significant role in pervasive computing systems. Such sensor networks may contain wireless sensor networks that are ensembles of small, smart, and cheap sensing and computing devices that permeate the environment, as well as high-bandwidth rich sensors such as satellite imaging systems, meteorological stations, air

quality stations, and security cameras. Emergency response, homeland security, and many other applications have a very real need to interconnect such diverse networks and access information in real-time. While Web service standards provide well-developed mechanisms for resource-intensive computing nodes, linking such mechanisms with wireless sensor networks is very challenging because of limited resources, volatile communication links, and often node mobility. This paper presents a service-oriented programming model and middleware for ad-hoc wireless sensor networks which permits discovery and access of Web services. Sensor network applications are realized as graphs of modular and autonomous services with well-defined interfaces that allow them to be published, discovered, and invoked over the network, providing a convenient mechanism for integrating services from heterogeneous sensor systems. Our approach provides dynamic discovery, composition, and binding of services based on an efficient localized constraint satisfaction algorithm that can be used for developing ambient-aware applications that adapt to changes in the environment. A tracking application that employs many inexpensive sensor nodes, as well as a Web service, is used to illustrate the approach. Our results demonstrate the feasibility of ambient-aware applications that interconnect wireless sensor networks and Web services.

[9] Matthew Emerson and Sandeep Neema and Janos Sztipanovits: "Metamodeling Languages and Metaprogrammable Tools," in *Handbook of Real-Time and Embedded Systems*, Ed. Insup Lee, Joseph Leung, Sang H. Son, CRC Press, 2006
Model-Integrated Computing, one practical manifestation of OMG's Model Driven Architecture, advocates the development of domain specific modeling languages for system design, specification, and analysis. Recent years have seen a variety of new tools and metamodeling languages which support MIC-style model-based design, including the Eclipse Modeling Framework (EMF) and the Microsoft Domain-Specific Languages (DSL) tools. This situation introduces new challenges for system designers, including the challenge of deciding which tool and language to adopt for a particular project. This paper provides two contributions to the research on model-based design. First, we provide a technical comparison of three modeling languages and their supporting tools: GME's MetaGME, EMF's Ecore, and Microsoft's developing Domain Model Designer language. Through this comparison we show that the consequences of choosing one metamodeling language over another are non-critical for language specification, since all of the metamodeling languages support the same fundamental language design concepts. Second, we generalize previous work to outline a set of tools and procedures which may be used to adapt an existing metaprogrammable tool suite to support new metamodeling languages while maintaining backwards compatibility with existing tools and models.

[10] Isaac Amundson, Manish Kushwaha, Xenofon Koutsoukos, Sandeep Neema, and Janos Sztipanovits. "Efficient Integration of Web Services in Ambient-aware Sensor Network Applications". *3rd IEEE/CreateNet International Workshop on Broadband Advanced Sensor Networks (BaseNets 2006)*, San Jose, CA, Oct. 2006.
Sensor webs are heterogeneous collections of sensor devices that collect information and interact with the environment. They consist of wireless sensor networks that are ensembles of small, smart, and cheap sensing and computing devices that permeate the environment as well as high-bandwidth rich sensors such as satellite imaging systems,

meteorological stations, air quality stations, and security cameras. Emergency response, homeland security, and many other applications have a very real need to interconnect such diverse networks and access information in real-time. While Internet protocols and Web standards provide well-developed mechanisms for accessing this information, linking such mechanisms with resource-constrained sensor networks is very challenging because of the volatility of the communication links. This paper presents a service-oriented programming model for sensor networks which permits discovery and access of Web services. Sensor network applications are realized as graphs of modular and autonomous services with well-defined interfaces that allow them to be described, published, discovered, and invoked over the network providing a convenient way for integrating services from heterogeneous sensor systems. Our approach provides dynamic discovery, composition, and binding of services based on an efficient localized constraint satisfaction algorithm that can be used for developing ambient-aware applications that adapt to changes in the environment. A tracking application that employs many inexpensive sensor nodes, as well as a Web service, is used to illustrate the approach. Our results demonstrate the feasibility of ambient-aware applications that interconnect wireless sensor networks and Web services.

[11] Karsai, G., Ledeczi, A., Neema, S., Sztipanovits, J.: The Model-Integrated Computing Toolsuite: Metaprogrammable Tools for Embedded Control System Design, *Proc. of the IEEE Joint Conference CCA, ISIC and CACSD*, Munich, Germany, 2006. Model-Integrated Computing is a development approach that advocates the use of Domain-Specific Modeling throughout the system development process and lifecycle. This paper describes and summarizes the generic and reusable soft-ware tools that support MIC and which can be tailored to solve a wide variety of modeling, analysis, and generation problems in an engineering process.

[12] Jackson, E., Sztipanovits, J.: "Towards A Formal Foundation For Domain Specific Modeling Languages," *Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'05)*, pp. 53-63, Seoul, Korea October 22-25, 2006 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 specific modeling languages (DSMLs). The success of the DSML approach has encouraged work on the heterogeneous com- position 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] Jackson, E., Sztipanovits, J.: "Constructive Techniques for Meta and Model

Level Reasoning," Models 07, (submitted)
The structural semantics of UML-based metamodeling were recently explored [1], providing a characterization of the models adhering to a metamodel. In particular, metamodels can be converted to a set of constraints expressed in a decidable subset of first-order logic, an extended Horn logic. We augment the constructive techniques found in logic programming, which are also based on an extended Horn logic, to produce constructive techniques for reasoning about models and metamodels. These methods have a number of practical applications: At the meta-level, it can be decided if a (composite) metamodel characterizes a non-empty set of models, and a member can be automatically constructed. At the model-level, it can be decided if a submodel has an embeddeding in a well-formed model, and the larger model can be constructed. This amounts to automatic model construction from an incomplete model. We describe the concrete algorithms for constructively solving these problems, and provide concrete examples.

[14] Jackson, E., Sztipanovits, J.: "Models as Structures: The Structural Semantics of Model-Based Design," Journal of Software and Systems Modeling (SOSYM), (submitted)
Model-based approaches to system design are now widespread and successful. These approaches make extensive use of model structure to facilitate domain specific abstractions and platform modeling. However, the structural semantics of model-based approaches are not well-understood. In this paper we develop the formal foundations for the structural semantics of model-based design. Additionally, we show how our formalization can be applied to existing tools, and how it yields algorithms for the analysis of domain-specific modeling languages (DSMLs) and model transformations.

[15] Emerson M., Sztipanovits J.: "Techniques for Metamodel Composition", *OOPSLA – 6th Workshop on Domain Specific Modeling*, 123-139, Portland, Oregon, October 22, 2006
The process of specifying an embedded system involves capturing complex interrelationships between the hardware domain, the software domain, and the engineering domain used to describe the environment in which the system will be embedded. Developers increasingly turn to domain-specific modeling techniques to manage this complexity, through such approaches as Model Integrated Computing and Model Driven Architecture. However, the specification of domain-specific modeling language syntax and semantics remains more of an art than a science. Typically, the syntax of a DSML is captured using a metamodel; however, there are few best-practices for metamodeling and no public collection of reusable metamodel to address common language specification requirements. There is a need for an advanced, comprehensive language design environment that offers tool support for a wide range of metamodel reuse strategies and the preservation of metamodeling best-practices. We outline existing techniques for the reuse and composition of metamodels, and propose a new metamodel composition technique we call Template Instantiation.

[16] Emerson M., Duncavage S., Mathe J., Sztipanovits J.: "WiNeSim: A Wireless Network Simulation Tool", *EMSOFT – 1rst International Workshop on Embedded*

Systems Security, Seoul, South Korea, October 26, 2006
We provide an overview of WiNeSim, a highly extensible wireless network modeling and simulation tool with a network attack modeling component. WiNeSim provides a high-level graphical modeling interface for the rapid declarative specification of network configurations, including the selection of node hardware, MAC protocols, and routing protocols. Furthermore, it enables users to specify certain network nodes to act as "smart" attackers who adapt their behavior and attack styles based on perceived network conditions in accordance with user-specified algorithms given as timed automata. WiNeSim is designed to easily integrate new network protocols and attack styles.

[17] Sztipanovits, J.: "Towards the Compositional Specification of Semantics for Heterogeneous Domain-Specific Modeling Languages," *Workshop on Foundations and Applications of Component-based Design*. Seoul, Korea October 26, 2006
Domain-Specific Modeling Languages (DSMLs) play fundamental role in the model-based design of embedded software and systems. While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of DSML semantics is still a hard problem. In previous work, we have developed methods and tools for the semantic anchoring of DSML-s. Semantic anchoring introduces a set of reusable "semantic units" that provides 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.

[18] Kai Chen, Janos Sztipanovits, Sandeep Neema: "Compositional Specification of Behavioral Semantics," *Technical report, ISIS-06-705,* 2006.
Domain-Specific Modeling Languages (DSMLs) play fundamental role in the model-based design of embedded software and systems. While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of DSML semantics is still a hard problem. 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.

[19] Kai Chen, Janos Sztipanovits, Sandeep Neema: "A Case Study on Semantic Unit Composition," Workshop on Modeling in Software Engineering, ICSE 2007 (MISE 2007).
In previous work we have discussed a semantic anchoring framework that enables the semantic specification of Domain-Specific Modeling Languages by specifying semantic anchoring rules to predefined semantic units. This framework is further extended to support heterogeneous systems by developing a method for the composition of semantic

units. In this paper, we explain the semantic unit composition through a case study.

[20] Sztipanovits, J., Bay, J., Rohrbough,, L., Sastry, S., Schmidt, D., Whitaker, N., Winter, D.: "ESCHER: A New Technology Transitioning Model in Embedded Systems and Software," *IEEE Computer*, March, 2007
The paper describes a new transitioning model, ESCHER, that focuses on establishing a quality controlled repository supported by industry.

[21] Kai Chen, Janos Sztipanovits, and Sandeep Neema: "Compositional Specification of Behavioral Semantics," *Proceedings of Design Automation and Test in Europe Conference (DATE 07)*, Nice, France, April 2-5, 2007
An emerging common trend in model-based design of embedded software and systems is the adoption of Domain-Specific Modeling Languages (DSMLs). While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of DSML semantics is still a hard problem. 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 (ASM) framework. In this paper, we extend the semantic anchoring framework to heterogeneous behaviors by developing a 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.

[22] Manish Kushwaha, Isaac Amundson, Xenofon Koutsoukos, Sandeep Neema, and Janos Sztipanovits. "OASiS: A Programming Framework for Service-Oriented Sensor Networks". *In IEEE/Create-Net COMSWARE 2007*, Bangalore, India, Jan. 2007
Wireless sensor networks consist of small, inexpensive devices which interact with the environment, communicate with each other, and perform distributed computations in order to monitor spatio-temporal phenomena. These devices are ideally suited for a variety of applications including object tracking, environmental monitoring, and homeland security. At present, sensor network technologies do not provide off-the-shelf solutions to users who lack low-level network programming experience. Because of limited resources, ad hoc deployments, and volatile wireless communication links, the development of distributed applications require the combination of both application and system-level logic. Programming frameworks and middleware for traditional distributed computing are not suitable for many of these problems due to the resource constraints and interactions with the physical world. To address these challenges we have developed OASiS1, a programming framework which provides abstractions for object-centric, ambient-aware, service-oriented sensor network applications. OASiS uses a well-defined model of computation based on globally asynchronous locally synchronous dataflow, and is complemented by a user-friendly modeling environment. Applications are realized as graphs of modular services and executed in response to the detection of physical phenomena. We have also implemented a suite of middleware services that support OASiS to provide a layer of abstraction shielding the low-level system complexities. A tracking application is used to illustrate the features of OASiS. Our results demonstrate the feasibility and the benefits of a service-oriented programming framework for

composing and deploying applications in resource constrained sensor networks.

[23] Fabrice Kordon and Janos Sztipanovits (ed.): "Networked Systems: Realization of Reliable Systems on Unreliable Networked Platforms," *Proceedings of the Sixth Monterey Workshop*, LNCS Vol. 4322, 2007
Networked computing is increasingly becoming the universal integrator for large-scale systems. In addition, new generation of wireless networked embedded systems rapidly create new technological environments that imply complex interdependencies amongst all layers of societal-scale critical infrastructure, such as transportation, energy distribution and telecommunication. This trend makes reliability and safety of networked computing a crucial issue and a technical precondition for building software intensive systems that are robust, fault tolerant, and highly available. The 12th Monterey Workshop on "Networked Systems: Realization of Reliable Systems on Unreliable Networked Platforms" focused on new, promising directions in achieving high software and system reliability in networked systems.

[24] Sztipanovits, J.: "Model-based Software Development," ESMD-SW Workshop, NASA, Houston, TX March 15-17, 2007
Model based software and system design is based on the end-to-end use of formal, composable and manipulable models in the product life-cycle. An emerging common thread is that modeling languages are domain-specific: they offer software developers concepts and notations that are tailored to capture essential characteristics of their application domain. Model Integrated Computing (MIC) developed at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University is part of this new direction. The presentation provides an overview of key principles, methods and tools of model-based software and systems design and discusses application directions and experience in high-confidence systems design, architecture exploration and model-based systems integration.

[25] Sztipanovits, J.: "Defining Behavioral Semantics for Domain Specific Modeling Languages," NSF Workshop on Semantics for Event-Based Modeling, RTAS, Bellevue, WA, April 3, 2007.
Domain-Specific Modeling Languages (DSMLs) play fundamental role in the model-based design of embedded software and systems. While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of DSML semantics is still a hard problem. In previous work, we have developed methods and tools for the semantic anchoring of DSML-s. Semantic anchoring introduces a set of reusable "semantic units" that provides reference semantics for basic behavioral categories using the Abstract State Machine framework. In this presentation, 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.

[26] I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P. J. Mosterman, "A Method for Efficient Simulation of Hybrid Bond Graphs," *International Conference*

on Bond Graph Modeling and Simulation (ICBGM 2007), pp. 177-184, Jan 2007.
The hybrid bond graph (HBG) paradigm is a uniform, multi-domain physics-based modeling language. It incorporates controlled and autonomous mode changes as idealized switching functions that enable the reconfiguration of energy flow paths to model hybrid physical systems. Building accurate and computationally efficient simulation mechanisms from HBG models is a challenging task, especially when there is no a priori knowledge of the subset of system modes that will be active during the simulation. In this work, we present an approach that exploits the inherent causal structure in HBG models to derive efficient hybrid simulation models as reconfigurable block diagram structures. We present a MATLAB/Simulink implementation of our approach and demonstrate its effectiveness using an electrical circuit example.

[27] M. Daigle, I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Efficient Simulation of Component-Based Hybrid Models Represented as Hybrid Bond Graphs," *Hybrid Systems: Computation and Control (HSCC 2007)*, Lecture Notes in Computer Science, vol. 4416, pp. 680-683, Apr 2007.
The complexity of modern embedded systems often requires the use of simulations for systematic design, analysis, and verification tasks. The nonlinear and hybrid nature of these systems make the building of accurate and computationally efficient simulation models very challenging. In this work, we adopt the Hybrid Bond Graph (HBG) paradigm, a uniform, multi-domain physics-based modeling language with local switching functions that enable the reconfiguration of energy flow paths to model hybrid systems. The inherent causal structure in HBG models is exploited to derive efficient hybrid simulation models as reconfigurable block diagram structures. We demonstrate our approach by modeling and analyzing the behavior of an electrical power system.

[28] Antoon Goderisa, Christopher Brooks, lkay Altintas, Edward A. Lee, "Composing Different Models of Computation in Ptolemy II and Kepler," 2007 Proceedings, International Conference on Computational Science (ICCS), May, 2007; To appear at International Conference on Computational Science (ICCS) 2007.

A model of computation (MoC) is a formal abstraction of execution in a computer. There is a need for composing MoCs in e-science. Kepler, which is based on Ptolemy II, is a scientific workflow environment that allows for MoC composition. This paper explains how MoCs are combined in Kepler and Ptolemy II and analyzes which combinations of MoCs are currently possible and useful. It demonstrates the approach by combining MoCs involving dataflow and finite state machines. The resulting classification should be relevant to other workflow environments wishing to combine multiple MoCs. Keywords: Model of computation, scientific workflow, Kepler, Ptolemy II.

[29] Takashi Nagata, Masayoshi Tomizuka, "Engine Torque Control Based on Discrete Event Model and Disturbance Observer," ASME International Mechanical Engineering Congress and Exposition (IMECE2007), (submitted), May, 2007; Draft submitted in May 2007, to be presented in Nov. 2007.

This paper presents a novel control method for torque generation in spark ignition (SI) engine. A model-based approach is employed to control engine torque output by adjusting throttle air intake with considerations of robust stability and performance. Discrete event engine model (DEM) is adopted with an addition of torque generation dynamics. Disturbance observer (DOB) techniques are utilized to achieve robust stability and performance by regarding the discrepancy between the actual plant and the nominal plant with desired plant characteristics as an equivalent disturbance input, which is estimated and cancelled. The desired plant behavior is stably realized by the DOB up to a bandwidth sufficient for a torque control application covered in our previous works. Numerical results show the effectiveness of the proposed scheme.

[30] Gang Zhou, Man-Kit Leung, and Edward A. Lee, "A Code Generation Framework for Actor-Oriented Models with Partial Evaluation," Proceedings of International Conference on Embedded Software and Systems 2007, LNCS 4523, Y.-H. Lee et al., 786-799, May, 2007.

Embedded software requires concurrency formalisms other than threads and mutexes used in traditional programming languages like C. Actor-oriented design presents a high level abstraction for composing concurrent components. However, high level abstraction often introduces overhead and results in slower system. We address the problem of generating efficient implementation for the systems with such a high level description. We use partial evaluation as an optimized compilation technique for actor-oriented models. We use a helper-based mechanism, which results in flexible and extensible code generation framework. The end result is that the benefit offered by high level abstraction comes with (almost) no performance penalty. The code generation framework has been released in open source form as part of Ptolemy II 6.0.1.

[31] Jongho Lee, Mikael Eklund, Shankar Sastry, Unpublished article, "Group pursuit policies for UUVs using nonlinear model predictive control and reachable sets," May, 2007.

A decentralized control scheme for large packs of unmanned underwater vehicles (UUV) using reachability computation and model predictive control is proposed and investigated. This scheme fits within a broader template based method for organizing and controlling large groups of UUVs and addressed one important mode of operation for these vehicles, specifically decentralized, coordinated group pursuit and tracking of intruders through the UUV network. The reachability computation and model predictive control schemes are presented along with simulation results with medium-sized packs of UUVs. Hardware implementation is also described for five UUVs

[32] Arkadeb Ghosal, Sri Kanajan, Randall Urbance, Alberto Sangiovanni-Vincentelli, "An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures," SAE, April, 2007.

One of the many challenges facing electronic system architects is how to provide a cost estimate related to design decisions over the entire life-cycle and product line of the

architecture. Various cost modeling techniques may be used to perform this estimation. However, the estimation is often done in an ad-hoc manner, based on specific design scenarios or business assumptions. This situation may yield an unfair comparison of architectural alternatives due to the limited scope of the evaluation. A preferred estimation method would involve rigorous cost modeling based on architectural design cost drivers similar to those used in the manufacturing (e.g. process-based technical cost modeling) or in the enterprise software domain (e.g. COCOMO). This paper describes an initial study of a cost model associated with automotive electronic system architecture. The model's intended use is to evaluate system cost drivers in response to various architectural decisions (e.g. choosing a communication bus topology or mapping a function to hardware). The primary cost driver categories explored are design and development, part fabrication, assembly and in-service costs. The preliminary version of this cost model focuses on describing the key influences on cost, but not the entire mathematical model. The paper presents the cost model with the help of influence diagrams and illustrates the use of the cost modeling methodology through an automotive case study – a steer-by-wire system. As future work, we propose to build a cost model and supporting methodology that accounts for architecture evolution to address the issue of evolving architecture requirements as well as when and where to employ new technology in the architecture.

[33] Nadathur Satish, Kaushik Ravindran and Kurt Keutzer, "A Decomposition-based Constraint Optimization Approach for Statically Scheduling Task Graphs with Communication Delays to Multiprocessors," 10th Conference of Design, Automation and Test in Europe (DATE-07), 230-235, April, 2007.

We present a decomposition strategy to speed up constraint optimization for a representative multiprocessor scheduling problem. In the manner of Benders decomposition, our technique solves relaxed versions of the problem and iteratively learns constraints to prune the solution space. Typical formulations suffer prohibitive run times even on medium-sized problems with less than 30 tasks. Our decomposition strategy enhances constraint optimization to robustly handle instances with over 100 tasks. Moreover, the extensibility of constraint formulations permits realistic application and resource constraints, which is a limitation of common heuristic methods for scheduling. The inherent extensibility, coupled with improved run times from a decomposition strategy, posit constraint optimization as a powerful tool for resource constrained scheduling and multiprocessor design space exploration.

[34] Thomas Huining Feng, Lynn Wang, Wei Zheng, Sri Kanajan, and Sanjit A. Seshia, "Automatic Model Generation for Black Box Real-Time Systems," Design, Automation and Test in Europe (DATE) Conference, April, 2007.

Embedded systems are often assembled from black box components. System-level analyses, including verification and timing analysis, typically assume the system description, such as RTL or source code, as an input. There is therefore a need to automatically generate formal models of black box components to facilitate analysis. We propose a new method to generate models of real-time embedded systems based on

machine learning from execution traces, under a given hypothesis about the system's model of computation. Our technique is based on a novel formulation of the model generation problem as learning a dependency graph that indicates partial ordering between tasks. Tests based on an industry case study demonstrate that the learning algorithm can scale up and that the deduced system model accurately reflects dependencies between tasks in the original design. These dependencies help us formally prove properties of the system and also extract data dependencies that are not explicitly stated in the specifications of black box components.

[35] Sumitra Ganesh, Aaron Ames, Ruzena Bajcsy, "Composition of Dynamical Systems for Estimation of Human Body Dynamics," Proceedings of 10th International Conference on Hybrid Systems Computation and Control 2007, Alberto Bemporad, Antonio Bicchi, Giorgio Buttazzo, 702-706, April, 2007.

(No abstract.)

[36] Wei Zheng, Marco Di Natale, Claudio Pinello, Paolo Giusto, Alberto Sangiovanni Vincentelli, "Synthesis of task and message activation models in real-time distributed automotive systems," Design, Automation and Test in Europe, April, 2007.

Modern automotive architectures support the execution of distributed safety- and time-critical functions on a complex networked system with several buses and tens of ECUs. Schedulability theory allows the analysis of the worst case end-to-end latencies and the evaluation of the possible architecture configurations options with respect to timing constraints. We present an optimization framework, based on an ILP formulation of the problem, to select the communication and synchronization model that exploits the trade-offs between the purely periodic and the precedence constrained data-driven activation models to meet the latency and jitter requirements of the application. We demonstrated its effectiveness by optimizing complex real-life GM architecture.

[37] Marco Di Natale, Wei Zheng, Claudio Pinello, Paolo Giusto, Alberto Sangiovanni-Vincentelli, "Optimizing end-to-end latencies by adaptation of the activation events in distributed automotive systems," Real-Time and Embedded Technology and Applications Symposium, April, 2007.

Schedulability theory provides support for the analysis of the worst case latencies in distributed computations when the architecture of the system is known and the communication and synchronization mechanisms have been defined. In the design of complex automotive systems, however, a great benefit of schedulability analysis may come from its use as an aid in the exploration of the software architecture configurations that can best support the target application. We present an optimization algorithm that leverages the trade-offs between the purely periodic and the data-driven activation models to meet the latency requirements of distributed vehicle functions. We demonstrate its effectiveness on a complex automotive architecture.

[38] Krishnendu Chatterjee and Thomas A. Henzinger, "Assume-guarantee Synthesis," TACAS, March, 2007.

The classical synthesis problem for reactive systems asks, given a proponent process A and an opponent process B, to refine A so that the closed-loop system A||B satisfies a given specification ϕ. The solution of this problem requires the computation of a winning strategy for proponent A in a game against opponent B. We define and study the *co-synthesis* problem, where the proponent A consists itself of two independent processes, $A=A_1||A_2$, with specifications ϕ_1 and ϕ_2, and the goal is to refine both A and A_2 so that $A_1||A_2||B$ satisfies $\phi_1 \wedge \phi_2$. For example, if the opponent B is a fair scheduler for the two processes A_1 and A_2, and ϕ_i specifies the requirements of mutual exclusion for A_i (e.g., starvation freedom), then the co-synthesis problem asks for the automatic synthesis of a mutual-exclusion protocol.

We show that co-synthesis defined classically, with the processes A_1 and A_2 either collaborating or competing, does not capture desirable solutions. Instead, the proper formulation of co-synthesis is the one where process A_1 competes with A_2 but not at the price of violating ϕ_1, and vice versa. We call this *assume-guarantee synthesis* and show that it can be solved by computing secure-equilibrium strategies. In particular, from mutual-exclusion requirements the assume-guarantee synthesis algorithm automatically computes Peterson's protocol.

[39] Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, "Generalized Parity Games," FOSSACS 07, March, 2007.

We consider games where the winning conditions are disjunctions (or dually, conjunctions) of parity conditions; we call them generalized parity games. These winning conditions, while omega-regular, arise naturally when considering fair simulation between parity automata, secure equilibria for parity conditions, and determinization of Rabin automata. We show that these games retain the computational complexity of Rabin and Streett conditions; i.e., they are NP-complete and co-NP-complete, respectively. The (co-)NP-hardness is proved for the special case of a conjunction/disjunction of two parity conditions, which is the case that arises in fair simulation and secure equilibria. However, considering these games as Rabin or Streett games is not optimal. We give an exposition of Zielonka's algorithm when specialized to this kind of games. The complexity of solving these games for k parity objectives with d parities, n states, and m edges is $O(n^{2kd} * m) * (k* d)! /(d!^k)$, as compared to $O(n^{2kd} * m) * (k* d)!$ when these games are solved as Rabin/Streett games. We also extend the subexponential algorithm for solving parity games recently introduced by Jurdzinski, Patterson, and Zwick to generalized parity games. The resulting complexity of solving generalized parity games is $n^{O(sqrt\{n\})} * (k* d)!/(d!^k)$. As a corollary we obtain an improved algorithm for Rabin and Streett games with d pairs, with time complexity $n^{O(sqrt\{n\})} * d!$.

[40] Krishnendu Chatterjee, "Optimal Strategy Synthesis for Stochastic Muller Games," FOSSACS 07, March, 2007.

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); and the quantitative problem asks for the maximal probability of winning (optimal winning) from each state. We consider omega-regular winning conditions formalized as Muller winning conditions. We present optimal memory bounds for pure almost-sure winning and optimal winning strategies in stochastic graph games with Muller winning conditions. We also present improved memory bounds for randomized almost-sure winning and optimal strategies.

[41] Alberto L. Sangiovanni-Vincentelli, *Proceedings of the IEEE*, 95(3):467-506, March 2007.

System-level design (SLD) is considered by many as the next frontier in electronic design automation (EDA). SLD means many things to different people since there is no wide agreement on a definition of the term. Academia, designers, and EDA experts have taken different avenues to attack the problem, for the most part springing from the basis of traditional EDA and trying to raise the level of abstraction at which integrated circuit designs are captured, analyzed, and synthesized from. However, my opinion is that this is just the tip of the iceberg of a much bigger problem that is common to all system industry. In particular, I believe that notwithstanding the obvious differences in the vertical industrial segments (for example, consumer, automotive, computing, and communication); there is a common underlying basis that can be explored. This basis may yield a novel EDA industry and even a novel engineering field that could bring substantial productivity gains not only to the semiconductor industry but to all system industries including industrial and automotive, communication and computing, avionics and building automation, space and agriculture, and health and security, in short, a real technical renaissance.

[42] Abhijit Davare, Douglas Densmore, Trevor Meyerowitz, Alessandro Pinto, Alberto Sangiovanni-Vincentelli, Guang Yang, Haibo Zeng, Qi Zhu, "A Next-Generation Design Framework for Platform-based Design," DVCon 2007, February, 2007.

The platform-based design methodology is based on the usage of formal modeling techniques, clearly defined abstraction levels and the separation of concerns to enable an effective design process. The METROPOLIS framework embodies the platform-based design methodology and has been applied to a number of case studies across multiple domains. Based on these experiences, we have identified three key features that need to be enhanced: heterogeneous IP import, orthogonalization of performance from behavior, and design space exploration. The next generation METRO II framework incorporates these advanced features. The main concepts underlying METRO II are described in this paper and illustrated with a small example.

[43] Joseph Gerard Makin and Alessandro Abate, Technical report, "A Neural Hybrid-System Model of the Basal Ganglia," EECS Department - University of California, at Berkeley, January, 2007.

The basal ganglia (BG) are a set of functionally related and structurally interconnected nuclei in the human brain which form part of a closed loop between cortex and thalamus, receiving input from the former and outputting to the latter. The BG have been implicated in motor control and cognitive switching tasks; in particular, it is believed that the BG function as a controller for motor tasks by selectively disinhibiting appropriate portions of the thalamus and hence activating, via a feedback loop, cortical regions. These switching behaviors are perforce discrete, whereas the underlying dynamics of neuron voltages and neurotransmitter levels are continuous-time, continuous state phenomena. To this end, we propose and simulate a hybrid automaton for modeling individual neurons that affords explicit representation of voltage discharges and discrete outputs along with continuous voltage dynamics within a single, elegant model; and which is amenable both to the construction of large networks—in particular the cortico-basalthalamic loops—and to analysis on such networks.

[44] An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures, *SAE 2007 Transactions: Journal of Passenger Cars: Electronic and Electrical Systems*, January 2007.

One of the many challenges facing electronic system architects is how to provide a cost estimate related to design decisions over the entire life-cycle and product line of the architecture. Various cost modeling techniques may be used to perform this estimation. However, the estimation is often done in an ad-hoc manner, based on specific design scenarios or business assumptions. This situation may yield an unfair comparison of architectural alternatives due to the limited scope of the evaluation. A preferred estimation method would involve rigorous cost modeling based on architectural design cost drivers similar to those used in the manufacturing (e.g. process-based technical cost modeling) or in the enterprise software domain (e.g. COCOMO). This paper describes an initial study of a cost model associated with automotive electronic system architecture. The model's intended use is to evaluate system cost drivers in response to various architectural decisions (e.g. choosing a communication bus topology or mapping a function to hardware). The primary cost driver categories explored are design and development, part fabrication, assembly and in-service costs. The preliminary version of this cost model focuses on describing the key influences on cost, but not the entire mathematical model. The paper presents the cost model with the help of influence diagrams and illustrates the use of the cost modeling methodology through an automotive case study – a steer-by-wire system. As future work, we propose to build a cost model and supporting methodology that accounts for architecture evolution to address the issue of evolving architecture requirements as well as when and where to employ new technology in the architecture.

[45] Krishnendu Chatterjee and Thomas A. Henzinger, Unpublished article, "Value Iteration," January, 2007; A Survey Paper submitted for publication in "25 Years in Model Checking".

We survey value iteration algorithms on graphs. Such algorithms can be used for determining the existence of certain paths (model checking), the existence of certain strategies (game solving), and the probabilities of certain events (performance analysis). We classify the algorithms according to the value domain (Boolean, probabilistic, or quantitative); according to the graph structure (nondeterministic, probabilistic, or multi-player); according to the desired property of paths (Borel level 1, 2, or 3); and according to the alternation depth and convergence rate of fixpoint computations.

[46] Eleftherios Matsikoudis and Edward A. Lee, Unpublished article, "Feedback in Strictly Causal Systems," January, 2007.

We ask whether strictly causal components form well defined systems when arranged in feedback configurations. The standard interpretation for such configurations induces a fixed-point constraint on the function modeling the component involved. We bring together ideas from the areas of set theory, order theory, and theory of generalized ultrametric spaces to establish the existence of a unique fixed point for any such function constructively, what has resisted more traditional approaches and has been a much coveted albeit elusive goal.

[47] Thomas A. Henzinger, "Games, time, and probability: Graph models for system design and analysis," Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), Lecture Notes in Computer Science, Springer, January, 2007.

Digital technology is increasingly deployed in safety-critical situations. This calls for systematic design and verification methodologies that can cope with three major sources of system complexity: concurrency, real time, and uncertainty. We advocate a two-step process: formal modeling followed by algorithmic analysis (or, "model building" followed by "model checking"). We model the concurrent components of a reactive system as potential collaborators or adversaries in a multi-player game with temporal objectives, such as system safety. The real-time aspect of embedded systems requires models that combine discrete state transitions and continuous state evolutions. Uncertainty in the environment is naturally modeled by probabilistic state changes. As a result, we obtain three orthogonal extensions of the basic state-transition graph model for reactive systems --game graphs, timed graphs, and stochastic graphs-- as well as combinations thereof. In this short text, we provide a uniform exposition of the underlying definitions. For verification algorithms, we refer the reader to the literature.

[48] Alex A. Kurzhanskiy, Pravin Varaiya, "Ellipsoidal Techniques for Reachability Analysis of Discrete-Time Linear Systems," *IEEE Transactions Automatic Control*, 52(1):26-38, January 2007.

This paper describes the computation of reach sets for discrete-time linear control systems with time-varying coefficients and ellipsoidal bounds on the controls and initial conditions. The algorithms construct external and internal ellipsoidal approximations that touch the reach set boundary from outside and from inside. Recurrence relations describe the time evolution of these approximations. An essential part of the paper deals with singular discrete-time linear systems.

[49] Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Aniruddha Gokhale, Sandeep Neema, and Jonathan Sprinkle, Paul A. Fishwick, "Domain-Specific Modeling (in CRC Handbook of Dynamic System Modeling)," 7, (in publication), CRC Press, 2007.

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.

[50] A. Abate S. Amin and M. Prandini and J. Lygeros and S. Sastry, A. Bemporad A. Bicchi and G. Buttazzo, "Computational Approaches to Reachability Analysis of Stochastic Hybrid Systems," 4-17, 4416, Springer Verlag, 2007.

This work investigates some of the computational issues involved in the solution of probabilistic reachability problems for discrete time, controlled stochastic hybrid systems. It is first argued that, under rather weak continuity assumptions on the stochastic kernels that characterize the dynamics of the system, the numerical solution of a discretized version of the probabilistic reachability problem is guaranteed to converge to the optimal one, as the discretization level decreases. With reference to a benchmark problem, it is then discussed how some of the structural properties of the hybrid system under study can be exploited to solve the probabilistic reachability problem more efficiently. Possible techniques that can increase the scale-up potential of the proposed numerical approximation scheme are suggested.

[51] A. Abate, A. D'Innocenzo, G. Pola, M.D. Di Benedetto and S. Sastry, A. Bemporad and A. Bicchi and G. Buttazzo, "The Concept of Deadlock and Livelock in Hybrid Control Systems," 628-632, 4416, Springer Verlag, 2007.

This short paper qualitatively introduces the definition of the concepts of Deadlock and Livelock for a general class of Hybrid Control Systems (HCS). Such a characterization hinges on three important aspects: firstly, the concept of composition of HCS; secondly, the general concept of specifications and their composition for HCS; finally, the dynamical structure and behaviors of HCS. The first aspect is introduced in a novel manner, including ideas from the literature of discrete transition systems and accounting for concepts such as that of dynamical feedback interconnection. The second point includes general properties that are of interest from a systems and control theory perspective. The third part categorizes the diverse and possibly pathological behaviors

that are distinctive of HCS. A first look at the problem of Deadlock and Livelock Verification concludes the manuscript.

[52] S. Amin and A. Abate and M. Prandini and J. Lygeros and S. Sastry, J. Hespanha and A. Tiwari, "Reachability analysis for controlled discrete time stochastic hybrid systems," 49-63, 3927, Springer Verlag, 2007.

A model for discrete time stochastic hybrid systems whose evolution can be influenced by some control input is proposed in this paper. With reference to the introduced class of systems, a methodology for probabilistic reachability analysis is developed that is relevant to 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 characterizing through some optimal cost function the set of initial conditions for the system such that safety is guaranteed with sufficiently high probability. The proposed methodology is applied to the problem of regulating the average temperature in a room by a thermostat controlling a heater.

[53] Aaron D. Ames, "Homotopy Meaningful Hybrid Model Structures," American Mathematical Society, 2007.

Hybrid systems are systems that display both discrete and continuous behavior and, therefore, have the ability to model a wide range of robotic systems such as those undergoing impacts. The main observation of this paper is that systems of this form relate in a natural manner to very special diagrams over a category, termed hybrid objects. Using the theory of model categories, which provides a method for "doing homotopy theory" on general categories satisfying certain axioms, we are able to understand the homotopy theoretic properties of such hybrid objects in terms of their "non-hybrid" counterparts. Specifically, given a model category, we obtain a "homotopy meaningful" model structure on the category of hybrid objects over this category with the same discrete structure, i.e., a model structure that relates to the original non-hybrid model structure by means of homotopy colimits, which necessarily exist. This paper, therefore, lays the groundwork for "hybrid homotopy theory."

[54] A. D. Ames and S. Sastry, "Hybrid Geometric Reduction of Hybrid Systems," Submitted to the IEEE Conference on Decision and Control, December, 2006.

This paper presents a unifying framework in which to carry out the hybrid geometric reduction of hybrid systems, generalizing classical reduction to a hybrid setting. Utilizing hybrid category theory, all of the major ingredients necessary for classical reduction can be hybridized through the notion of a hybrid object and a hybrid morphism over a general category. By leveraging the results of Marsden and Weinstein, we are able to show that when there is a hybrid symplectic manifold (the hybrid phase space) on which a hybrid Lie group acts symplectically, we can reduce the hybrid phase space to another hybrid symplectic manifold in which the hybrid symmetries are "divided out." In addition, hybrid trajectories of a hybrid Hamiltonian on the hybrid phase space determine corresponding hybrid trajectories on the reduced hybrid space.

[55] Alessandro Abate, Ashish Tiwari and S. Shankar Sastry, Technical report, "The concept of Box Invariance for biologically-inspired dynamical systems," EECS Department - University of California, at Berkeley, December, 2006.

In this paper we introduce a special notion of Invariance Set for certain classes of dynamical systems: the concept has been inspired by our experience with models drawn from Biology. We claim that Box Invariance, that is, the existence of "boxed" invariant regions, is a characteristic of many biologically-inspired dynamical models, especially those derived from stoichiometric reactions. Moreover, box invariance is quite useful for the verification of safety properties of such systems. This paper presents effective characterization of this notion for linear and affine systems, the study of the dynamical properties it subsumes, computational aspects of checking for box invariance, and a comparison with related concepts in the literature. The concept is illustrated using two models from biology.

[56] Allen Y. Yang, John Wright, Shankar Sastry, and Yi Ma, Technical report, "Unsupervised Segmentation of Natural Images via Lossy Data Compression," UC Berkeley, UCB/EECS-2006-195, December, 2006.

In this paper, we cast natural-image segmentation as a problem of clustering texture features as multivariate mixed data. We model the distribution of the texture features using a mixture of Gaussian distributions. However, unlike most existing clustering methods, we allow the mixture components to be degenerate or nearly-degenerate. We contend that this assumption is particularly important for mid-level image segmentation, where degeneracy is typically introduced by using a common feature representation for different textures. We show that such a mixture distribution can be effectively segmented by a simple agglomerative clustering algorithm derived from a lossy data compression approach. Using simple fixed-size Gaussian windows as texture features, the algorithm segments an image by minimizing the overall coding length of all the feature vectors. In terms of a variety of performance indices, our algorithm compares favorably against other well-known image segmentation methods on the Berkeley image database.

[57] Ye Zhou and Edward A. Lee, "A Causality Interface for Deadlock Analysis in Dataflow," EMSOFT 2006, October, 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.

[58] Abhijit Davare, Jike Chong, Qi Zhu, Douglas Densmore and Alberto Sangiovanni-Vincentelli, "An Overlap-based MILP Formulation for Task Allocation and Scheduling," submitted, October, 2006.

The deployment of applications on heterogeneous multi-core platforms is one of the most important challenges in the embedded systems design flow. In this paper, we develop an automated approach to tackle this problem that uses an efficient and extensible Mixed Integer Linear Programming (MILP) formulation. We show the effectiveness of our approach with extensive computational testing. For a case study involving the deployment of a Motion JPEG encoder application onto a Xilinx Virtex II Pro FPGA platform, we demonstrate that our automated approach can accurately capture the design space and yield systems that are competitive with manual designs.

[59] Arkadeb Ghosal, Thomas A. Henzinger, Daniel Iercan, Christoph Kirsch, Alberto Sangiovanni-Vincentelli., " A Hierarchical Coordination Language for Interacting Real-Time Tasks," EMSOFT 2006, October, 2006.

We designed and implemented a new programming language called Hierarchical Timing Language (HTL) for hard real-time systems. Critical timing constraints are specified within the language, and ensured by the compiler. Programs in HTL 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 supporting 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. HTL is a coordination language, in that individual tasks can be implemented in "foreign" languages. As a case study, we present a distributed HTL implementation of an automotive steer-by-wire controller.

[60] , "6th OOPSLA Workshop On Domain-Specific Modeling," Juha-Pekka Tolvanen, Matti Rossi, Jonathan Sprinkle, University of Jyvaskala, October, 2006.

The 6th DSM workshop featured 22 research and position papers describing new ideas at either a practical or theoretical level. On the practical side, several papers described application of modeling techniques within a specific domain. In addition to industrial projects, several authors from academia presented research ideas that initiate and forward the technical underpinnings of domain-specific modeling. Workshop had over 40 participants.

[61] Bor-Yuh Evan Chang, Matthew Harren, and George C. Necula, "Analysis of Low-Level Code Using Cooperating Decompilers," The 13th International Static Analysis Symposium (SAS), Kwangkeun Yi, 318-335, September, 2006;

Analysis or verification of low-level code is useful for minimizing the disconnect between what is verified and what is actually executed and is necessary when source code is unavailable or is, say, intermingled with inline assembly. We present a modular framework for building pipelines of cooperating decompilers that gradually lift the level

of the language to something appropriate for source-level 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. 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 Java-like 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.

[62] Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Marielle Stoelinga, "Quantitative Compositional Pricing," QEST 06, September, 2006.

We present a compositional theory of system verification, where specifications assign real-numbered costs to systems. These costs can express a wide variety of quantitative system properties, such as resource consumption, price, or a measure of how well a system satisfies its specification. The theory supports the composition of systems and specifications, and the hiding of variables. Boolean refinement relations are replaced by real-numbered distances between descriptions of a system at different levels of detail. We show that the classical Boolean rules for compositional reasoning have quantitative counterparts in our setting. While our general theory allows costs to be specified by arbitrary cost functions, we also consider a class of linear cost functions, which give rise to an instance of our framework where all operations are computable in polynomial time.

[63] Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger, "Strategy Improvement for Concurrent Reachability Games," QEST 06, September, 2006.

A concurrent reachability game is a two-player game played on a graph: at each state, the players simultaneously and independently select moves; the two moves determine jointly a probability distribution over the successor states. The objective for player 1 consists in reaching a set of target states; the objective for player 2 is to prevent this, so that the game is zero-sum. Our contributions are two-fold. First, we present a simple proof of the fact that in concurrent reachability games, for all epsilon>0, memoryless epsilon-optimal strategies exist. A memoryless strategy is independent of the history of plays, and an epsilon-optimal strategy achieves the objective with probability within epsilon of the value of the game. In contrast to previous proofs of this fact, which rely on the limit behavior of discounted games using advanced Puisieux series analysis, our proof is elementary and combinatorial. Second, we present a strategy-improvement (a.k.a. policy-iteration) algorithm for concurrent games with reachability objectives.

[64] Krishnendu Chatterjee, "Concurrent Games with Tail Objectives," CSL 06, September, 2006.

We study infinite stochastic games played by two-players over a finite state space, with objectives specified by sets of infinite traces. The games are concurrent (players make moves simultaneously and independently), stochastic (the next state is determined by a

probability distribution that depends on the current state and chosen moves of the players) and infinite (proceeds for infinite number of rounds). The analysis of concurrent stochastic games can be classified into: quantitative analysis, analyzing the optimum value of the game; and qualitative analysis, analyzing the set of states with optimum value 1. We consider concurrent games with tail objectives, i.e., objectives that are independent of the finite-prefix of traces, and show that the class of tail objectives is strictly richer than the omega-regular objectives. We develop new proof techniques to extend several properties of concurrent games with omega-regular objectives to concurrent games with tail objectives. We prove the positive limit-one property for tail objectives, that states for all concurrent games if the optimum value for a player is positive for a tail objective Phi at some state, then there is a state where the optimum value is 1 for Phi, for the player. We also show that the optimum values of zero-sum (strictly conflicting objectives) games with tail objectives can be related to equilibrium values of nonzero-sum (not strictly conflicting objectives) games with simpler reachability objectives. A consequence of our analysis presents a polynomial time reduction of the quantitative analysis of tail objectives to the qualitative analysis for the sub-class of one-player stochastic games (Markov decision processes).

[65] Krishnendu Chatterjee, "Nash Equilibrium for Upward-Closed Objectives," CSL 06, September, 2006.

We study infinite stochastic games played by n-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 players' goals are not necessarily conflicting), and undiscounted. We show that if each player has an upward-closed objective, then there exists an epsilon-Nash equilibrium in memory less strategies, for every epsilon>0; and exact Nash equilibria need not exist. Upward-closure of an objective means that if a set Z of infinitely repeating states is winning, then all supersets of Z of infinitely repeating states are also winning. Memoryless strategies are strategies that are independent of history of plays and depend only on the current state. We also 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 games with upward-closed objectives for all players can be computed by computing epsilon-Nash equilibrium values of nonzero-sum concurrent games with reachability objectives for all players and a polynomial procedure. As a consequence we establish that values of an epsilon-Nash equilibrium can be computed in TFNP (total functional NP), and hence in EXPTIME.

[66] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger and Jean-Francois Raskin, "Algorithms for Omega-Regular Games with Imperfect Information," CSL 06, September, 2006.

We study observation-based strategies for two-player turn-based games on graphs with omega-regular objectives. An observation-based strategy relies on imperfect information

about the history of a play, namely, on the past sequence of observations. Such games occur in the synthesis of a controller that does not see the private state of the plant. Our main results are twofold. First, we give a fixed-point algorithm for computing the set of states from which a player can win with a deterministic observation-based strategy for any omega-regular objective. The fixed point is computed in the lattice of antichains of state sets. This algorithm has the advantages of being directed by the objective and of avoiding an explicit subset construction on the game graph. Second, we give an algorithm for computing the set of states from which a player can win with probability 1 with a randomized observation-based strategy for a Buchi objective. This set is of interest because in the absence of perfect information, randomized strategies are more powerful than deterministic ones. We show that our algorithms are optimal by proving matching lower bounds.

[67] Thomas A. Henzinger and Vinayak Prabhu, "Timed alternating-time temporal logic," FORMATS 2006, September, 2006.

We add freeze quantifiers to the game logic ATL in order to specify real-time objectives for games played on timed structures. We define the semantics of the resulting logic TATL by restricting the players to physically meaningful strategies, which do not prevent time from diverging. We show that TATL can be model checked over timed automaton games. We also specify timed optimization problems for physically meaningful strategies, and we show that for timed automaton games, the optimal answers can be approximated to within any degree of precision.

[68] Pannag R Sanketi, J. Carlos Zavala, J. K. Hedrick, M. Wilcutts, T. Kaga, "A Simplified Catalytic Converter Model for Automotive Coldstart Applications with Adaptive Parameter Fitting," 8th International Symposium on Advanced Vehicle Control, August, 2006; .

It is well known that a major portion of the unburned hydrocarbon (HC) emissions in a typical drive cycle of an automotive engine are produced in the initial 1-2 minutes of operation, commonly called as the "coldstart" period. Catalyst light-off is essential for reducing these emissions. Model-based paradigm is used to develop a control-oriented, thermodynamics based simple catalyst model for coldstart analysis. The catalyst thermal submodel is modeled as a hybrid system consisting of three discrete states and one continuous state. The discrete states are "initial warm-up", "evaporation of condensed gas" and "light-off". The continuous dynamics consists of the catalyst temperature. In each of the discrete states, energy balance of a control mass is used to model the catalyst temperature. Parameter adaptation algorithm (PAA) is used to identify the parameters in each of the discrete states. Effectiveness of the average values of the adjusted parameters is also given. 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.

[69] Takashi Nagata, Hwan Hur, Masayoshi Tomizuka, "Model-Based Control for Smooth Gear Shifting by Engine-AT Collaboration," 8th International Symposium on Advanced Vehicle Control (AVEC '06), 635-640, August, 2006.

A collaborative control scheme between engine and automatic transmission (AT) gearbox is proposed to realize a smoother gear shifting for improving riding comfort. A systematic procedure is developed to obtain a desired engine torque profile and a desired hydraulics actuation profile for the AT gearbox which will minimize shocks due to gear shifting. Tracking controls must be applied to both the engine and the AT gearbox to follow these profiles. For the model-based approach, an AT gearbox model is devised using hybrid systems techniques to represent the discrete-event nature of automatic transmissions. Numerical evaluations have shown the effectiveness of the proposed scheme.

[70] Mark L. McKelvin, Jr., Claudio Pinello, Sri Kanajan, Joseph Wysocki, Alberto Sangiovanni-Vincentelli, "Model-Based Design of Heterogeneous Systems for Fault Tree Analysis," 24th International System Safety Conference, Rodney J. Simmons, Ph.D., Norman J. Gauthier, System Safety Society, 400-409, August, 2006.

We introduce a model-based approach to heterogeneous system design that enables the automatic generation of fault trees for analyzing system reliability properties. This approach extends our previous work that addressed the generation of fault trees from a dataflow model. In this new context, heterogeneous systems are composed of interacting discrete-time components, such as an electronic feedback controller, and continuous-time components, such as a plant. More recent work in computer-aided fault-tree generation methods is based on functional models of the system to produce a system fault tree automatically. Yet, most of these approaches were not applied to heterogeneous systems. Furthermore, these approaches continued to rely on intuition to create fault trees. Since in this approach fault tree generation is disjoint from the system modeling, consistency problems may arise when the structure and behavior of the system model is not accurately reflected. Our approach is different since we use a model of the system specified as a set of mathematical equations to derive the system fault modes and ultimately produce fault trees for heterogeneous systems.

[71] Krishnendu Chatterjee and Thomas A. Henzinger, "Strategy Improvement for Stochastic Rabin and Streett Games," CONCUR 06, August, 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 Rabin or Streett objectives. These games are NP-complete and coNP-complete, respectively. The value of the game for a player at a state s given an objective Phi is the maximal probability with which the player can guarantee the satisfaction of Phi from s. We present a strategy-improvement algorithm to compute values in stochastic Rabin games, where an improvement step involves solving Markov decision processes (MDPs) and nonstochastic Rabin games. The algorithm also computes values for stochastic Streett games but does not directly yield an optimal strategy for Streett

objectives. We then show how to obtain an optimal strategy for Streett objectives by solving certain nonstochastic Streett games.

[72] Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, "Algorithms for Buchi Games," GDV 06, August, 2006.

The classical algorithm for solving Buchi games requires time O(n * m) for game graphs with n states and m edges. For game graphs with constant outdegree, the best known algorithm has running time O(n*n /log n). We present two new algorithms for Buchi games. First, we give an algorithm that performs at most O(m) more work than the classical algorithm, but runs in time O(n) on infinitely many graphs of constant outdegree on which the classical algorithm requires time O(n*n). Second, we give an algorithm with running time O(n * m *log delta(n)/log n), where 1<= delta(n) <= n, is the outdegree of the game graph. Note that this algorithm performs asymptotically better than the classical algorithm if delta(n)=O(log n).

[73] Xiaojun Liu, Eleftherios Matsikoudis, and Edward A. Lee, "Modeling Timed Concurrent Systems," CONCUR 2006 - Concurrency Theory, 17th International Conference, Christel Baier and Holger Hermanns, 1-15, August, 2006.

Timed concurrent systems are widely 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, that is, sets of (semantically) time-stamped events. The denotational semantics of such systems is traditionally formulated in a metric space, wherein causal components are modeled as 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.

[74] Thomas A. Henzinger and Joseph Sifakis, "The embedded systems design challenge," Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science, Springer, August, 2006.

We summarize some current trends in embedded systems design and point out some of their characteristics, such as the chasm between analytical and computational models, and the gap between safety-critical and best-effort engineering practices. We call for a coherent scientific foundation for embedded systems design, and we discuss a few key demands on such a foundation: the need for encompassing several manifestations of heterogeneity, and the need for constructivity in design. We believe that the development

of a satisfactory Embedded Systems Design Science provides a timely challenge and opportunity for reinvigorating computer science.

[75] Qi Zhu, Abhijit Davare and Alberto Sangiovanni-Vincentelli, "A Semantic-Driven Synthesis Flow for Platform-Based Design," submitted to Fourth ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE'06), July, 2006.

The separation of concerns between functionality and architecture is a powerful technique used to facilitate design reuse at the system level. This separation of concerns and the successive refinement of the design by mapping functionality onto architecture are the core concepts in platform-based design. The goal of mapping is to optimize a set of objective functions while satisfying constraints on the mapped design. The mapping step can be seen as a synthesis process. While applying the methodology to a number of case studies, we have found that to gain the benefits of correct-by-construction deployment and rapid design space exploration, neither the semantics nor the abstraction levels for modeling can be chosen in an ad-hoc manner. In this paper, we propose a semantics-driven synthesis flow, in which the abstraction level and operational semantics are determined formally by using the concept of a common semantic domain between functionality and architecture. By doing so, a formal synthesis procedure can be defined and algorithms for automatic optimal mapping derived. By applying this approach to the previously mentioned case studies, we demonstrate how this approach can be used to significantly improve the effectiveness of platform-based design.

[76] A. D. Ames, R. D. Gregg, E. D. B. Wendel and S. Sastry, "Towards the Geometric Reduction of Controlled Three-Dimensional Robotic Bipedal Walkers," Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, July, 2006.

The purpose of this paper is to apply methods from geometric mechanics to the analysis and control of robotic bipedal walkers. We begin by introducing a generalization of Routhian reduction, functional Routhian Reduction, which allows for the conserved quantities to be functions of the cyclic variables rather than constants. Since bipedal robotic walkers are naturally modeled as hybrid systems, which are inherently nonsmooth, in order to apply this framework to these systems it is necessary to first extend functional Routhian reduction to a hybrid setting. We apply this extension, along with potential shaping and controlled symmetries, to derive a feedback control law for a three-dimensional bipedal walker that provably results in walking gaits on flat ground.

[77] A. Abate, M. Chen and S. Sastry, "Analysis of an Implementable Application Layer Scheme for Flow Control over Wireless Networks," Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, July, 2006.

This paper deals with the problem of congestion control and packet exchange on a wireless network. The mathematical model of the protocol is inspired by and extends a known fluid flow scheme for the control of congestion on a wired network. The necessity to introduce a specific wireless model is motivated by the presence of channel error; often

this error (due to intrinsic noise or channel corruption) is not known exactly. This motivates the approximation of parts of the structure of the model with binary functions, whose switching point can be precisely known. These new discontinuous elements, while in practice greatly simplifying the structure of the algorithm (they carry a single bit of information), complicate the theoretical analysis of its dynamical properties. We therefore approximate them with continuous functions with proper limiting behavior: they thus preserver the simple shape and yield themselves to analysis as well. Given this setup, we then investigate the important issues of existence and uniqueness of the equilibrium for the dynamical system, and of local asymptotic stability. Furthermore, we show that this equilibrium solves a concave net utility optimization problem, of which the classical one for wired networks is a special case. The take away point of this work is that the scheme we propose to handle the traffic on a wireless network is not only innovative and meaningful, but has also the potential to be modified and translated into practical implementations.

[78] Adam Cataldo, Edward Lee, Xiaojun Liu, Eleftherios Matsikoudis, and Haiyang Zheng, "A Constructive Fixed-Point Theorem and the Feedback Semantics of Timed Systems," Workshop on Discrete Event Systems, July, 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

[79] Aaron D. Ames, Robert D. Gregg, Eric D.B. Wendel and Shankar Sastry, "Towards the Geometric Reduction of Controlled Three-Dimensional Bipedal Robotic Walkers," 3rd Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, July, 2006.

The purpose of this paper is to apply methods from geometric mechanics to the analysis and control of bipedal robotic walkers. We begin by introducing a generalization of Routhian reduction, functional Routhian Reduction, which allows for the conserved quantities to be functions of the cyclic variables rather than constants. Since bipedal

robotic walkers are naturally modeled as hybrid systems, which are inherently nonsmooth, in order to apply this framework to these systems it is necessary to first extend functional Routhian reduction to a hybrid setting. We apply this extension, along with potential shaping and controlled symmetries, to derive a feedback control law that provably results in walking gaits on flat ground for a three-dimensional bipedal walker given walking gaits in two-dimensions.

[80] Aaron D. Ames, Robert D. Gregg, Haiyang Zheng, Prof. Shankar Sastry, "Is There Life After Zeno? Taking Executions past the Breaking (Zeno) Point," 2007 American Control Conference, ACS, June, 2006; Presented in conjunction with BEARS 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.

[81] Shinjiro Kakita, Yosinori Watanabe, Douglas Densmore, Abhijit Davare, Alberto Sangiovanni-Vincentelli, "Functional Model Exploration for Multimedia Applications via Algebraic Operators," ACSD 2006 - Sixth International Conference on Application of Concurrency to System Design, June, 2006.

An optimized functional design space exploration method for multimedia applications is proposed. 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 use 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.

[82] A. D. Ames and S. Sastry, " Hybrid Routhian Reduction of Lagrangian Hybrid Systems," American Control Conference, June, 2006.

This paper extends Routhian reduction to a hybrid setting, i.e., to systems that display both continuous and discrete behavior. We begin by considering a Lagrangian together with a configuration space with unilateral constraints on the set of admissible configurations. This naturally yields the notion of a hybrid Lagrangian, from which we obtain a Lagrangian hybrid system in a way analogous to the association of a Lagrangian vector field to a Lagrangian. We first give general conditions on when it is possible to reduce a cyclic Lagrangian hybrid system, and explicitly compute the reduced Lagrangian hybrid system in the case when it is obtained from a cyclic hybrid Lagrangian.

[83] A. D. Ames and S. Sastry, "Hybrid Cotangent Bundle Reduction of Simple Hybrid Mechanical Systems with Symmetry," American Control Conference, June, 2006.

This paper begins by introducing the notion of a simple hybrid mechanical system, which generalizes mechanical systems to include unilateral constraints on the configuration space. From such a system we obtain, explicitly, a simple hybrid system. The main contribution of this paper is to provide conditions on when it is possible to reduce the phase space of hybrid systems obtained from simple hybrid mechanical systems, and general simple hybrid systems, due to symmetries in the systems. Specifically, given a Hamiltonian G-space---which is the ingredient needed to reduce continuous systems---we find conditions on the hybrid system and the G-space so that reduction can be carried out in a hybrid setting---conditions that are explicitly related to conditions on the original hybrid mechanical system.

[84] A. D. Ames, H. Zheng, R. D. Gregg and S. Sastry, "Is there Life after Zeno? Taking Executions Past the Breaking (Zeno) Point," American Control Conference, June, 2006.

Understanding Zeno phenomena plays an important role in understanding hybrid systems. A natural---and intriguing---question to ask is: what happens after a Zeno point? Inspired by the construction of Filippov, we propose a method for extending Zeno executions past a Zeno point for a class of hybrid systems: Lagrangian hybrid systems. We argue that after the Zeno point is reached, the hybrid system should switch to a holonomically constrained dynamical system, where the holonomic constraints are based on the unilateral constraints on the configuration space that originally defined the hybrid system. These principles are substantiated with a series of examples.

[85] A. Abate, A. Ames and S. Sastry, "Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems," Proceedings of the 25th American Control Conference, June, 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.

[86] A. Abate, A. Ames and S. Sastry, "A-Priori Detection of Zeno Behavior in Communication Networks Modeled as Hybrid Systems," Proceedings of the 25th American Control Conference, June, 2006.

In this paper, we show that the sufficient conditions for the existence of Zeno behavior in hybrid systems correctly predict such executions in a modeling instance of the fluid-flow approximation of the TCP-like protocol for wireless communication networks.

[87] A. Abate and A. Tiwari, "Box Invariance of hybrid and switched systems," Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems, IFAC, June, 2006.

This paper investigates the concept of box invariance for classes of hybrid and switched systems. After motivating and defining the notion, we present a concise summary of results on its characterization for single-domain dynamical systems. The notion is then extended to the case of hybrid and switched systems. We provide sufficient conditions for a hybrid or switched system to be box invariant. Models of many real systems, especially those drawn from biology, have been found to be box invariant. This paper illustrates the concept using a pharmacodynamic model of blood glucose metabolism.

[88] Aaron D. Ames and Shankar Sastry, "Hybrid Routhian Reduction of Lagrangian Hybrid Systems," American Control Conference, June, 2006.

This paper extends Routhian reduction to a hybrid setting, i.e., to systems that display both continuous and discrete behavior. We begin by considering a Lagrangian together with a configuration space with unilateral constraints on the set of admissible configurations. This naturally yields the notion of a hybrid Lagrangian, from which we obtain a Lagrangian hybrid system in a way analogous to the association of a Lagrangian vector field to a Lagrangian. We first give general conditions on when it is possible to reduce a cyclic Lagrangian hybrid system, and explicitly compute the reduced Lagrangian hybrid system in the case when it is obtained from a cyclic hybrid Lagrangian.

[89] Aaron D. Ames and Shankar Sastry, "Hybrid Cotangent Bundle Reduction of Simple Hybrid Mechanical Systems with Symmetry," American Control Conference, June, 2006.

This paper begins by introducing the notion of a simple hybrid mechanical system, which generalizes mechanical systems to include unilateral constraints on the configuration space. From such a system we obtain, explicitly, a simple hybrid system. The main contribution of this paper is to provide conditions on when it is possible to reduce the phase space of hybrid systems obtained from simple hybrid mechanical systems, and

general simple hybrid systems, due to symmetries in the systems. Specifically, given a Hamiltonian G-space—-which is the ingredient needed to reduce continuous systems—-we find conditions on the hybrid system and the G-space so that reduction can be carried out in a hybrid setting—-conditions that are explicitly related to conditions on the original hybrid mechanical system.

[90] Aaron D. Ames, Haiyang Zheng, Robert D. Gregg and Shankar Sastry, "Is there Life after Zeno? Taking Executions Past the Breaking (Zeno) Point," American Control Conference, June, 2006.

Understanding Zeno phenomena plays an important role in understanding hybrid systems. A natural and intriguing question to ask is: what happens after a Zeno point? Inspired by the construction of Filippov, we propose a method for extending Zeno executions past a Zeno point for a class of hybrid systems: Lagrangian hybrid systems. We argue that after the Zeno point is reached, the hybrid system should switch to a holonomically constrained dynamical system, where the holonomic constraints are based on the unilateral constraints on the configuration space that originally defined the hybrid system. These principles are substantiated with a series of examples.

[91] Haiyang Zheng, Edward A. Lee and Aaron D. Ames, Joao Hespanha, Ashish Tiwari, "Beyond Zeno: Get on with It!," Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 3927, 2006, 3-540-33170-0.

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.

[92] A. D. Ames, P. Tabuada and S. Sastry, "On the Stability of Zeno Equilibria," 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.

3. Outreach

3.1. 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.

3.2. 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.

3.2.1. 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.

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.

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: Autonomous Systems: Algorithms and Implementation (UCB, EECS 290n)
Instructor: Dr. Jonathan Sprinkle,
Prof. S. Shankar Sastry

Autonomous systems can be thought of as robust control systems that can use discrete and continuous control inputs to choose a locally or globally optimal future state. To say that an autonomous system is intelligent is considered redundant by many researchers; though the distinction can be made that intelligent autonomous systems should react in an intuitive manner to stimuli. This course approached the domain of autonomous systems from the perspective of algorithms and their implementation upon an existing software and hardware infrastructure. Our "driving" example will in fact be the newest DARPA Grand Challenge instantiation: the DARPA Urban Challenge.

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 real-time 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: 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.

3.2.2. SUPERB-IT Program

SUPERB-IT—2006

The Center for Hybrid and Embedded Software Systems is proud to sponsor four undergraduate students from diverse backgrounds and cultures to participate in SUPERB-IT. These students will interact with individual mentors throughout the summer, and perform research and supporting activities in the area of hybrid and embedded systems. This year, all participants were women engineers.

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 2006 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 four SUPERB-IT students in 2006, and organized projects (described below in the abstracts from their papers) in hybrid systems theory, wireless sensor networks, dynamical systems simulation, and autonomous systems. 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 2006 were Dominique Duncan, Nandita Mitra, Nashlie Sephus, and Heather Taylor. 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. Four 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/2006/superb/index.html

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.

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 from Kansas and now a rising senior at the University of Chicago, majoring in Mathematics and Polish Literature. She is working this summer with Mentor Alex Kurzhanskiy.

Her areas of interest are highway traffic control, stochastic systems, and biomedical applications of math and statistics. Her SUPERB work is the flow analysis and control of highway traffic. Dominique plans on pursuing a PhD in electrical engineering and will be applying to schools in Fall 2006.

She is Vice President of the University of Chicago Math Club, President of the UofC Polish American Student Association, member of Women in Science, and member of IEEE CSS Women in Control group.

Project: Tool for probabilistic safety verification of stochastic hybrid systems
Student: Nandita Andromeda Mitra—*Rutgers, the State University of New Jersey*
Mentor: Saurabh Amin and Alessandro Abate

Many safety critical systems like air traffic control involve modeling their behavior as hybrid systems. The effect of uncertain system dynamics and external inputs can be incorporated by modeling the system as a controlled stochastic hybrid system (SHS). Design of controllers for SHS that guarantees a certain safety criterion can be posed as a quantitative verification problem. The goal of this project is to develop a computational tool for stochastic reachability analysis of a benchmark SHS.

Nandita is a rising senior at Rutgers, the State University of New Jersey, majoring in Electrical Engineering. She is working with Saurabh Amin and Alessandro Abate and her project is making a toolbox for probabilistic safety verification of stochastic hybrid systems.

Nandita was born in Hawaii and went to Bangladesh with her parents. After studying two years in Bangladesh University of Engineering and Technology she transferred credit to Rutgers and started as a junior there from Fall 2005

She is the treasurer for IEEE, Rutgers for the academic year of 2006-2007.

Nandita is interested in wireless communication and has become interested in stochastic hybrid systems after coming to Berkeley. Her future plan is to continue working in these fields.

Project: Autopilot for an Ultra-Light, Flying Wing
Student: Nashlie H. Sephus—*Mississippi State University*
Mentor: Todd Templeton

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 expanded polypropylene (EPP) foam. This aircraft is also challenging from a control perspective because it is vulnerable to wind and can only carry a minimal payload. First, we develop optimal local trajectories given current wind conditions. These local trajectories are used to determine the path that the vehicle should try to maintain between widely-spaced waypoints, and should trade off between overshooting corners and maintaining the desired trajectory. The testing and results of this portion are performed in MATLAB in order to plot and view the optimizing functionality. We then implement these local trajectories in the Zagi autopilot (written in C) to enable it to follow an incrementally-specified global trajectory with future planning. Initial tests are performed using the CRRCsim simulator interfaced to the Zagi hardware. Results from these tests prove that the planning autopilot is better than the existing autopilot because it directs a path along three points at a time versus only one point. Also, this higher level planner is greedy in that it reaches a desired optimal trajectory without using extra minimizing functionality. Future work involves final testing on a real Zagi at Richmond Field Station.

Nashlie Sephus is a rising senior at Mississippi State University, majoring in Computer Engineering. She is working this summer on an autopilot for a crossbow aircraft with mentor Todd Templeton.

Nashlie's research interests include embedded systems, robotics, and high performance computing. She has also completed three semesters of Cooperative Education in the Information Technology field.

After graduation, she plans to enter graduate school. Her hobbies and other interests include music, tennis, golf, pool, and shopping.

Project: Multihop Routing Simulation of TinyOS-Based Wireless Sensor Networks in Viptos
Student: Heather Taylor—*University of Vermont*
Mentor: Elaine Cheong

Wireless Sensor Networks are a burgeoning area of research and applications 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 system for the Berkeley Motes, which are small embedded systems capable of

collecting audio, temperature and other kinds of sensor data and transmitting it via a radio. A TinyOS simulator called TOSSIM is used, a key piece of which is the ability to simulate a network topology once it is functioning. Viptos extends the capabilities of the TinyOS simulator to allow simulation of heterogeneous networks. The final goal of this research is to create a graphical representation of the communication between motes in a multihop network simulation in Viptos. This will be done by adding an entity to Viptos which will collect transmitted information and display a graphical representation of that communication between nodes.

Heather Taylor is a rising senior studying Electrical Computer Engineering at the University of Vermont. This summer she is working with her mentor Elaine Cheong.

Her areas of interest include Wireless Sensor Networks, RFID technology, and embedded systems. This summer she will be working on multihop applications in Viptos.

Her involvement with UVM chapters include: Chair, IEEE; Webmaster, SWE; McNair Scholars, URECA!, & Tau Beta Pi. She plans to pursue her PhD after graduation.

3.2.3. 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 2006 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 2006 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. The projects are listed below.

Project: Radio Controlled Car Controller
Undergraduates: Jessica Kane, Thao Nguyen
Graduate Student Mentor: Graham Hemingway

> Overview: Using the existing infrastructure for autonomous helicopter flight, students have equipped a small radio-controlled car with sensors so that it can localize itself, then wrote a controller in Simulink to have the car do things such as follow a line autonomously.

Project: Hybrid System Modeling / Fault Diagnosis
Undergraduates: Nathaniel Allotey, Brian Turnbull
Graduate Student Mentor: Wu Jian

> Overview: Students used Simulink/Stateflow to build a model of the three tank system, then designed Stateflow controllers and ran those on the actual system, and also performed diagnosis experiments. Also experimented with adaptive control.

Project: Controlling Lego Robots Using a Synchronous-Reactive Model of Computation
Undergraduates: Javier Lara, Darren White
Graduate Student Mentor: Ethan Jackson

> Overview: Using the synchronous-reactive model of computation, students have built various controllers for the Lego Mindstorm robots, to follow a geometric pattern on the floor.

Project: Exploring with Lego Robots
Undergraduates: Daniel Limbrick, Emily Sherrill
Graduate Student Mentor: Daniel Balasubramanian

Overview: Students used a Max232 interface to adapt a Lego Mindstorms robot to use bluetooth communication, then using java, had the robot explore a maze, communicate the layout of the maze back to a computer, then built a controller to allow the robot to be driven from the computer screen.

4. Publications and Products

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

4.1. Journal Publications

- Arkadeb Ghosal, Sri Kanajan, Randall Urbance, Alberto Sangiovanni-Vincentelli, *SAE 2007 Transactions: Journal of Passenger Cars: Electronic and Electrical Systems*, "An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures," pp. 844-856, March 2007; ISBN: 978-0-7680-1839-4.

- Alberto L. Sangiovanni-Vincentelli, *Proceedings of the IEEE*, "Quo Vadis SLD: Reasoning about Trends and Challenges of System-Level Design," 95(3):467-506, March 2007.

- Alex A. Kurzhanskiy, Pravin Varaiya, *IEEE Transactions Automatic Control*, "Ellipsoidal Techniques for Reachability Analysis of Discrete-Time Linear Systems," 52(1):26-38, January 2007.

- A. Agrawal, Gabor Karsai, Sandeep Neema, Feng Shi, Attila Vizhanyo, *Journal on Software and System Modeling*, "The Design of a Language for Model Transformations," 5(3):261-288, September 2006.

- Sztipanovits, J., Bay, J., Rohrbough, L., Sastry, S., Schmidt, D., Whitaker, N., Winter, D., *IEEE Computer*, "ESCHER: A New Technology Transitioning Model," 40(3):90-92, March 2007.

4.2. Conference Papers

- Aaron D. Ames, Robert D. Gregg, Haiyang Zheng, Prof. Shankar Sastry, "Is There Life After Zeno? Taking Executions past the Breaking (Zeno) Point," 2007 American Control Conference, ACS, June, 2006; Presented in conjunction with BEARS 2006.

- Bor-Yuh Evan Chang, Matthew Harren, and George C. Necula, "Analysis of Low-Level Code Using Cooperating Decompilers," The 13th International Static Analysis Symposium (SAS), Kwangkeun Yi (ed.), 318-335, September, 2006.

- Ye Zhou and Edward A. Lee, "A Causality Interface for Deadlock Analysis in Dataflow," Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'06), Sang Lyul Min, Wang Yi (eds.), ACM, 44-52, October, 2006.

- Pannag R Sanketi, J. Carlos Zavala, J. K. Hedrick, M. Wilcutts, T. Kaga, "A Simplified Catalytic Converter Model for Automotive Coldstart Applications with Adaptive Parameter Fitting," 8th International Symposium on Advanced Vehicle Control, August, 2006.

- Shinjiro Kakita, Yosinori Watanabe, Douglas Densmore, Abhijit Davare, Alberto Sangiovanni-Vincentelli, "Functional Model Exploration for Multimedia Applications via Algebraic Operators," ACSD 2006 - Sixth International Conference on Application of Concurrency to System Design, 229-238, June, 2006.

- Qi Zhu, Abhijit Davare and Alberto Sangiovanni-Vincentelli, "A Semantic-Driven Synthesis Flow for Platform-Based Design," submitted to Fourth ACM-IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE'06), July, 2006.

- Abhijit Davare, Jike Chong, Qi Zhu, Douglas Densmore and Alberto Sangiovanni-Vincentelli, "An Overlap-based MILP Formulation for Task Allocation and Scheduling," submitted, October, 2006.

- A. D. Ames and S. Sastry, "Hybrid Geometric Reduction of Hybrid Systems," 2006 45th IEEE Conference on Decision and Control, 923-929, December, 2006.

- A. D. Ames, R. D. Gregg, E. D. B. Wendel and S. Sastry, "Towards the Geometric Reduction of Controlled Three-Dimensional Robotic Bipedal Walkers," Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, July, 2006.

- A. D. Ames and S. Sastry, "Hybrid Routhian Reduction of Lagrangian Hybrid Systems," American Control Conference, June, 2006.

- A. D. Ames and S. Sastry, "Hybrid Cotangent Bundle Reduction of Simple Hybrid Mechanical Systems with Symmetry," American Control Conference, June, 2006.

- A. D. Ames, H. Zheng, R. D. Gregg and S. Sastry, "Is there Life after Zeno? Taking Executions Past the Breaking (Zeno) Point," American Control Conference, June, 2006.

- Antoon Goderisa, Christopher Brooks, lkay Altintas, Edward A. Lee, "Composing Different Models of Computation in Ptolemy II and Kepler," 2007 Proceedings, International Conference on Computational Science (ICCS), May, 2007; To appear at International Conference on Computational Science (ICCS) 2007.

- Takashi Nagata, Hwan Hur, Masayoshi Tomizuka, "Model-Based Control for Smooth Gear Shifting by Engine-AT Collaboration," 8th International Symposium on Advanced Vehicle Control (AVEC '06), 635-640, August, 2006.

- A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic Reachability for Safety and Regulation of Controlled Discrete-Time Stochastic Hybrid Systems," Proceedings of the 45th IEEE Conference on Decision and Control, IEEE, December, 2006.

- A. Abate, A. Ames and S. Sastry, "Error Bounds Based Stochastic Approximations and Simulations of Hybrid Dynamical Systems," Proceedings of the 25th American Control Conference, June, 2006.

- A. Abate, A. Ames and S. Sastry, "A-Priori Detection of Zeno Behavior in Communication Networks Modeled as Hybrid Systems," Proceedings of the 25th American Control Conference, June, 2006.

- A. Abate and A. Tiwari, "Box Invariance of hybrid and switched systems," Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems, IFAC, June, 2006.

- A. Abate, M. Chen and S. Sastry, "Analysis of an Implementable Application Layer Scheme for Flow Control over Wireless Networks," Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, July, 2006.

- Arkadeb Ghosal, Thomas A. Henzinger, Daniel Iercan, Christoph Kirsch, Alberto Sangiovanni-Vincentelli, "A Hierarchical Coordination Language for Interacting Real-Time Tasks," Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'06), Sang Lyul Min, Wang Yi (eds.), ACM, 132-141, October, 2006.

- Arkadeb Ghosal, Sri Kanajan, Randall Urbance, Alberto Sangiovanni-Vincentelli, "An Initial Study on Monetary Cost Evaluation for the Design of Automotive Electrical Architectures," SAE, April, 2007.

- Nadathur Satish, Kaushik Ravindran and Kurt Keutzer, "A Decomposition-based Constraint Optimization Approach for Statically Scheduling Task Graphs with Communication Delays to Multiprocessors," 10th Conference of Design, Automation and Test in Europe (DATE-07), 230-235, April, 2007.

- Abhijit Davare, Douglas Densmore, Trevor Meyerowitz, Alessandro Pinto, Alberto Sangiovanni-Vincentelli, Guang Yang, Haibo Zeng, Qi Zhu, "A Next-Generation Design Framework for Platform-based Design," DVCon 2007, February, 2007.

- Mark L. McKelvin, Jr., Claudio Pinello, Sri Kanajan, Joseph Wysocki, Alberto Sangiovanni-Vincentelli, "Model-Based Design of Heterogeneous Systems for Fault Tree Analysis," 24th International System Safety Conference, Rodney J. Simmons, Ph.D., Norman J. Gauthier (eds.), System Safety Society, 400-409, August, 2006.

- Krishnendu Chatterjee and Thomas A. Henzinger, "Strategy Improvement for Stochastic Rabin and Streett Games," CONCUR 2006 - Concurrency Theory, 17th International Conference, Christel Baier and Holger Hermanns (eds.), 375-389, August, 2006.

- Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Marielle Stoelinga, "Quantitative Compositional Pricing," QEST 06, September, 2006.

- Krishnendu Chatterjee, Luca de Alfaro and Thomas A. Henzinger, "Strategy Improvement for Concurrent Reachability Games," QEST 06, September, 2006.

- Krishnendu Chatterjee, "Concurrent Games with Tail Objectives," CSL 06, September, 2006.

- Krishnendu Chatterjee, "Nash Equilibrium for Upward-Closed Objectives," CSL 06, September, 2006.

- Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger and Jean-Francois Raskin, "Algorithms for Omega-Regular Games with Imperfect Information," CSL 06, September, 2006.

- Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, "Algorithms for Buchi Games," GDV 06, August, 2006.

- Krishnendu Chatterjee and Thomas A. Henzinger, "Assume-guarantee Synthesis," TACAS, March, 2007.

- Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, "Generalized Parity Games," FOSSACS 07, March, 2007.

- Krishnendu Chatterjee, "Optimal Strategy Synthesis for Stochastic Muller Games," FOSSACS 07, March, 2007.

- Thomas Huining Feng, Lynn Wang, Wei Zheng, Sri Kanajan, and Sanjit A. Seshia, "Automatic Model Generation for Black Box Real-Time Systems," Design, Automation and Test in Europe (DATE) Conference, April, 2007.

- Takashi Nagata, Masayoshi Tomizuka, "Engine Torque Control Based on Discrete Event Model and Disturbance Observer," ASME International Mechanical Engineering Congress and Exposition (IMECE2007), (submitted), May, 2007; Draft submitted in May 2007, to be presented in Nov. 2007.

- Sumitra Ganesh, Aaron Ames, Ruzena Bajcsy, "Composition of Dynamical Systems for Estimation of Human Body Dynamics," Proceedings of 10th International Conference on Hybrid Systems Computation and Control 2007, Alberto Bemporad, Antonio Bicchi, Giorgio Buttazzo (eds.), 702-706, April, 2007.

- Thomas A. Henzinger and Vinayak Prabhu, "Timed alternating-time temporal logic," Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, LCNS 4202, Asarin, Eugene; Bouyer, Patricia (eds.), 1-17, September, 2006.

- Gang Zhou, Man-Kit Leung, and Edward A. Lee, "A Code Generation Framework for Actor-Oriented Models with Partial Evaluation," Proceedings of International Conference on Embedded Software and Systems 2007, LNCS 4523, Y.-H. Lee et al. (ed.), 786-799, May, 2007.

- Xiaojun Liu, Eleftherios Matsikoudis, and Edward A. Lee, "Modeling Timed Concurrent Systems," CONCUR 2006 - Concurrency Theory, 17th International Conference, Christel Baier and Holger Hermanns (eds.), 1-15, August, 2006.

- Adam Cataldo, Edward Lee, Xiaojun Liu, Eleftherios Matsikoudis, and Haiyang Zheng, "A Constructive Fixed-Point Theorem and the Feedback Semantics of Timed Systems," Workshop on Discrete Event Systems, July, 2006.

- Aaron D. Ames and Shankar Sastry, "Hybrid Routhian Reduction of Lagrangian Hybrid Systems," American Control Conference, June, 2006.

- Aaron D. Ames and Shankar Sastry, "Hybrid Cotangent Bundle Reduction of Simple Hybrid Mechanical Systems with Symmetry," American Control Conference, June, 2006.

- Aaron D. Ames, Haiyang Zheng, Robert D. Gregg and Shankar Sastry, "Is there Life after Zeno? Taking Executions Past the Breaking (Zeno) Point," American Control Conference, June, 2006.

- Aaron D. Ames, Robert D. Gregg, Eric D.B. Wendel and Shankar Sastry, "Towards the Geometric Reduction of Controlled Three-Dimensional Bipedal Robotic Walkers," 3rd Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, July, 2006.

- Aaron D. Ames and Shankar Sastry, "Hybrid Geometric Reduction of Hybrid Systems," 45th IEEE Conference on Decision and Control, 923 - 929, December, 2006.

- Thomas A. Henzinger and Joseph Sifakis, "The embedded systems design challenge," Proceedings of the 14th International Symposium on Formal Methods (FM), Lecture Notes in Computer Science, Springer, August, 2006.

- Thomas A. Henzinger, "Games, time, and probability: Graph models for system design and analysis," Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), Lecture Notes in Computer Science, Springer, January, 2007.

- Wei Zheng, Marco Di Natale, Claudio Pinello, Paolo Giusto, Alberto Sangiovanni-Vincentelli, "Synthesis of task and message activation models in real-time distributed automotive systems," Design, Automation and Test in Europe, April, 2007.

- Marco Di Natale, Wei Zheng, Claudio Pinello, Paolo Giusto, Alberto Sangiovanni-Vincentelli, "Optimizing end-to-end latencies by adaptation of the activation events in distributed automotive systems," Real-Time and Embedded Technology and Applications Symposium, April, 2007.

- D. Balasubramanian, A. Narayanan, S. Neema, F. Shi, R. Thibodeaux, G. Karsai, "A Subgraph Operator for Graph Transformation Languages," Proceedings of 6th International Workshop on Graph Transformation and Visual Modeling Techniques, Braga, Portugal, March, 2007.

- G. Karsai, A. Narayanan, "On the Correctness of Model Transformations in the Development of Embedded Systems," Proceedings of the 2006 Monterey Workshop, October, 2006; (Paper presented at conference and submitted to publisher).

- Isaac Amundson, Manish Kushwaha, Xenofon Koutsoukos, Sandeep Neema, and Janos Sztipanovits, "Efficient Integration of Web Services in Ambient-aware Sensor Network Applications," 3rd IEEE/CreateNet International Workshop on Broadband Advanced Sensor Networks (BaseNets 2006), October, 2006.

- Karsai, G., Ledeczi, A., Neema, S., Sztipanovits, J., "The Model-Integrated Computing Toolsuite: Metaprogrammable Tools for Embedded Control System Design," Proc. of the IEEE Joint Conference CCA, ISIC and CACSD, Munich, Germany, 50-55, October, 2006.

- Jackson, E., Sztipanovits, J., "Towards A Formal Foundation For Domain Specific Modeling Languages," Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'06), Sang Lyul Min, Wang Yi (eds.), ACM, 53-63, October, 2006.

- Emerson M., Sztipanovits J., "Techniques for Metamodel Composition," OOPSLA – 6th Workshop on Domain Specific Modeling, 123-139, October, 2006.

- Emerson M., Duncavage S., Mathe J., Sztipanovits J., "WiNeSim: A Wireless Network Simulation Tool," Proceedings of the Sixth ACM International Conference on Embedded Software (EMSOFT'06), Sang Lyul Min, Wang Yi (eds.), ACM, October, 2006.

- Sztipanovits, J., "Towards the Compositional Specification of Semantics for Heterogeneous Domain-Specific Modeling Languages," Workshop on Foundation of Composition, October, 2006.

- Kai Chen, Janos Sztipanovits, Sandeep Neema, "A Case Study on Semantic Unit Composition," Workshop on Modeling in Software Engineering, ICSE 2007 (MISE 2007), May, 2007.

- Kai Chen, Janos Sztipanovits, Sandeep Neema, "Compositional Specification of Behavioral Semantics," Proceedings of Design Automation and Test in Europe Conference (DATE 07), 906-911, April, 2007.

- Manish Kushwaha, Isaac Amundson, Xenofon Koutsoukos, Sandeep Neema, and Janos Sztipanovits, "OASiS: A Programming Framework for Service-Oriented Sensor Networks," In IEEE/Create-Net COMSWARE 2007, January, 2007.

- Sztipanovits, J., "Model-based Software Development," ESMD-SW Workshop, NASA, March, 2007.

- I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and P. J. Mosterman, "A Method for Efficient Simulation of Hybrid Bond Graphs," International Conference on Bond Graph Modeling and Simulation (ICBGM 2007), 177-184, January, 2007.

- M. Daigle, I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Efficient Simulation of Component-Based Hybrid Models Represented as Hybrid Bond Graphs," Hybrid Systems: Computation and Control (HSCC 2007), Lecture Notes in Computer Science, vol. 4416, 680-683, April, 2007.

- Yang Zhao, Jie Liu and Edward A. Lee, "A Programming Model for Time-Synchronized Distributed Real-Time Systems," 13th IEEE Real Time and Embedded Technology and Applications Symposium, 2007. RTAS '07, 259 - 268, April, 2007.

- Edward A. Lee and Stephen Edwards, "Precision Timed (PRET) Computation in Cyber-Physical Systems," National Workshop on High Confidence Software Platforms for Cyber-Physical Systems: Research Needs and Roadmap, November, 2006.

- Edward A. Lee and Yang Zhao, "Reinventing Computing for Real Time," Proceedings of the 2006 Monterey Workshop, October, 2006.

- Yang Zhao, Edward A. Lee and Jie Liu, "Application of Programming Temporally Integrated Distributed Embedded Systems," Proceedings of 2006 IEEE 1588 Conference, October, 2006.

- Edward A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?," Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, October, 2006.

- Edward A. Lee, "Concurrent Semantics without the Notions of State or State Transitions," Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, LCNS 4202, Asarin, Eugene; Bouyer, Patricia (eds.), 18-31, September, 2006

4.3. Books, Reports, and Other One-Time Publications

- Haiyang Zheng, Edward A. Lee and Aaron D. Ames, Joao Hespanha, Ashish Tiwari (eds.), "Beyond Zeno: Get on with It!," 568-582, 3927, Springer Berlin/Heidelberg, 2006; From "Hybrid Systems: Computation and Control, Proceedings". 3927, ISBN: 3-540-33170-0.

- A. D. Ames, P. Tabuada and S. Sastry, "On the Stability of Zeno Equilibria," 34-48, Lecture Notes in Com, 3927, Springer-Verlag, 2006.

- Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Aniruddha Gokhale, Sandeep Neema, and Jonathan Sprinkle, Paul A. Fishwick (ed.), "Domain-Specific Modeling (in CRC Handbook of Dynamic System Modeling)," 7, (in publication), CRC Press, 2007.

- A. Abate S. Amin and M. Prandini and J. Lygeros and S. Sastry, A. Bemporad A. Bicchi and G. Buttazzo (eds.), "Computational Approaches to Reachability Analysis of Stochastic Hybrid Systems," 4-17, 4416, Springer Verlag, 2007.

- A. Abate, A. D'Innocenzo, G. Pola, M.D. Di Benedetto and S. Sastry, A. Bemporad and A. Bicchi and G. Buttazzo (eds.), "The Concept of Deadlock and Livelock in Hybrid Control Systems," 628-632, 4416, Springer Verlag, 2007.

- S. Amin and A. Abate and M. Prandini and J. Lygeros and S. Sastry, J. Hespanha and A. Tiwari (eds.), "Reachability analysis for controlled discrete time stochastic hybrid systems," 49-63, 3927, Springer Verlag, 2007.

- Aaron D. Ames, "Homotopy Meaningful Hybrid Model Structures," American Mathematical Society, 2007; To appear in "Topology and Robotics", AMS Contemporary Mathematics Series, 2007.

- Matthew Emerson. Sandeep Neema. Janos Sztipanovits; Insup Lee, Joseph Leung, Sang H. Son (eds.), "Handbook of Real-Time and Embedded Systems," CRC Press, 2006; ISBN: 1584886781.

- Alessandro Abate, Ashish Tiwari and S. Shankar Sastry, Technical report, "The concept of Box Invariance for biologically-inspired dynamical systems," University of California, Berkeley, UCB/EECS-2006-185, December, 2006.

- Joseph Gerard Makin and Alessandro Abate, Technical report, "A Neural Hybrid-System Model of the Basal Ganglia," University of California, Berkeley, UCB/EECS-2007-16, January, 2007.

- Allen Y. Yang, John Wright, Shankar Sastry, and Yi Ma, Technical report, "Unsupervised Segmentation of Natural Images via Lossy Data Compression," University of California, Berkeley, UCB/EECS-2006-195, December, 2006.

- Kai Chen, Janos Sztipanovits, Sandeep Neema, Technical report, "Compositional Specification of Behavioral Semantics," Institute for Software Integrated Systems (ISIS), ISIS-06-705, June, 2006.

- Edward A. Lee, Technical report, "Computing Foundations and Practice for Cyber-Physical Systems: A Preliminary Report," University of California, Berkeley, UCB/EECS-2007-72, May, 2007.

- Gang Zhou, Man-Kit Leung and Edward A. Lee, Technical report, "A Code Generation Framework for Actor-Oriented Models with Partial Evaluation," University of California, Berkeley, UCB/EECS-2007-29, February, 2007.

- Yang Zhao, Yuhong Xiong, Edward A. Lee, Xiaojun Liu and Lizhi C. Zhong, Technical report, "The Design and Application of Structured Types in Ptolemy II," University of California, Berkeley, UCB/EECS-2007-21, January, 2007.

- Christopher Brooks, Edward A. Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao and Haiyang Zheng, Technical report, "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II)," University of California, Berkeley, UCB/EECS-2007-7, February, 2007.

- Christopher Brooks, Edward A. Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao and Haiyang Zheng, Technical report, "Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture)," University of California, Berkeley, UCB/EECS-2007-8, February, 2007.

- Christopher Brooks, Edward A. Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao and Haiyang Zheng, Technical report, "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)," University of California, Berkeley, UCB/EECS-2007-9, February, 2007.

- Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, Technical report, "Strategy Logic," University of California, Berkeley, UCB/EECS-2007-78, May, 2007.

- Thomas Brihaye, Thomas A. Henzinger, Vinayak Prabhu and Jean-Francois Raskin, Technical report, "Minimum-Time Reachability in Timed Games," University of California, Berkeley, UCB/EECS-2007-47, April, 2007.

- Edward A. Lee, Xiaojun Liu and Stephen Andrew Neuendorffer, Technical report, "Classes and Inheritance in Actor-Oriented Design," University of California, Berkeley, UCB/EECS-2006-154, November, 2006.

- Stephen Edwards and Edward A. Lee, Technical report, "The Case for the Precision Timed (PRET) Machine," University of California, Berkeley, UCB/EECS-2006-149, November, 2006.

- Ye Zhou and Edward A. Lee, Technical report, "Causality Interfaces for Actor Networks," University of California, Berkeley, UCB/EECS-2006-148, November, 2006.

- Krishnendu Chatterjee, Thomas A. Henzinger and Nir Piterman, Technical report, "Generalized Parity Games," University of California, Berkeley, UCB/EECS-2006-144, November, 2006.

- Krishnendu Chatterjee, Rupak Majumdar and Thomas A. Henzinger, Technical report, "Stochastic Limit-Average Games are in EXPTIME," University of California, Berkeley, UCB/EECS-2006-143, November, 2006.

- Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger and Jean-Francois Raskin, Technical report, "Algorithms for Omega-Regular Games with Incomplete Information," University of California, Berkeley, UCB/EECS-2006-89, June, 2006.

- Krishnendu Chatterjee and Thomas A. Henzinger, Technical report, "Reduction of Stochastic Parity to Stochastic Mean-payoff Games," University of California, Berkeley, UCB/EECS-2006-140, November, 2006.

- Abhijit Davare, Jike Chong, Qi Zhu, Douglas Michael Densmore and Alberto L. Sangiovanni-Vincentelli, Technical report, "Classification, Customization, and Characterization: Using MILP for Task Allocation and Scheduling," University of California, Berkeley, UCB/EECS-2006-166, December, 2006.

- Ethan Jackson, Technical report, "The Software Engineering of Domain-Specific Modeling Languages: A Survey Through Examples," Institute For Software Integrated Systems (ISIS), ISIS-07-807, March, 2007.

4.4. 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.

4.4.1. 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 have 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.

4.4.2. 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.

4.4.3. "Foundations" The 2006-2007 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 "Foundations", which will examine how the work pursued in hybrid systems and embedded software systems can be used in other domains, as well as understanding how to apply foundational results appropriately. An additional angle in this thread is to present an interesting application which can be subdivided into its component parts, or examined from a theoretical standpoint. 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 Design Flow for the Development, Characterization, and Refinement of System Level Architectural Services"
  Douglas Densmore, UC Berkeley, May 15, 2007.
- "Automating Sequential Clock Gating for Power Optimization"
  Anmol Mathur, Calypto Design Systems, May 1, 2007.
- "Closed-loop impulse control and the theory of Hybrid Systems"
  Alexander B. Kurzhanski, UC Berkeley, April 24, 2007.
- "Automated Extraction of Inductive Invariants to Aid Model Checking"
  Mike Case, UC Berkeley, April 10, 2007.
- "From Concept to Silicon"
  Vason P. Srini, UC Berkeley, April 3, 2007.
- "SAT Sweeping with Local Observability Don't-Cares"
  Nathan Kitchen, UC Berkeley, March 20, 2007.
- "Selective Term-Level Abstraction Using Type-Inference"
  Bryan Brady, UC Berkeley, March 13, 2007.
- "Models for Data-Flow Sequential Processes"
  Mark B. Josephs, London South Bank University, March 13, 2007.
- "Control of Hybrid Systems: Theory, Computation and Applications"
  Manfred Morari, ETH Zurich, March 6, 2007.
- "Time-Portable Real-Time Programming with Exotasks"
  Christoph Kirsch, University of Salzburg, February 27, 2007.
- "High Level Mathematical Modeling and Parallel/GRID Computing with Modelica using OpenModelica"
  Peter Fritzson, Linköpings Universitet, February 20, 2007.
- "Discrete Event Models: Getting the Semantics Right"
  Edward A. Lee, UC Berkeley, February 6, 2007.
- "SCADA for distributed systems: application to the control of irrigation canals"
  Xavier Litrico, Cemagref, January 30, 2007.
- "Resource-Aware Programming"
  Walid Taha, Rice University, January 23, 2007.
- "A Model-Driven Approach to Embedded Control System Implementation"
  Jan F. Broenink, University of Twente, January 19, 2007.
- "The Power of Higher-Order Components in System Design"
  Adam Cataldo, UC Berkeley, December 12, 2006.

- "Some Results on Optimal Estimation and Control for Lossy Networked Control Systems"
  Luca Schenato, University of Padova, December 5, 2006.
- "Graphical System Design: Bringing Embedded Design to the Masses in Science and Engineering"
  Hugo A. Andrade and Zach Nelson, National Instruments, November 28, 2006.
- "Dynamical constrained Impulse system analysis through viability approach and applications"
  Patrick Saint-Pierre, University Paris Dauphine, November 21, 2006.
- "SHIM: A Scheduling-Independent Concurrent Language for Embedded Systems"
  Stephen A. Edwards, Columbia University, November 8, 2006.
- "Port-based Modeling and Control for Efficient Bipedal Walking Machines"
  Vincent Duindam, UC Berkeley, October 31 and November 7, 2006.
- "Using mathematical modeling to help decode biological circuits"
  Claire Tomlin, UC Berkeley & Stanford, October 17, 2006.
- "Control of Hybrid Systems: a Robust Finite State Machine Approach"
  Danielle C. Tarraf, Caltech, September 29, 2006.
- "Modeling with the Timing Definition Language (TDL)"
  Wolfgang Pree, University of Salzburg, September 26, 2006.
- "Advanced Visual Tracking with Bayesian Filter"
  Li-Chen Fu, NTU Taiwan, August 8, 2006

4.4.4. 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.

6th OOPSLA Workshop on Domain-Specific Modeling

During the OOPSLA Conference in October of 2006, Dr. Jonathan Sprinkle participated in, and helped to organize, the 6th 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, Vancouver 2004 and San Diego 2005). 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/

2007 International Symposium on Code Generation and Optimization (CGO)
March 11-14, 2007: Professor Edward A. Lee developed a position statement, "Are new languages necessary for multicore?" and presented it a panel by the same name at CGO.

Real-Time and Embedded Technology and Applications Symposium (RTAS)
April 3-6, 2007: Professor Edward A. Lee presented the keynote presentation, "Is Truly Real-Time Computing Becoming Unachievable?," at RTAS in Bellevue, WA on April 3-6, 2007.

6th ACM & IEEE Conference on Embedded Software (EMSOFT'06)
October 22-25, 2006: Professor Tom Henzinger was on the executive committee and chaired the advisory committee. Profs. Edward A. Lee, Alberto Sangiovanni-Vincentelli, and Janos Sztipanovits were members of the advisory committee.

4.4.5. 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.

4.5. Other Specific Product

The following software packages have been made available during this review period on the Chess website, http://chess.eecs.berkeley.edu:

- Metropolis consists of an infrastructure, a tool set, and design methodologies for various application domains. The infrastructure provides a mechanism such that heterogeneous components of a system can be represented uniformly and tools for formal methods can be applied naturally. The latest release, Metropolis 1.1.2 was made available on October 12, 2006 and may be found at: http://chess.eecs.berkeley.edu/chess/forum/17.html

- The Generic Modeling Environment (GME) 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, GME 6.11.9, was released on December 1, 2006 and may be found at: http://www.isis.vanderbilt.edu/projects/gme/.

- GReAT (Graph Rewriting And Transformation) is a component technology of GME comprised of a metamodel based graph transformation language useful for the specification and implementation of model-to-model transformations. The latest release,

GReAT 1.6.0, was released on December 1, 2006 and may be found at: http://www.escherinstitute.org/Plone/tools/suites/mic/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, UDM 3.1.1, was released on December 1, 2006 and may be found at: http://www.escherinstitute.org/Plone/tools/suites/mic/udm

- 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 1.1 was released on December 10, 2006 and is available at www.eecs.berkeley.edu/~akurzhan/ellipsoids

- 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 5.6 was released on January 15, 2007 and is available as part of Ptolemy or as a standalone download at http://ptolemy.eecs.berkeley.edu/java/ptplot

- Ptolemy II 6.0.2 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. The Ptolemy II 6.0.2 was released by UC Berkeley on February 4, 2007 and 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 type check 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 1.3.6 was released on February 5, 2007 and is available at http://cil.sourceforge.net.

- 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 1.0.2 was released on February 9, 2007 and is available at http://ptolemy.eecs.berkeley.edu/viptos/

- 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 0.9.5 was released on April 23, 2007 and is available at http://javasketch.sourceforge.net/

5. Contributions

This section summarizes the major contributions during this reporting period.

5.1. Within Discipline

5.1.1. 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 tool chains 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.

5.1.2. 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

5.1.3. 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 tool suite, 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.

5.1.4. 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.

5.2. 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.

5.3. 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.

5.4. 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.

5.5. 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.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-0225610

UNIVERSITY OF CALIFORNIA, BERKELEY

September 7, 2008

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2007 – May 31, 2008

Contents

1 Participants **3**
1.1 People . . . 3
1.2 Partner Organizations: . . . 3
1.3 Collaborators: . . . 3

2 Activities and Findings **5**
2.1 Project Activities . . . 5
2.1.1 ITR Events . . . 5
2.1.2 Hybrid Systems Theory . . . 6
2.1.3 Deep Compositionality . . . 6
2.1.4 Robust Hybrid Systems . . . 6
2.1.5 Hybrid Systems and Systems Biology . . . 7
2.2 ProjectFindings . . . 8

3 Outreach **24**
3.1 Project Training and Development . . . 24
3.2 Outreach Activities . . . 24
3.2.1 Curriculum Development for Modern Systems Science (MSS) . . . 24
3.2.2 Undergrad Course Insertion and Transfer . . . 25
3.2.3 Graduate Courses . . . 26

4 Publications and Products **28**
4.1 Technical reports . . . 28
4.2 Software . . . 29
4.3 PhD theses . . . 29
4.4 Conference papers . . . 29
4.5 Book chapters or sections . . . 31
4.6 Journal articles . . . 31
4.7 Dissemination . . . 31
4.7.1 The 2007-2008 Chess seminar series . . . 31
4.7.2 Workshops and Invited Talks . . . 34
4.7.3 General Dissemination . . . 34
4.8 Other Specific Products . . . 35

5 Contributions **35**
5.1 Within Discipline . . . 35
5.1.1 Hybrid Systems Theory . . . 35
5.1.2 Model-Based Design . . . 36
5.1.3 Advanced Tool Architectures . . . 37
5.1.4 Experimental Research . . . 37
5.2 Other Disciplines . . . 38
5.3 Human Resource Development . . . 38
5.4 Integration of Research and Education . . . 39

5.5 Beyond Science and Engineering 39

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)
CLAIRE TOMLIN (UC BERKELEY, EECS)
FACULTY INVESTIGATORS:
ALEXANDRE BAYEN (UC BERKELEY, CIVIL ENGINEERING)
POST DOCTORAL RESEARCHER:
JONATHAN SPRINKLE (SUMMER)[1] (UC BERKELEY)
GRADUATE STUDENTS:
ALESSANDRO ABATE (SUMMER) (UC BERKELEY, PROF. TOMLIN)
SAURABH AMIN (UC BERKELEY, PROF. SASTRY, PROF. BAYEN)
ANIL ASWANI (UC BERKELEY, PROF. TOMLIN)
ARINDAM CHAKRABARTI (UC BERKELEY, PROF. HENZINGER)
KRISHNENDU CHATTERJEE (SUMMER) (UC BERKELEY, PROF. HENZINGER)
ABHIJIT DAVARE (SUMMER) (UC BERKELEY, PROF. SANGIOVANNI-VINCENTELLI)
MILOS DREZGIC (UC BERKELEY, PROF. SASTRY)
ARKEDEB GHOSAL (SUMMER) (UC BERKELEY, PROF. SANGIOVANNI-VINCENTELLI)
SLOBODAN MATIC (UC BERKELEY, PROF. HENZINGER)
ALESSANDRO PINTO (SUMMER) (UC BERKELEY, PROF. SANGIOVANNI-VINCENTELLI)
VINAYAK PRABHU (UC BERKELEY, PROF. HENZINGER)
TECHNICAL STAFF, SYSTEMS ADMINISTRATION:
MARY P STEWART (UC BERKELEY)
BUSINESS ADMINISTRATOR:
TRACEY RICHARDS (UC BERKELEY)
EXECUTIVE DIRECTOR:
CHRISTOPHER BROOKS (UC BERKELEY)

1.2 Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY

1.3 Collaborators:

AARON AMES (CALTECH)
ANIL ASWANI (STANFORD UNIVERSITY)
JEFF AXELROD (STANFORD UNIVERSITY)

[1] RECEIVED FUNDING ONLY DURING THE SUMMER

DIRK BEYER (SIMON FRASER UNIVERSITY)
THOMAS BRIHAYE (UNIVERSITE DE MONS-HAINAUT)
LUCA CARLONI (COLUMBIA UNIVERSITY)
ALESSANDRO D'INNOCENZO (UNIVERSITY OF PENNSYLVANIA)
MASSIMILIANO D'ANGELO (UNIVERSITY OF L'AQUILA AND PARADES GEIE)
LUCA DE ALFARO (UNIVERSITY OF CALIFORNIA, SANTA CRUZ)
DOUGLAS DENSMORE (UNIVERSITY OF CALIFORNIA, BERKELEY)
MARIKA DI BENEDETTO (UNIVERSITY OF L'AQUILA)
MARCO DI NATALE (SCUOLA SUPERIORE SANT'ANNA)
LAURENT EL GHAOUI (UNIVERSITY OF CALIFORNIA, BERKELEY)
CARLO FISCHIONE (UNIVERSITY OF CALIFORNIA, BERKELEY)
ANIRUDDA GOKHALE (VANDERBILT UNIVERSITY)
JEFF GRAY (VANDERBILT UNIVERSITY)
FALK HANTE (UNIVERSITY OF ERLANG)
DANIEL IERCAN (UNIVERSITY OF SALZBURG)
MARCIN JURDZINSKI (UNIVERSITY OF CALIFORNIA, BERKELEY)
ANDREW B. KAHNG (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
SRI KANAJAN (GENERAL MOTORS)
STEVEN KELLY (VANDERBILT)
CHRISTOPH KIRSCH (UNIVERSITY OF SALZBURG)
DOMINIK LANGEN (INFINEON)
JIE LIU (MICROSOFT RESEARCH
JOHN LYGEROS (ETH ZURICH)
FREDDY MANG (UNIVERSITY OF CALIFORNIA, BERKELEY)
RUPAK MAJUMDAR (UNIVERSITY OF CALIFORNIA, LOS ANGELES)
SWAMY MUDDU (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
CLAUDIO PINELLO (CADENCE DESIGN SYSTEMS)
G. POLA (UNIVERSITY OF L'AQUILA)
MARIA PRANDINI (MILANO)
JEAN-FRANCOIS RASKIN (UNIVERSITE LIBRE DE BRUXELLES)
MIRKO SAUERMANN (INFINEON)
KAMBIZ SAMADI (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
EELCO SCHOLTE (UNITED TECHNOLOGIES RESEARCH CENTER)
KOUSHIK SEN (UNIVERSITY OF CALIFORNIA, BERKELEY)
PUNEET SHARMA (UNIVERSITY OF CALIFORNIA, SAN DIEGO)
ASHISH TIWARI (SRI INTERNATIONAL)
JUHA-PEKKA TOLVANEN (VANDERBILT UNIVERSITY)
RANDALL URBANCE (GENERAL MOTORS)
QI ZHU (UNIVERSITY OF CALIFORNIA, BERKELEY)

2 Activities and Findings

2.1 Project Activities

This is the sixth Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software." This year was a no-cost extension for certain researchers at the University of California, Berkeley (Center for Hybrid and Embedded Systems and Software (CHESS), http://chess.eecs.berkeley.edu. Research at the other CHESS partners: 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 ended before the period covered by this report.
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.
The CHESS ITR grant has been instrumental in supporting the launch of Tomlin's new Hybrid Systems Laboratory in Cory Hall. Specifically, the grant continues to support several new directions in systems biology, centered on the development of hybrid systems models and analysis tools for the analysis and deeper understanding of several protein regulatory networks. The grant has supported Tomlin, her PhD student Anil Aswani, and a Berkeley undergraduate, Nicholas Boyd. Two additional Berkeley undergraduates, Harendra Guturu and Eugene Li, have worked on the project though have been supported by external fellowships. The research experience obtained by these undergraduates has been instrumental in helping them decide their next steps: Guturu was accepted and is currently starting the PhD program in Electrical Engineering at Stanford, and Li has been accepted into the 5th year Masters program at Berkeley and will continue working on the project this year and next. Boyd will continue working on the project as an undergraduate this year.

2.1.1 ITR Events

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

- Workshop: From Embedded Systems to Cyber-Physical Systems: a Review of the State-of-the-Art and Research Needs, April 21, 2008, St. Louis, MO

  The CPS Workshop was held in conjunction with RTAS and sponsored in part by the European Community Artist Network of Excellence and COMBEST STREP.

  The theme of the workshop was presenting an overarching view of methodologies and theories for the design of embedded and critical systems as it has emerged in the past five years and discussing the future in terms of the extension of the notion of embedded systems to Cyber-Physical Systems (CPS). In the overview of the present status of the discipline, the workshop will address heterogeneous system composition, design methods based on abstraction and refinement, interface theories, mapping of abstract entities to implementation platforms and industrial applications. The presentations will also feature industry representatives who will give their perspective of what are the gaping holes in the state of the art in their business segment and how to bridge academic accomplishments with industrial practice. The discussion about the extension of the theories and methodologies to the new generation of CPS will review the necessary steps

and a possible roadmap for research. The discussion will also include public research organizations. European Community representatives will provide the state-of-the-art and the research initiatives on embedded systems in the EU.

The program and presentations are available at
http://chess.eecs.berkeley.edu/conferences/08/StLouis/index.htm

- The Sydney-Berkeley Driving Team participated in the DARPA Grand Challenge [1] For details, see
http://chess.eecs.berkeley.edu/dgc3

- 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://www.eecs.berkeley.edu/BEARS/2008/index.html

- A weekly Chess seminar was held at Berkeley. The speakers and topics are listed in Section 4.7.1, presentations for the seminar 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. As year six was a no-cost extension, we include only thrust areas funded by the no-cost extension.

2.1.2 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.
During the period covered by this report, Professor Henzinger's group made the following advancements:

1. New algorithms and complexity results for the verification and control of probabilistic systems (which are modeled as stochastic games). [2], [3], [4], [5]

2. New algorithms for the verification and control of real-time systems (which are modeled as timed games). [6], [7], [8], [9]

3. New algorithm for control under budget constraints. [10]

2.1.3 Deep Compositionality

Professor Henzinger's group developed CHIC, a checker for interface compatibility with applications to web services. [11], [12], [13],

2.1.4 Robust Hybrid Systems

Professor Henzinger's group developed a hierarchical coordination language for real-time tasks extended with reliability constraints (in the Giotto tradition). [14]

2.1.5 Hybrid Systems and Systems Biology

The CHESS ITR has enabled a new collaboration, between Tomlin's group and a group of developmental biologists at Lawrence Berkeley Labs and the Department of Molecular and Cell Biology at Berkeley. This group, led by Dr. Mark Biggin and Professor Mike Eisen, are studying the early Drosophila development. They have developed state of the art tools for RNA and protein data collection, and have collaborated with computer vision researchers to develop a "virtual embryo", visualizing all data at once on a 3D representation of the Drosophila embryo. We have begun a collaboration with their group to design dynamic models of this system: modeling RNA and protein concentrations to try to uncover the detailed interactions between these gene products that are key in fly development. We are developing continuous and hybrid models to represent the dynamics of this system.

Early patterning in the Drosophila melanogaster embryo occurs through a complicated network of interactions involving proteins and mRNA. One such system is the pattern of hunchback mRNA in the presence of Bicoid and Kruppel protein. This system is well-studied, but there is disagreement amongst biologists between two general models. Our aim is to provide evidence to support one of the two models in contention, and we do this through system identification methods. Our general approach is to do nonlinear regression on a parametric, nonlinear partial differential equation model which incorporates transcription, diffusion, and degradation. We perform the nonlinear regression and analyze the results of the nonlinear regression. We interpret the results in the biological context, and we also compare our results to previous work on this system.

In terms of hybrid model development, we have focused in particular on the relationship between a particular class of hybrid systems, known as piecewise affine (PWA) systems, and monotone systems (which have certain properties making them amenable to stability analysis). Monotone systems are order-preserving systems: given a partial order on any two initial conditions, the trajectories of the monotone system preserve this partial order through time. There is a rich theory of strong results about the dynamics and stability of monotone systems with continuous vector fields. These existing results do not apply to piecewise affine (PWA) systems, which have discontinuous vector fields. Though the previous work on monotone systems has largely been theoretical, there is growing interest in monotone systems due to the realization that many systems in biology are monotone. Our work considers the relationship between monotone and PWA systems, which have found applications in biology. Understanding which conditions are sufficient for a PWA system to be monotone is useful, both for understanding the dynamics as well as for designing controllers. In our work, we characterize monotonicity of PWA systems. Then, we prove analogs of the Kamke-Muller and related graph theoretical theorems, both of which provide sufficient conditions for a system with continuous vector field to be monotone. Our analogs give sufficient conditions for a PWA system to be a monotone system.

More generally, we have been studying the topology of graphs representing biological influence models, and investigating the development of a corresponding "control theory" for these graphs. The traditional control scheme has been to input a signal into a plant, where the signal is derived from either an open-loop or a closed-loop. This control strategy requires that the plant be able to accept inputs or can be modified to do so. However, this situation is not always true in biological genetic networks; in these systems, there is often no input or obvious modification to allow inputs. We believe that they require a new paradigm for control. Biotechnology techniques are such that it is easier to make topological changes to a

genetic network than it is to either change the states of the pathway or add more elements to the pathway. Thus, for such genetic networks it is important to develop a theory of control based on making large scale changes (e.g. genetic mutations) to the topology of the network; we provide steps towards such a theory. We highlight some useful results from monotone and hybrid systems theory, and show how these results can be used for such a topological control scheme. We consider the cancer-related p53 pathway as an example; we analyze this system using control theory and devise a controller.

2.2 ProjectFindings

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 4 below, including publications representing findings that were reported in the previous annual report.

- [1]Ben Upcroft, Michael Moser, Alex Makarenko, David Johnson, Ashod Donikian, Alen Alempijevic, Robert Fitch, Will Uther, Esten Ingar Grtli, Jan Biermeyer, Humberto Gonzalez, Todd Templeton, Vason P. srini, Jonathan Sprinkle. Technical report, "DARPA Urban Challenge Technical Paper: Sydney-Berkeley Driving Team," University of Sydney; University of Technology, Sydney; University of California, Berkeley, June, 2007.

  The Sydney-Berkeley Driving Team is a collaboration between academic and research personnel from (in alphabetical order) the National Information and Communication Technology of Australia, University of California, Berkeley, University of Sydney, and the University of Technology, Sydney. This document describes the planning, actuation, simulation, communication, theoretical tasks, advancements, and projections necessary for the team to compete in the DARPA Urban Challenge. Among our major accomplishments, we claim the ability for distributed code development through the use of our component-based middleware, a high-confidence testbed which was designed and implemented from the ground up by our engineers, prototype testing in months, and robust software design and development allowing a seamless transition between simulation and online testing.

- [15]Abhijit Davare, Qi Zhu, Marco Di Natale, Claudio Pinello, Sri Kanajan, Alberto Sangiovanni-Vincentelli. "Period Optimization for Hard Real-time Distributed Automotive Systems," Design Automation Conference, 278-283, June, 2007.

  The complexity and physical distribution of modern active-safety automotive applications requires the use of distributed architectures. These architectures consist of multiple electronic control units (ECUs) connected with standardized buses. The most common configuration features periodic activation of tasks and messages coupled with run-time priority-based scheduling. The correct deployment of applications on such architectures requires end-to end latency deadlines to be met. This is challenging since deadlines must be enforced across a set of ECUs and buses, each of which supports multiple functionality. The need for accommodating legacy tasks and messages further complicates the scenario. In this work, we automatically assign task and message periods for distributed automotive systems. This is accomplished by leveraging schedulability analysis within a convex optimization framework to simultaneously assign periods and satisfy end-to-end latency constraints. Our approach is applied to an industrial case study as well as an example taken from the literature and is shown to be both effective and efficient.

- [16]Trevor Meyerowitz. PhD thesis, "Single and Multi-CPU Performance Modeling for Embedded Systems," University of California at Berkeley, April, 2008.

  The combination of increasing design complexity, increasing concurrency, growing heterogeneity, and decreasing time to market windows has caused a crisis for embedded system developers. To deal with this problem, dedicated hardware is being replaced

by a growing number of microprocessors in these systems, making software a dominant factor in design time and cost. The use of higher level models for design space exploration and early software development is critical. Much progress has been made on increasing the speed of cycle-level simulators for microprocessors, but they may still be too slow for large scale systems and are too low-level (i.e. they require a detailed implementation) for effective design space exploration. Furthermore, constructing such optimized simulators is a significant task because the particularities of the hardware must be accounted for. For this reason, these simulators are hardly flexible. This thesis focuses on modeling the performance of software executing on embedded processors in the context of a heterogeneous multi-processor system on chip in a more flexible and scalable manner than current approaches. We contend that such systems need to be modeled at a higher level of abstraction and, to ensure accuracy, the higher level must have a connection to lower-levels. First, we describe different levels of abstraction for modeling such systems and how their speed and accuracy relate. Next, the high-level modeling of both individual processing elements and also a bus-based microprocessor system are presented. Finally, an approach for automatically annotating timing information obtained from a cycle-level model back to the original application source code is developed. The annotated source code can then be simulated without the underlying architecture and still maintain good timing accuracy. These methods are driven by execution traces produced by lower level models and were developed for ARM microprocessors and MuSIC, a heterogeneous multiprocessor for Software Defined Radio from Infineon. The annotated source code executed between one to three orders of magnitude faster than equivalent cycle-level models, with good accuracy for most applications tested.

- [17]Trevor Meyerowitz, Dominik Langen, Mirko Sauermann, Alberto Sangiovanni-Vincentelli. "Source-Level Timing Annotation and Simulation for a Heterogeneous Multiprocessor," Design Automation Test Europe, IEEE, March, 2008.

  A generic and retargetable tool flow is presented that enables the export of timing data from software running on a cycle-accurate Virtual Prototype (VP) to a concurrent functional simulator. First, an annotation framework takes information gathered from running an application on the VP and automatically annotates the line-level delays back to the original source code. Then, a SystemC-based timed functional simulator runs the annotated source code much faster than the VP while preserving timing accuracy. This simulator is API-compatible with the multiprocessor's operating system. Therefore, it can compile and run unmodified applications on the host PC. This flow has been implemented for MuSIC(Multiple SIMD Cores), a heterogeneous multiprocessor developed at Infineon to support Software Defined Radio (SDR). When compared with an optimized cycle-accurate VP of MuSIC on a variety of tests, including a multiprocessor JPEG encoder, the accuracy is within 20%, with speedups from 10x to 1000x.

- [18]Ethan Jackson. Technical report, "The Software Engineering of Domain-Specific Modeling Languages: A Survey Through Examples," Institute For Software Integrated Systems (ISIS), ISIS-07-807, March, 2008.

  This paper presents the fundamental concepts of model-based design to the broader software engineering community. We examine model-based design from the perspective

of domain-specific modeling languages (DSMLs). DSMLs capture the structure, behavioral characteristics, and abstractions of complex problem domains. Model transformations defined between language syntaxes serve as high-level specifications of domain-specific compilers. Additionally, transformations are used to change abstraction levels. This paper is example driven and includes examples from a number of tools including ASML [1], Ptolemy II [2], GME [3], and GReAT [4].

- [19]Krishnendu Chatterjee, Tom Henzinger, Daniel Iercan, Christoph Kirsch, Claudio Pinello, Alberto Sangiovanni-Vincentelli. "Logical Reliability of Interacting Real-Time Tasks," Design, Automation and Test in Europe, 2008. DATE '08, 909-914, March, 2008.

  We propose the notion of logical reliability for real-time program tasks that interact through periodically updated program variables. We describe a reliability analysis that checks if the given short-term (e.g., single-period) reliability of a program variable update in an implementation is sufficient to meet the logical reliability requirement (of the program variable) in the long run. We then present a notion of design by refinement where a task can be refined by another task that writes to program variables with less logical reliability. The resulting analysis can be combined with an incremental schedulability analysis for interacting real-time tasks proposed earlier for the Hierarchical Timing Language (HTL), a coordination language for distributed real-time systems. We implemented a logical-reliability-enhanced prototype of the compiler and runtime infrastructure for HTL.

- [2]Krishnendu Chatterjee, Tom Henzinger, Koushik Sen. "Model-Checking omega-Regular Properties of Interval Markov Chains," Foundations of Software Science and Computation Structure (FoSSaCS) 2008, Roberto M. Amadio (ed.), 302-317, March, 2008.

  We study the problem of model checking Interval-valued Discrete-time Markov Chains (IDTMC). IDTMCs are discrete-time finite Markov Chains for which the exact transition probabilities are not known. Instead in IDTMCs, each transition is associated with an interval in which the actual transition probability must lie. We consider two semantic interpretations for the uncertainty in the transition probabilities of an IDTMC. In the first interpretation, we think of an IDTMC as representing a (possibly uncountable) family of (classical) discrete-time Markov Chains, where each member of the family is a Markov Chain whose transition probabilities lie within the interval range given in the IDTMC. We call this semantic interpretation Uncertain Markov Chains (UMC). In the second semantics for an IDTMC, which we call Interval Markov Decision Process (IMDP), we view the uncertainty as being resolved through non-determinism. In other words, each time a state is visited, we adversarially pick a transition distribution that respects the interval constraints, and take a probabilistic step according to the chosen distribution. We introduce a logic omega-PCTL that can express liveness, strong fairness, and omega-regular properties (such properties cannot be expressed in PCTL). We show that the omega-PCTL model checking problem for Uncertain Markov Chain semantics is decidable in PSPACE (same as the best known upper bound for PCTL) and for Interval Markov Decision Process semantics is decidable in coNP (improving the previous known PSPACE bound for PCTL). We also show that the qualitative fragment of the logic can be solved in coNP for the UMC interpretation, and can be

solved in polynomial time for a sub-class of UMCs. We also prove lower bounds for these model checking problems.We show that the model checking problem of IDTMCs with LTL formulas can be solved for both UMC and IMDP semantics by reduction to the model checking problem of IDTMC with omega-PCTL formulas.

- [20]Douglas Densmore, Trevor Meyerowitz, Abhijit Davare, Qi Zhu, Guang Yang. Technical report, "Metro II Execution Semantics for Mapping," University of California, Berkeley, UCB/EECS-2008-16, February, 2008.

  This document presents three proposals for the execution semantics of mapping in Metro II. Mapping is the relationship between what a system does (functionality) and how it does it (architecture). The main concern is whether the functionality and architecture models should execute concurrently or sequentially during simulation. Proposal #1 presents sequential execution with the functionality being executed before the architecture. Proposal #2 also presents sequential execution, but with the architecture executing before the functionality. Finally, Proposal #3 presents concurrent execution. Processes are present in the architecture to execute simultaneously with the events mapped to them in the functionality. Each of these three proposals is demonstrated on a set of design scenarios with hand traces illustrating their execution. Additionally general assumptions, glossary terms, and proposal-specific assumptions made regarding the execution semantics are discussed. Finally, the proposals are compared and contrasted, especially regarding how they can properly implement the examples and the general semantic assumptions.

- [14]Arkadeb Ghosal. PhD thesis, "A Hierarchical Coordination Language for Reliable Real-Time Tasks," EECS Department, University of California, Berkeley, January, 2008.

  Complex requirements, time-to-market pressure and regulatory constraints have made the designing of embedded systems extremely challenging. This is evident by the increase in effort and expenditure for design of safety-driven real-time control dominated applications like automotive and avionic controllers. Design processes are often challenged by lack of proper programming tools for specifying and verifying critical requirements (e.g. timing and reliability) of such applications. Platform based design, an approach for designing embedded systems, addresses the above concerns by separating requirement from architecture. The requirement specifies the intended behavior of an application while the architecture specifies the guarantees (e.g. execution speed, failure rate etc). An implementation, a mapping of the requirement on the architecture, is then analyzed for correctness. The orthogonalization of concerns makes the specification and analyses simpler. An effective use of such design methodology has been proposed in Logical Execution Time (LET) model of real-time tasks. The model separates the timing requirements (specified by release and termination instances of a task) from the architecture guarantees (specified by worst-case execution time of the task). This dissertation proposes a coordination language, Hierarchical Timing Language (HTL), that captures the timing and reliability requirements of real-time applications. An implementation of the program on an architecture is then analyzed to check whether desired timing and reliability requirements are met or not. The core framework extends the LET model by accounting for reliability and refinement. The reliability model separates the reliability requirements of tasks from the reliability guar-

antees of the architecture. The requirement expresses the desired long-term reliability while the architecture provides a short-term reliability guarantee (e.g. failure rate for each iteration). The analysis checks if the short-term guarantee ensures the desired long-term reliability. The refinement model allows replacing a task by another task during program execution. Refinement preserves schedulability and reliability, i.e., if a refined task is schedulable and reliable for an implementation, then the refining task is also schedulable and reliable for the implementation. Refinement helps in concise specification without overloading analysis. The work presents the formal model, the analyses (both with and without refinement), and a compiler for HTL programs. The compiler checks composition and refinement constraints, performs schedulability and reliability analyses, and generates code for implementation of an HTL program on a virtual machine. Three real-time controllers, one each from automatic control, automotive control and avionic control, are used to illustrate the steps in modeling and analyzing HTL programs. Advisor: Alberto L. Sangiovanni-Vincentelli and Thomas A. Henzinger

- [6]Krishnendu Chatterjee, Tom Henzinger, Vinayak Prabhu. Technical report, "Trading Infinite Memory for Uniform Randomness in Timed Games," EECS Department University of California, Berkeley, UCB/EECS-2008-4, January, 2008.

  We consider concurrent two-player timed automaton games with omega-regular objectives specified as parity conditions. These games offer an appropriate model for the synthesis of real-time controllers. Earlier works on timed games focused on pure strategies for each player. We study, for the first time, the use of randomized strategies in such games. While pure (i.e., nonrandomized) strategies in timed games require infinite memory for winning even with respect to reachability objectives, we show that randomized strategies can win with finite memory with respect to all parity objectives. Also, the synthesized randomized real-time controllers are much simpler in structure than the corresponding pure controllers, and therefore easier to implement. For safety objectives we prove the existence of pure finite-memory winning strategies. Finally, while randomization helps in simplifying the strategies required for winning timed parity games, we prove that randomization does not help in winning at more states.

- [21]Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Markov Set-Chains as Abstractions of Stochastic Hybrid Systems," Springer Verlag, 2008; Chapter to appear in "Hybrid Systems: Computation and Control", 2008.

  The objective of this study is to introduce an abstraction procedure that applies to a general class of dynamical systems, that is to discrete-time stochastic hybrid systems (dt-SHS). The procedure abstracts the original dt-SHS into a Markov set-chain (MSC) in two steps. First, a Markov chain (MC) is obtained by partitioning the hybrid state space, according to a controllable parameter, into non-overlapping domains and computing transition probabilities for these domains according to the dynamics of the dt-SHS. Second, explicit error bounds for the abstraction that depend on the above parameter are derived, and are associated to the computed transition probabilities of the MC, thus obtaining a MSC. We show that one can arbitrarily increase the accuracy of the abstraction by tuning the controllable parameter, albeit at an increase of the cardinality of the MSC. Resorting to a number of results from the MSC literature allows

the analysis of the dynamics of the original dt-SHS. In the present work, the asymptotic behavior of the dt-SHS dynamics is assessed within the abstracted framework.

- [10]Krishnendu Chatterjee, Tom Henzinger, Rupak Majumdar. "Controller Synthesis with Budget Constraints," HSCC 2008, 2008.

  We study the controller synthesis problem under budget constraints. In this problem, there is a cost associated with making an observation, and a controller can make only a limited number of observations in each round so that the total cost of the observations does not exceed a given fixed budget. The controller must ensure some omega-regular requirement subject to the budget constraint. Budget constraints arise in designing and implementing controllers for resource-constrained embedded systems, where a controller may not have enough power, time, or bandwidth to obtain data from all sensors in each round. They lead to games of imperfect information, where the unknown information is not fixed a priori, but can vary from round to round, based on the choices made by the controller how to allocate its budget. We show that the budget-constrained synthesis problem for omega-regular objectives is complete for exponential time. In addition to studying synthesis under a fixed budget constraint, we study the budget optimization problem, where given a plant, an objective, and observation costs, we have to find a controller that achieves the objective with minimal average accumulated cost (or minimal peak cost). We show that this problem is reducible to a game of imperfect information where the winning objective is a conjunction of an omega-regular condition and a long-run average condition (or a least max-cost condition), and this again leads to an exponential-time algorithm. Finally, we extend our results to games over infinite state spaces, and show that the budget-constrained synthesis problem is decidable for infinite state games with stable quotients of finite index. Consequently, the discrete time budget-constrained synthesis problem is decidable for rectangular hybrid automata.

- [22]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Approximation of General Stochastic Hybrid Systems by Switching Diffusions with Random Hybrid Jumps," Springer Verlag, 2008; Chapter to appear in "Hybrid Systems: Computation and Control," 2008 .

  In this work we propose an approximation scheme to transform a general stochastic hybrid system (SHS) into a SHS without forced transitions due to spatial guards. Such switching mechanisms are replaced by spontaneous transitions with state-dependent transition intensities (jump rates). The resulting switching diffusion process with random hybrid jumps is shown to converge in distribution to the original stochastic hybrid system execution. The obtained approximation can be useful for various purposes such as, on the computational side, simulation and reachability analysis, as well as for the theoretical investigation of the model. More generally, it is suggested that SHS which are endowed exclusively with random jumping events are simpler than those that present spatial forcing transitions. In the opening of this work, the general SHS model is presented, a few of its basic properties are discussed, and the concept of generator is introduced. The second part of the paper describes the approximation procedure, introduces the new SHS model, and proves, under some assumptions, its weak convergence to the original system.

- [7]Tom Henzinger, Krishnendu Chatterjee, Vinayak Prabhu. "Timed Parity Games:

Complexity and Robustness," FORMATS: Formal Modeling and Analysis of Timed Systems, 2008; To appear.

We consider two-player games played in real time on game structures with clocks and parity objectives. The games are concurrent in that at each turn, both players independently propose a time delay and an action, and the action with the shorter delay is chosen. To prevent a player from winning by blocking time, we restrict each player to strategies that ensure that the player cannot be responsible for causing a zeno run. First, we present an efficient reduction of these games to turn-based (i.e., nonconcurrent) finite-state (i.e., untimed) parity games. The states of the resulting game are pairs of clock regions of the original game. Our reduction improves the best known complexity for solving timed parity games. Moreover, the rich class of algorithms for classical parity games can now be applied to timed parity games. Second, we consider two restricted classes of strategies for the player that represents the controller in a real-time synthesis problem, namely, limit-robust and bounded-robust strategies. Using a limit-robust strategy, the controller cannot choose an exact real-valued time delay but must allow for some nonzero jitter in each of its actions. If there is a given lower bound on the jitter, then the strategy is bounded-robust. We show that exact strategies are more powerful than limit-robust strategies, which are more powerful than bounded-robust strategies for any bound. For both kinds of robust strategies, we present efficient reductions to standard timed automaton games. These reductions provide algorithms for the synthesis of robust real-time controllers.

- [8]Krishnendu Chatterjee, Tom Henzinger, Vinayak Prabhu. "Trading Infinite Memory for Uniform Randomness in Timed Games," HSCC: Hybrid Systems – Computation and Control, 2008.

  We consider concurrent two-player timed automaton games with omega-regular objectives specified as parity conditions. These games offer an appropriate model for the synthesis of real-time controllers. Earlier works on timed games focused on pure strategies for each player. We study, for the first time, the use of randomized strategies in such games. While pure (i.e., nonrandomized) strategies in timed games require infinite memory for winning even with respect to reachability objectives, we show that randomized strategies can win with finite memory with respect to all parity objectives. Also, the synthesized randomized real-time controllers are much simpler in structure than the corresponding pure controllers, and therefore easier to implement. For safety objectives we prove the existence of pure finite-memory winning strategies. Finally, while randomization helps in simplifying the strategies required for winning timed parity games, we prove that randomization does not help in winning at more states.

- [23]Saurabh Amin, Falk Hante, Alexandre Bayen. Technical report, "Exponential stability of switched hyperbolic systems in a bounded domain," UC Berkeley, 2008.

  We consider switching in time among a finite family of systems governed by linear hyperbolic partial differential equations on a bounded space interval. The switching system is fairly general in that the space dependent system matrix functions as well as the boundary conditions may switch in time. For the case in which the switching occurs between hyperbolic systems in the canonical diagonal form, we provide two sets of sufficient conditions for the switched system to be exponentially stable under arbitrary switching signals. These results are generalizations of the corresponding results for

the un-switched case. Furthermore, we provide an explicit dwell-time bound on the switching signals that guarantee exponential stability of the switched system under the assumption that each of the individual systems are stable. Our results of stability under arbitrary switching generalize to the case in which switching occurs between non-diagonal hyperbolic systems that are diagonalizable using a common transformation. For the case in which no such transformation exists, we prove existence of a dwell-time bound on the switching signals such that exponential stability is guaranteed.

- [24]Anil Aswani, Claire Tomlin. IEEE TAC, "Monotone Piecewise Affine Systems," 2008; Submitted, to appear in 2009.

  (No abstract.)

- [25]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. "Neuro-Dynamic Programming for Probabilistic Reachability of Stochastic Hybrid Systems," Submitted, 2008.

  (No abstract.)

- [26]Anil Aswani, Claire Tomlin. "Topology Based Control of Biological Genetic Networks," CDC, 2008; Submitted.

  The traditional controller scheme has been to input a signal into a plant, where the signal is derived from either an open-loop or a closed-loop. This control strategy requires that our plant is able to accept inputs or can be modified to do so. However, this situation is not always true in biological genetic networks; in these systems, there is often no input or obvious modification to allow inputs. Many genetic networks are different, and we believe that they require a new paradigm for control. Biotechnology techniques are such that it is easier to make topological changes to a genetic network than it is to either change the states of the pathway or add more elements to the pathway (i.e. changing the "circuit"). Thus, for such genetic networks it is important to develop a theory of control based on making large-scale changes (e.g. genetic mutations) to the topology of the network. We highlight some useful results from monotone and hybrid systems theory, and show how these results can be used for such a topological controller scheme. We consider the cancer-related, p53 pathway as an example. We analyze the system using control theory and devise a controller.

- [27]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. Automatica, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," 2008; To appear.

  In this work, probabilistic reachability over a finite horizon is investigated for a class of discrete time stochastic hybrid systems with control inputs. A suitable embedding of the reachability problem in a stochastic control framework reveals that it is amenable to two complementary interpretations, leading to dual algorithms for reachability computations. In particular, the set of initial conditions providing a certain probabilistic guarantee that the system will keep evolving within a desired 'safe' region of the state space is characterized in terms of a value function, and 'maximally safe' Markov policies are determined via dynamic programming. These results are of interest not only for safety analysis and design, but also for solving those regulation and stabilization problems that can be reinterpreted as safety problems. The temperature regulation problem presented in the paper as case study is one such case.

- [28]Alessandro DInnocenzo, Alessandro Abate, Maria D. Di Benedetto, S. Shankar Sastry. "Approximate Abstractions of Discrete-Time Controlled Stochastic Hybrid Systems," Submitted, 2008.

  (No abstract.)

- [29]Saurabh Amin, Falk Hante, Alexandre Bayen. Magnus Egerstedt and Bud Mishra, (eds.), "On stability of switched linear hyperbolic conservation laws with reflecting boundaries," 602-605, Hybrid Systems: Comp, Springer-Verlag, 2008.

  We consider stability of an infinite dimensional switching system, posed as a system of linear hyperbolic partial differential equations (PDEs) with reflecting boundaries, where the system parameters and the boundary conditions switch in time. Asymptotic stability of the solution for arbitrary switching is proved under commutativity of the advective velocity matrices and a joint spectral radius condition involving the boundary data.

- [11]Dirk Beyer, Arindam Chakrabarti, Krishnendu Chatterjee, Luca de Alfaro, Tom Henzinger, Marcin Jurdzinski, Freddy Mang, Cindy Song. "CHIC: Checking Interface Compatibility," UC Berkeley, November, 2007.

  CHIC is a modular verifier for behavioral compatibility checking of software and hardware components. The goal of CHIC is to be able to check that the interfaces for software or hardware components provide guarantees that satisfy the assumptions they make about each other. CHIC supports a variety of interface property specification formalisms: synchronous assume/guarantee interfaces, resource interfaces, web service interfaces, etc.

- [3]Krishnendu Chatterjee. "Stochastic Muller Games are PSPACE-complete," FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 436-448, December, 2007.

  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); and the quantitative problem asks for the maximal probability of winning (optimal winning) from each state. We consider omega-regular winning conditions formalized as Muller winning conditions. We present optimal memory bounds for pure (deterministic) almost-sure winning and optimal winning strategies in stochastic graph games with Muller winning conditions. We also present improved memory bounds for randomized almost-sure winning and optimal strategies. We study the complexity of stochastic Muller games and show that the quantitative analysis problem is PSPACE-complete. Our results are relevant in synthesis of stochastic reactive processes.

- [12]Arindam Chakrabarti. PhD thesis, "A Framework for Compositional Design and Analysis of Systems," UC Berkeley, December, 2007.

Complex system design today calls for compositional design and implementation. However each component is designed with certain assumptions about the environment it is meant to operate in, and delivering certain guarantees if those assumptions are satisfied; numerous inter-component interaction errors are introduced in the manual and error-prone integration process as there is little support in design environments for machine-readably representing these assumptions and guarantees and automatically checking consistency during integration. Based on Interface Automata we propose a framework for compositional design and analysis of systems: a set of domain-specific automata-theoretic type systems for compositional system specification and analysis by behavioral specification of open systems. We focus on three different domains: component-based hardware systems communicating on bidirectional wires. concurrent distributed recursive message-passing software systems, and embedded software system components operating in resource-constrained environments. For these domains we present approaches to formally represent the assumptions and conditional guarantees between interacting open system components. Composition of such components produces new components with the appropriate assumptions and guarantees. We check satisfaction of temporal logic specifications by such components, and the substitutability of one component with another in an arbitrary context. Using this framework one can analyze large systems incrementally without needing extensive summary information to close the system at each stage. Furthermore, we focus only on the inter-component interaction behavior without dealing with the full implementation details of each component. Many of the merits of automata-theoretic model-checking are combined with the compositionality afforded by type-system based techniques. We also present an integer-based extension of the conventional boolean verification framework motivated by our interface formalism for embedded software components. Our algorithms for checking the behavioral compatibility of component interfaces are available in our tool Chic, which can be used as a plug-in for the Java IDE JBuilder and the heterogenous modeling and design environment Ptolemy II. Finally, we address the complementary problem of partitioning a large system into meaningful coherent components by analyzing the interaction patterns between its basic elements. We demonstrate the usefulness of our partitioning approach by evaluating its efficacy in improving unit-test branch coverage for a large software system implemented in C.

- [30]Saurabh Amin, Alexandre Bayen, Laurent El Ghaoui, S. Shankar Sastry. "Robust feasibility for control of water flow in a canal reservoir system," Decision and Control, 2007 46th IEEE Conference on, 1571-1577, December, 2007.

  A robust control problem for distant downstream control of a reservoir-canal system modeled by Saint-Venant equations is investigated. The problem is to regulate the release of water at the upstream end such that the measured water level (or stage) at the downstream end does not deviate outside of prescribed bounds under the effect of downstream perturbations. Under the assumption of small perturbations, the Saint-Venant model is linearized around a steady state flow. The resulting linear model is discretized to obtain a linear state-space model using a method of characteristics based numerical scheme. For the state space model, the control is the upstream discharge deviation, the disturbance is the downstream discharge deviation and the output is the downstream stage deviation; the deviations are defined with respect to the steady state. The sets of admissible control, disturbance and output trajectories are modeled

by polytopes. It is shown that the control problem can be formulated as a robust feasibility problem. Using linear programming duality, conditions for existence of a robustly feasible solution are derived. These conditions, being affine in the control variables, are checked using linear programming. The proposed method is applied to control a typical reservoir- canal system.

- [31]Aaron Ames, Alessandro Abate, S. Shankar Sastry. "Sufficient Conditions for the Existence of Zeno Behavior in Nonlinear Hybrid Systems via Constant Approximations," 46th IEEE Conference on Decision and Control and European Control, 4033-4038, December, 2007.

  The existence of Zeno behavior in hybrid systems is related to a certain type of equilibria, termed Zeno equilibria, that are invariant under the discrete, but not the continuous, dynamics of a hybrid system. In analogy to the standard procedure of linearizing a vector field at an equilibrium point to determine its stability, in this paper we study the local behavior of a hybrid system near a Zeno equilibrium point by considering the value of the vector field on each domain at this point, i.e., we consider constant approximations of nonlinear hybrid systems. By means of these constant approximations, we are able to derive conditions that simultaneously imply both the existence of Zeno behavior and the local exponential stability of a Zeno equilibrium point. Moreover, since these conditions are in terms of the value of the vector field on each domain at a point, they are remarkably easy to verify.

- [32]Alessandro Abate, Ashish Tiwari, S. Shankar Sastry. "The concept of Box Invariance for biologically-inspired dynamical systems," 46th IEEE Conference on Decision and Control and European Control, 5162-5167, December, 2007.

  In this paper, motivated in particular by models drawn from biology, we introduce the notion of box invariant dynamical systems. We argue that box invariance, that is, the existence of a box-shaped positively invariant region, is a characteristic of many biologically-inspired dynamical models. Box invariance is also useful for the verification of stability and safety properties of such systems. This paper presents effective characterization of this notion for some classes of systems, computational results on checking box invariance, the study of the dynamical properties it subsumes, and a comparison with related concepts in the literature. The concept is illustrated using models derived from different case studies in biology.

- [4]Krishnendu Chatterjee. "Markov Decision Processes with Multiple Long-run Average Objectives," FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 473-484, December, 2007.

  We consider Markov decision processes (MDPs) with multiple long-run average 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 approximated by a memoryless strategy, for all. In contrast to the single-objective case, the memoryless strategy may require randomization. We show that the Pareto curve can be approximated (a) in polynomial time in the size of the MDP for irreducible MDPs; and (b) in polynomial space in the size of the MDP for all MDPs. 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 for irreducible MDPs and in NP for all MDPs. These results provide algorithms for design exploration in MDP models with multiple long-run average objectives.

- [33]Alessandro Abate. PhD thesis, "Probabilistic Reachability for Stochastic Hybrid Systems: Theory, Computations, and Applications," University of California, Berkeley, November, 2007.

  Stochastic Hybrid Systems are probabilistic models suitable at describing the dynamics of variables presenting interleaved and interacting continuous and discrete components.

  Engineering systems like communication networks or automotive and air traffic control systems, financial and industrial processes like market and manufacturing models, and natural systems like biological and ecological environments exhibit compound behaviors arising from the compositions and interactions between their heterogeneous components. Hybrid Systems are mathematical models that are by definition suitable to describe such complex systems.

  The effect of the uncertainty upon the involved discrete and continuous dynamics—both endogenously and exogenously to the system—is virtually unquestionable for biological systems and often inevitable for engineering systems, and naturally leads to the employment of stochastic hybrid models.

  The first part of this dissertation introduces gradually the modeling framework and focuses on some of its features. In particular, two sequential approximation procedures are introduced, which translate a general stochastic hybrid framework into a new probabilistic model. Their convergence properties are sketched. It is argued that the obtained model is more predisposed to analysis and computations.

  The kernel of the thesis concentrates on understanding the theoretical and computational issues associated with an original notion of probabilistic reachability for controlled stochastic hybrid systems. The formal approach is based on formulating reachability analysis as a stochastic optimal control problem, which is solved via dynamic programming. A number of related and significant control problems, such as that of probabilistic safety, are reinterpreted with this approach. The technique is also computationally tested on a benchmark case study throughout the whole work. Moreover, a methodological application of the concept in the area of Systems Biology is presented: a model for the production of antibiotic as a component of the stress response network for the bacterium Bacillus subtilis is described. The model allows one to reinterpret the survival analysis for the single bacterial cell as a probabilistic safety specification problem, which is then studied by the aforementioned technique.

  In conclusion, this dissertation aims at introducing a novel concept of probabilistic reachability that is both formally rigorous, computationally analyzable and of applicative interest. Furthermore, by the introduction of convergent approximation procedures, the thesis relates and positively compares the presented approach with other techniques in the literature.

  Advisor: S. Shankar Sastry

- [34]Alessandro Pinto, Luca Carloni, Alberto Sangiovanni-Vincentelli. "A Communication Synthesis Infrastructure for Heterogeneous Networked Control Systems and Its Application to Building Automation and Control," EMSOFT 2007, October, 2007.

In networked control systems the controller of a physically distributed plant is implemented as a collection of tightly interacting, concurrent processes running on a distributed execution platform. The execution platform consists of a set of heterogeneous components (sensors, actuators, and controllers) that interact through a hierarchical communication network. We propose a methodology and a framework for design exploration and automatic synthesis of the communication network. We present how our approach can be applied to the design of control systems for intelligent buildings. The input specification of the control system includes (i) the constraints on the location of its components, which are imposed by the plant, (ii) the communication requirements among the components, and (iii) an estimation of the real-time constraints for the correct behavior of the algorithms implementing the control law. The output produces an implementation of the control networks that is obtained by combining elements from a pre-defined library of communication links, protocols, interfaces, and switches. The implementation is optimal in the sense that it satisfies the given specification while minimizing an objective function that captures the overall cost of the network implementation.

- [35]A. Abate, Y. Bai, N. Sznajder, C. Talcott, A. Tiwari. "Quantitative and Probabilistic Modeling in Pathway Logic," Proceedings of the 7th IEEE International Conference on BioInformatics and BioEngineering, 922-929, October, 2007.

  This paper presents a study of possible extensions of pathway logic to represent and reason about semiquantitative and probabilistic aspects of biological processes. The underlying theme is the annotation of reaction rules with affinity information that can be used in different simulation strategies. Several such strategies were implemented, and experiments carried out to test feasibility, and to compare results of different approaches. Dimerization in the ErbB signalling network, important in cancer biology, was used as a test case.

- [5]Krishnendu Chatterjee. PhD thesis, "Stochastic Omega-Regular Games," EECS Department, University of California, Berkeley, October, 2007.

  We study games played on graphs with omega-regular conditions specified as parity, Rabin, Streett or Muller conditions. These games have applications in the verification, synthesis, modeling, testing, and compatibility checking of reactive systems. Important distinctions between graph games are as follows: (a) turn-based vs. concurrent games, depending on whether at a state of the game only a single player makes a move, or players make moves simultaneously; (b) deterministic vs. stochastic, depending on whether the transition function is a deterministic or a probabilistic function over successor states; and (c) zero-sum vs. non-zero-sum, depending on whether the objectives of the players are strictly conflicting or not. We establish that the decision problem for turn-based stochastic zero-sum games with Rabin, Streett, and Muller objectives are NP-complete, coNP-complete, and PSPACE-complete, respectively, substantially improving the previously known 3EXPTIME bound. We also present strategy improvement style algorithms for turn-based stochastic Rabin and Streett games. In the case of concurrent stochastic zero-sum games with parity objectives we obtain a PSPACE bound, again improving the previously known 3EXPTIME bound. As a consequence, concurrent stochastic zero-sum games with Rabin, Streett, and Muller objectives can be solved in EXPSPACE, improving the previously known 4EXPTIME bound. We

also present an elementary and combinatorial proof of the existence of memoryless epsilon-optimal strategies in concurrent stochastic games with reachability objectives, for all real epsilon¿0, where an epsilon-optimal strategy achieves the value of the game with in epsilon against all strategies of the opponent. We also use the proof techniques to present a strategy improvement style algorithm for concurrent stochastic reachability games. We then go beyond omega-regular objectives and study the complexity of an important class of quantitative objectives, namely, limit-average objectives. In the case of limit-average games, the states of the graph is labeled with rewards and the goal is to maximize the long-run average of the rewards. We show that concurrent stochastic zero-sum games with limit-average objectives can be solved in EXPTIME. Finally, we introduce a new notion of equilibrium, called secure equilibrium, in non-zero-sum games which captures the notion of conditional competitiveness. We prove the existence of unique maximal secure equilibrium payoff profiles in turn-based deterministic games, and present algorithms to compute such payoff profiles. We also show how the notion of secure equilibrium extends the assume-guarantee style of reasoning in the game theoretic framework.

- [36]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. "Probabilistic Safety and Optimal Control for Survival Analysis of Bacillus Subtilis," Proceedings of the 2nd Conference on Foundations of Systems Biology in Engineering, 527-532, September, 2007.

  The investigation of the stress response network of Bacillus Subtilis ATCC 6633 offers a detailed explanation of how the bacterium reacts to competitive environmental conditions, among the many options, by producing the antibiotic subtilin in order to directly suppress other cells while getting immunized. The mechanisms of this generation are fairly well understood and described by a genetic and protein pathway that involves some non-deterministic interplay between the involved quantities: in particular, the presence of switching modes exhibits the activation/deactivation of certain genes and the production of proteins; these transitions in turn depend non-linearly on the above quantities.

  According to the general principles of evolution, we may postulate that the way this pathway functions is according to certain criteria and levels of optimality; in this context optimality is intended as a measure of personal fitness or, in the particular instance, of own survival. In particular, one would expect that the switches in the network happen 'optimally' in the above sense.

  In this work, we look at a recently developed dynamical model for the genetic network describing the production of subtilin and propose modifications for the model to bring it in line with other evidence reported in the literature. We obtain a system that presents partially decoupled high-level dynamics (those dealing with the population size and the nutrient level) and low-level ones (those describing the mechanism of generation of subtilin by the single cell). The high-level model is non-linear and deterministic, while the low-level one is piecewise-affine, hybrid and stochastic.

  The new model allows one to reinterpret the survival analysis for the single B. subtilis cell and study it as a probabilistic, decentralized safety specification problem over a short time horizon; it is 'probabilistic' because of the certainly stochastic dynamics, as well as according to possible 'trembling' features of the actions; it is 'over a short

time horizon' because of the greedy nature of the survival games that are played at this level; it is naturally 'decentralized' because each entity, while optimizing for its own fitness (which depends on global information), does not communicate with the competitors, nor has knowledge of their actions; furthermore, we motivate that the solution of the problem may not be globally optimal.

Using recently developed techniques for probabilistic verification in a stochastic hybrid systems setting, we reinterpret the above probabilistic safety problem as a (stochastic) optimal control one, where the controls are (possibly randomized) functions of the state-space that encode the switches in the network. Finally, the solution of this short-time-horizon, stochastic and decentralized optimal control problem yields the structure of the switching behaviors under study. Matching these outcomes with the data in the literature allows concluding that the corresponding mechanisms in the subtilin production network function with a degree of optimality, according to certain survival criteria.

- [9]Thomas Brihaye, Tom Henzinger, Vinayak Prabhu, Jean-Franois Raskin. "Minimum-time reachability in timed games," ICALP 2007 Automata, Languages and Programming, 825-837, July, 2007.

  We consider the minimum-time reachability problem in concurrent two-player timed automaton game structures. We show how to compute the minimum time needed by a player to reach a target location against all possible choices of the opponent.We do not put any syntactic restriction on the game structure, nor do we require any player to guarantee time divergence.We only require players to use receptive strategies which do not block time. The minimal time is computed in part using a fixpoint expression, which we show can be evaluated on equivalence classes of a non-trivial extension of the clock-region equivalence relation for timed automata.

- [13]Dirk Beyer, Arindam Chakrabarti, Tom Henzinger, Sanjit A. Seshia. "An Application of Web-Service Interfaces," IEEE International Conference on Web Services (ICWS) 2007, IEEE Computer Society Press, 831-838, July, 2007.

  We present a case study to illustrate our formalism for the specification and verification of the method-invocation behavior of web-service applications constructed from asynchronously interacting multi-threaded distributed components. Our model is expressive enough to allow the representation of recursion and dynamic thread creation, and yet permits the algorithmic analysis of the following two questions: (1) Does a given service satisfy a safety specification? (2)Can a given service be substituted by a another service in an arbitrary context? Our case study is based on the Amazon.com E-Commerce Services (ECS) platform.

- [37]Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Anirudda Gokhale, Sandeep Neema, Jonathan Sprinkle. Paul A. Fishwick (ed.), "Domain-Specific Modeling (in CRC Handbook of Dynamic System Modeling)," 7, (in publication), CRC Press, 2007.

  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.

- [38]A. Abate S. Amin and M. Prandini and J. Lygeros and S. Sastry. A. Bemporad A. Bicchi and G. Buttazzo (eds.), "Computational Approaches to Reachability Analysis of Stochastic Hybrid Systems," 4-17, 4416, Springer Verlag, 2007.

  This work investigates some of the computational issues involved in the solution of probabilistic reachability problems for discretetime, controlled stochastic hybrid systems. It is first argued that, under rather weak continuity assumptions on the stochastic kernels that characterize the dynamics of the system, the numerical solution of a discretized version of the probabilistic reachability problem is guaranteed to converge to the optimal one, as the discretization level decreases. With reference to a benchmark problem, it is then discussed how some of the structural properties of the hybrid system under study can be exploited to solve the probabilistic reachability problem more efficiently. Possible techniques that can increase the scale-up potential of the proposed numerical approximation scheme are suggested.

- [39]A. Abate, A. D'Innocenzo, G. Pola, M. D. Di Benedetto, S. S. Sastry. A. Bemporad and A. Bicchi and G. Buttazzo (eds.), "The Concept of Deadlock and Livelock in Hybrid Control Systems," 628-632, 4416, Springer Verlag, 2007.

  This short paper qualitatively introduces the definition of the concepts of Deadlock and Livelock for a general class of Hybrid Control Systems (HCS). Such a characterization hinges on three important aspects: firstly, the concept of composition of HCS; secondly, the general concept of specifications and their composition for HCS; finally, the dynamical structure and behaviors of HCS. The first aspect is introduced in a novel manner, including ideas from the literature of discrete transition systems and accounting for concepts such as that of dynamical feedback interconnection. The second point includes general properties that are of interest from a systems and control theory perspective. The third part categorizes the diverse and possibly pathological behaviors that are distinctive of HCS. A first look at the problem of Deadlock and Livelock Verification concludes the manuscript.

- [40]Aaron Ames. Michael Farber, R . Ghrist, M. Burger, D . Koditschek (eds.), "Homotopy Meaningful Hybrid Model Structures," 121-144, American Mathematical Society, 2007.

  Hybrid systems are systems that display both discrete and continuous behavior and, therefore, have the ability to model a wide range of robotic systems such as those undergoing impacts. The main observation of this paper is that systems of this form relate in a natural manner to very special diagrams over a category, termed hybrid objects. Using the theory of model categories, which provides a method for "doing homotopy theory" on general categories satisfying certain axioms, we are able to understand the homotopy theoretic properties of such hybrid objects in terms of their "non-hybrid" counterparts. Specifically, given a model category, we obtain a "homotopy meaningful" model structure on the category of hybrid objects over this category with the same discrete structure, i.e., a model structure that relates to the original non-hybrid model structure by means of homotopy colimits, which necessarily exist. This paper, therefore, lays the groundwork for "hybrid homotopy theory."

3 Outreach

3.1 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.

3.2 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.

3.2.1 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.

3.2.2 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.

Course: Introduction to Embedded Systems (UCB EECS 124)
http://chess.eecs.berkeley.edu/eecs124/

Instructors:
Prof. Edward A. Lee
Prof. Sanjit A. Seshia
Prof. Claire J. Tomlin

> EECS 124 is a new course, being offered on a pilot basis in Spring 2008, intended to introduce students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.
>
> A major theme of this course will be on the interplay of practical design with formal models of systems, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

Course: Introduction to Control Design Techniques (UCB EECS 128)
http://inst.eecs.berkeley.edu/ ee128/fa08//

Instructor:
Prof. Claire J. Tomlin

> In 2008, Professor Tomlin has redesigned the undergraduate control theory and engineering course, EECS 128, adding new labs and course material. The new material will be taught in the Fall Semester of 2008.
>
> The abstract for the class is below:
>
> Root-locus and frequency response techniques for control system synthesis. State-

space techniques for modeling, full-state feedback regulator design, pole placement, and observer design. Combined observer and regulator design. Lab experiments on computers connected to mechanical systems.

- Transfer function and state space models for control system analysis and synthesis. Pole locations and relationship to time response. Root locus methods. Stability.
- Feedback. Review of single-input single output (SISO) analysis and control methods in the frequency domain (Bode, Nyquist).
- SISO analysis and control using state space models. The matrix exponential and its relationship to time response. Controllability and observability. Combining state feedback with observers.
- Multi-input multi-output analysis and control using state space models.
- The linear quadratic regulator.

3.2.3 Graduate Courses

As part of the no-cost extension, a course in embedded systems was taught in the area of embedded and hybrid systems, as well as systems modeling. This course is a reflection of the teaching and curriculum goals of the ITR and its affiliated faculty.

Course: Linear System Theory(UCB EE221A)
http://inst.eecs.berkeley.edu/ ee221A/fa08//

Instructor: Claire J. Tomlin

Professor Tomlin modernizing the graduate course in linear system theory, EECS 221A, adding units in linear programming and more general optimization. The new material will be taught in the Fall Semester of 2008.

The abstract for the class is below:

This course provides a comprehensive introduction to the modeling, analysis, and control of linear dynamical systems. Topics include: A review of linear algebra and matrix theory. The solutions of linear equations. Least-squares approximation and linear programming. Linear ordinary differential equations: existence and uniqueness of solutions, the state-transition matrix and matrix exponential. Input-output and internal stability; the method of Lyapunov. Controllability and observability; basic realization theory. Control and observer design: pole placement, state estimation. Linear quadratic optimal control: Riccati equation and properties of the LQ regulator. Advanced topics such as robust control and hybrid system theory will be presented based on allowable time and interest from the class.

This course provides a solid foundation for students doing research that requires the design and use of dynamic models. Students in control, circuits, signal processing, communications and networking are encouraged to take this course.

- Linear Algebra: Fields, vector spaces, subspaces, bases, dimension, range and Null spaces, linear operators, norms, inner products, adjoints.
- Matrix Theory: Eigenspaces, Jordan form, Hermitian forms, positive definiteness, singular value decomposition, functions of matrices, spectral mapping theorem, computational aspects.

- Optimization: Linear equations, least-squares approximation, linear programming.
- Differential Equations: existence and uniqueness of solutions, Lipschitz continuity, linear ordinary differential equations, the notion of state, the state-transition matrix.
- Stability: Internal stability, input-output stability, the method of Lyapunov.
- Linear Systems - open-loop aspects: controllability and observability, duality, canonical forms, the Kalman decomposition, realization theory, minimal realizations.
- Linear systems - feedback aspects: pole placement, stabilizability and detectability, observers, state estimation, the separation principle.
- Linear quadratic optimal control: least-squares control and estimation, Riccati equations, properties of the LQ regulator.
- Advanced topics: robust control, hybrid systems.

Course: Embedded System Design: Models, Validation, and Synthesis (UCB EE249)
http://inst.eecs.berkeley.edu/ ee249/fa07/
Instructor: Prof. Alberto Sangiovanni-Vincentelli

Embedded systems are electronics systems that sense physical quantities, elaborate the data and respond to the environment by sending commands to actuators. These computing systems are everywhere: in our homes, automobiles, and work place. Their complexity increases steadily: a top-of-the-line car electrical system may include more than 80 processors that control its power train (engine and transmission) as well as its stability (suspension and chassis), interior functionality (air conditioning, displays), stability, communication (cellular) and entertainment; the comfort and security of a modern building requires the installation of thousands of sensors reporting measurements to central computers that run sophisticated control algorithms for energy-use optimization and safety functions. New methods are needed to allow designing reliable and secure distributed systems quickly, inexpensively and, most importantly, with no errors to avoid recalls and expensive retrofits. We argue that a novel system theory is needed that at the same time is computational and physical, bringing together the traditional computer science abstraction, where the physical world has been carefully and artfully hidden, and classical system theory that deals with the physical foundations of engineering where quantities such as time, power and geometric dimensions play a fundamental role in the models upon which this theory is based. The basis of this theory cannot be but a set of novel abstractions that partially expose the physical reality to the higher levels and methods to manipulate the abstractions and link them in a coherent whole.

This class presents approaches to the new system science based on theories, methods and tools that were in part developed at the Berkeley Center for Hybrid and Embedded Software Systems (CHESS) and the Giga-scale System Research Center (GSRC) where heterogeneity, concurrency, multiple levels of abstraction play an important

role and where a set of correct-by-construction refinement techniques are introduced as a way of reducing substantially design time and errors. Real-life applications including car electronics and building automation are used to illustrate system-level design methodologies and tools.

4 Publications and Products

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

4.1 Technical reports

- [1]Ben Upcroft, Michael Moser, Alex Makarenko, David Johnson, Ashod Donikian, Alen Alempijevic, Robert Fitch, Will Uther, Esten Ingar Grtli, Jan Biermeyer, Humberto Gonzalez, Todd Templeton, Vason P. srini, Jonathan Sprinkle. Technical report, "DARPA Urban Challenge Technical Paper: Sydney-Berkeley Driving Team," University of Sydney; University of Technology, Sydney; University of California, Berkeley, June, 2007.

- [18]Ethan Jackson. Technical report, "The Software Engineering of Domain-Specific Modeling Languages: A Survey Through Examples," Institute For Software Integrated Systems (ISIS), ISIS-07-807, March, 2008.

- [20]Douglas Densmore, Trevor Meyerowitz, Abhijit Davare, Qi Zhu, Guang Yang. Technical report, "Metro II Execution Semantics for Mapping," University of California, Berkeley, UCB/EECS-2008-16, February, 2008.

- [6]Krishnendu Chatterjee, Tom Henzinger, Vinayak Prabhu. Technical report, "Trading Infinite Memory for Uniform Randomness in Timed Games," EECS Department University of California, Berkeley, UCB/EECS-2008-4, January, 2008.

- [23]Saurabh Amin, Falk Hante, Alexandre Bayen. Technical report, "Exponential stability of switched hyperbolic systems in a bounded domain," UC Berkeley, 2008.

4.2 Software

- [11]Dirk Beyer, Arindam Chakrabarti, Krishnendu Chatterjee, Luca de Alfaro, Tom Henzinger, Marcin Jurdzinski, Freddy Mang, Cindy Song. "CHIC: Checking Interface Compatibility," UC Berkeley, November, 2007.

4.3 PhD theses

- [16]Trevor Meyerowitz. PhD thesis, "Single and Multi-CPU Performance Modeling for Embedded Systems," University of California at Berkeley, April, 2008.

- [14]Arkadeb Ghosal. PhD thesis, "A Hierarchical Coordination Language for Reliable Real-Time Tasks," EECS Department, University of California, Berkeley, January, 2008.

- [12]Arindam Chakrabarti. PhD thesis, "A Framework for Compositional Design and Analysis of Systems," UC Berkeley, December, 2007.

- [33]Alessandro Abate. PhD thesis, "Probabilistic Reachability for Stochastic Hybrid Systems: Theory, Computations, and Applications," University of California, Berkeley, November, 2007.

- [5]Krishnendu Chatterjee. PhD thesis, "Stochastic Omega-Regular Games," EECS Department, University of California, Berkeley, October, 2007.
- [41]Daniel Lazaro Cuadrado. PhD thesis, "Automated Distribution Simulation in Ptolemy II," Aalborg University, April, 2008.

4.4 Conference papers

- [17]Trevor Meyerowitz, Dominik Langen, Mirko Sauermann, Alberto Sangiovanni-Vincentelli. "Source-Level Timing Annotation and Simulation for a Heterogeneous Multiprocessor," Design Automation Test Europe, IEEE, March, 2008.
- [19]Krishnendu Chatterjee, Tom Henzinger, Daniel Iercan, Christoph Kirsch, Claudio Pinello, Alberto Sangiovanni-Vincentelli. "Logical Reliability of Interacting Real-Time Tasks," Design, Automation and Test in Europe, 2008. DATE '08, 909-914, March, 2008.
- [2]Krishnendu Chatterjee, Tom Henzinger, Koushik Sen. "Model-Checking omega-Regular Properties of Interval Markov Chains," Foundations of Software Science and Computation Structure (FoSSaCS) 2008, Roberto M. Amadio (ed.), 302-317, March, 2008.
- [10]Krishnendu Chatterjee, Tom Henzinger, Rupak Majumdar. "Controller Synthesis with Budget Constraints," HSCC 2008, 2008.
- [32]Alessandro Abate, Ashish Tiwari, S. Shankar Sastry. "The concept of Box Invariance for biologically-inspired dynamical systems," 46th IEEE Conference on Decision and Control and European Control, 5162-5167, December, 2007.
- [30]Saurabh Amin, Alexandre Bayen, Laurent El Ghaoui, S. Shankar Sastry. "Robust feasibility for control of water flow in a canal reservoir system," Decision and Control, 2007 46th IEEE Conference on, 1571-1577, December, 2007.
- [35]A. Abate, Y. Bai, N. Sznajder, C. Talcott, A. Tiwari. "Quantitative and Probabilistic Modeling in Pathway Logic," Proceedings of the 7th IEEE International Conference on BioInformatics and BioEngineering, 922-929, October, 2007.
- [36]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. "Probabilistic Safety and Optimal Control for Survival Analysis of Bacillus Subtilis," Proceedings of the 2nd Conference on Foundations of Systems Biology in Engineering, 527-532, September, 2007.
- [4]Krishnendu Chatterjee. "Markov Decision Processes with Multiple Long-run Average Objectives," FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 473-484, December, 2007.
- [3]Krishnendu Chatterjee. "Stochastic Muller Games are PSPACE-complete," FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 436-448, December, 2007.

- [8]Krishnendu Chatterjee, Tom Henzinger, Vinayak Prabhu. "Trading Infinite Memory for Uniform Randomness in Timed Games," HSCC: Hybrid Systems – Computation and Control, 2008.

- [7]Tom Henzinger, Krishnendu Chatterjee, Vinayak Prabhu. "Timed Parity Games: Complexity and Robustness," FORMATS: Formal Modeling and Analysis of Timed Systems, 2008; To appear.

- [31]Aaron Ames, Alessandro Abate, S. Shankar Sastry. "Sufficient Conditions for the Existence of Zeno Behavior in Nonlinear Hybrid Systems via Constant Approximations," 46th IEEE Conference on Decision and Control and European Control, 4033-4038, December, 2007.

- [9]Thomas Brihaye, Tom Henzinger, Vinayak Prabhu, Jean-Franois Raskin. "Minimum-time reachability in timed games," ICALP 2007 Automata, Languages and Programming, 825-837, July, 2007.

- [34]Alessandro Pinto, Luca Carloni, Alberto Sangiovanni-Vincentelli. "A Communication Synthesis Infrastructure for Heterogeneous Networked Control Systems and Its Application to Building Automation and Control," EMSOFT 2007, October, 2007.

- [13]Dirk Beyer, Arindam Chakrabarti, Tom Henzinger, Sanjit A. Seshia. "An Application of Web-Service Interfaces," IEEE International Conference on Web Services (ICWS) 2007, IEEE Computer Society Press, 831-838, July, 2007.

- [15]Abhijit Davare, Qi Zhu, Marco Di Natale, Claudio Pinello, Sri Kanajan, Alberto Sangiovanni-Vincentelli. "Period Optimization for Hard Real-time Distributed Automotive Systems," Design Automation Conference, 278-283, June, 2007.

4.5 Book chapters or sections

- [39]A. Abate, A. D'Innocenzo, G. Pola, M. D. Di Benedetto, S. S. Sastry. A. Bemporad and A. Bicchi and G. Buttazzo (eds.), "The Concept of Deadlock and Livelock in Hybrid Control Systems," 628-632, 4416, Springer Verlag, 2007.

- [38]A. Abate S. Amin and M. Prandini and J. Lygeros and S. Sastry. A. Bemporad A. Bicchi and G. Buttazzo (eds.), "Computational Approaches to Reachability Analysis of Stochastic Hybrid Systems," 4-17, 4416, Springer Verlag, 2007.

- [40]Aaron Ames. Michael Farber, R . Ghrist, M. Burger, D . Koditschek (eds.), "Homotopy Meaningful Hybrid Model Structures," 121-144, American Mathematical Society, 2007.

- [37]Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Anirudda Gokhale, Sandeep Neema, Jonathan Sprinkle. Paul A. Fishwick (ed.), "Domain-Specific Modeling (in CRC Handbook of Dynamic System Modeling)," 7, (in publication), CRC Press, 2007.

- [22]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Approximation of General Stochastic Hybrid Systems by Switching Diffusions with Random Hybrid Jumps," Springer Verlag, 2008; Chapter to appear in "Hybrid Systems: Computation and Control," 2008 .

- [21]Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Markov Set-Chains as Abstractions of Stochastic Hybrid Systems," Springer Verlag, 2008; Chapter to appear in "Hybrid Systems: Computation and Control", 2008.

- [29]Saurabh Amin, Falk Hante, Alexandre Bayen. Magnus Egerstedt and Bud Mishra, (eds.), "On stability of switched linear hyperbolic conservation laws with reflecting boundaries," 602-605, Hybrid Systems: Comp, Springer-Verlag, 2008.

4.6 Journal articles

- [27]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. Automatica, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," 2008; To appear.

4.7 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.

4.7.1 The 2007-2008 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 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

- "Formal Specification and Analysis of Real-Time Systems in Real-Time Maude"
  Peter Csaba Olveczky, University of Oslo, May 13, 2008

- "Partial Evaluation for Optimized Compilation of Actor-Oriented Models"
  Gang Zhou, Monday, May 12, 2008, 3:30-4:30

- "Specification and Analysis of Electronic Contracts"
  Gerardo Schneider, University of Oslo, May 6, 2008

- "Anytime Control Algorithms for Embedded Real-Time Systems"
  Luca Greco, University of Salerno, April 29, 2008

- "When can a UAV get smart with its operator, and say 'NO!'?"
  Prof. Jonathan Sprinkle, University of Arizona, April 15, 2008

- "Model-Based Design of a Power Window System: Modeling, Simulation, and Validation"
  Pieter J. Mosterman, The MathWorks, April 8, 2008.

- "From Automated Software Testing to Likely Program Invariant Generation"
  Koushik Sen, UC Berkeley, March 18, 2008.
- "Numerical solution of nonlinear differential equations in musical synthesis"
  David Yeh, Stanford, March 11, 2008.
- "Single and Multi-CPU Performance Modeling for Embedded Systems"
  Trevor Meyerowitz, UC Berkeley, February 26, 2008.
- "Model-Based Development of Fault-Tolerant Real-Time Systems"
  Prof. Alois Knoll, Technical University of Munich, February 19, 2008.
- "Enhancing the Visual Experience on the Mobile Computing and Communications Platforms"
  Achin Bhowmik, Intel Corporation, February 12, 2008.
- "Inventing and Prototyping Social Devices
  Michael Winter, Stupid Fun Club, February 5, 2008.
- "A Hierarchical Coordination Language for Reliable Real-Time Tasks"
  Arkadeb Ghosal, UC Berkeley, January 22, 2008.
- "Algorithms for an Autonomous Car"
  Edwin Olson, MIT CSAIL, January 8, 2008.
- "Reducing Energy consumption in Wireless Sensor Networks"
  Carlo Fischione, UC Berkeley, December 11, 2007.
- "From Specifications to Systems"
  Orna Kupferman, Hebrew University, December 4, 2007.
- "Communication Synthesis with Applications to On-Chip Communication and Building Automation Systems"
  Alessandro Pinto, UC Berkeley, November 27, 2007.
- "The Theory of Fast and Robust Adaptation"
  Naira Hovakimyan, Virginia Tech, November 13, 2007.
- "Using the Principles of Synchronous Languages in Discrete-event and Continuous-time Models"
  Edward Lee, UC Berkeley, October 23, 2007.
- "Design of Robust Dynamic Networks"
  Andrzej Banaszuk, United Technologies, October 16, 2007.
- "From Actors to Gates"
  Jorn Janneck, Xilinx Research Labs, October 9, 2007.
- "Ingredients for Successful System Level Automation & Design Methodology - Support for Multiple Models of Computation, Directed test case generation, Reflection & Introspection and Service-oriented tool integration environment"
  Hiren Patel, UC Berkeley, October 4, 2007.

- "Graphical System Design"
  David Fuller, National Instruments, September 26, 2007.

- "Stochastic Omega-Regular Games"
  Krishnendu Chatterjee, UC Berkeley, September 25, 2007.

- "A Multi-Threaded Reactive Processor"
  Reinhard von Hanxleden, Christian-Albrechts-Universitat (CAU) Kiel, September 18, 2007.

- "The Timing Definition Language (TDL) domain in Ptolemy"
  Stefan Resmerita, University of Salzburg, September 13, 2007.

- "Problems in Resource Modeling and Scheduling for Embedded Systems"
  Feng Zhao, Microsoft Research, September 11, 2007.

- "Symbolic Reachability Analysis of Lazy Linear Hybrid Automata"
  Susmit Jha, UC Berkeley, September 4, 2007.

- "A Formal Framework for the Correct-by-construction and Verification of Distributed Time Triggered Systems"
  Dr. Ramesh, GM India Science Lab, August 28, 2007.

4.7.2 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.

- "Grand Challenges for Real-Time Systems"
  Thomas A. Henzinger, keynote lecture, 20th Euromicro Conference on Real-Time Systems (ECRTS), Prague, Czech Republic, July 2008.

- "Challenges in Embedded Systems Design: Predictability and Robustness"
  Thomas A. Henzinger, invited lecture, Royal Society Meeting: From Computers to Ubiquitous Computing, London, United Kingdom, March 2008.

- "Three Sources of Infinity in Computation: Nontermination, Real Time and Probabilistic Choice"
  Thomas A. Henzinger, keynote lecture, First International Conference on Infinity in Logic and Computation (ILC), Cape Town, South Africa, November 2007.

- "Quantitative Generalizations of Languages"
  Thomas A. Henzinger, keynote lecture, 11th International Conference on Developments in Language Theory (DLT), Turku, Finland, July 2007.

- "Modeling, Verification, and Synthesis of Component Interfaces"
  Thomas A. Henzinger, invited tutorial, 19th International Conference on Computer-Aided Verification (CAV), Berlin, Germany, July 2007.

- "The Embedded Systems Design Challenge"
  Thomas A. Henzinger, keynote lecture, 12th International Workshop on Formal Methods for Industrial-Critical Systems (FMICS), Berlin, Germany, July 2007.

- " Using Mathematical Models to Understand Planar Cell Polarity "
  Claire J. Tomlin, plenary talk, International Conference on Systems Biology, Long Beach, October 2007.

- " Mathematical Models for Protein Regulatory Networks "
  Claire J. Tomlin, plenary talk, International Federation of Automatic Control, Nonlinear COntrol Systems Workshop, Pretoria, South Africa, August 2007.

- "Embedded Intelligence "
  Shankar Sastry, plenary talk, IEEE CASE Conference, Tempe, AZ, September 2007.

4.7.3 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.

4.8 Other Specific Products

The following software packages have been made available during this review period on the Chess website, http://chess.eecs.berkeley.edu:

- The Checker for Interface Compatibility (CHIC) is a modular verifier for behavioral compatibility checking of software and hardware components. The goal of CHIC is to be able to check that the interfaces for software or hardware components provide guarantees that satisfy the assumptions they make about each other. CHIC supports a variety of interface property specification formalisms: synchronous assume/guarantee interfaces, resource interfaces, web service interfaces, etc. The latest release, CHIC-1.2 was made available on May 30, 2008 and may be found at: http://www.eecs.berkeley.edu/ arindam/chic/

5 Contributions

This section summarizes the major contributions during this reporting period.

5.1 Within Discipline

5.1.1 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 tool chains 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.

5.1.2 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

5.1.3 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 tool suite, 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.

5.1.4 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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.

References

[1] Ben Upcroft, Michael Moser, Alex Makarenko, David Johnson, Ashod Donikian, Alen Alempijevic, Robert Fitch, Will Uther, Esten Ingar Grtli, Jan Biermeyer, Humberto Gonzalez, Todd Templeton, Vason P. srini, and Jonathan Sprinkle. Darpa urban challenge technical paper: Sydney-berkeley driving team. Technical report, University of Sydney; University of Technology, Sydney; University of California, Berkeley, June 2007.

[2] Krishnendu Chatterjee, Tom Henzinger, and Koushik Sen. Model-checking omega-regular properties of interval markov chains. In Roberto M. Amadio, editor, Foundations of Software Science and Computation Structure (FoSSaCS) 2008, pages 302–317, March 2008.

[3] Krishnendu Chatterjee. Stochastic muller games are pspace-complete. In FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, pages 436–448, December 2007.

[4] Krishnendu Chatterjee. Markov decision processes with multiple long-run average objectives. In FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, pages 473–484, December 2007.

[5] Krishnendu Chatterjee. Stochastic Omega-Regular Games. PhD thesis, EECS Department, University of California, Berkeley, October 2007.

[6] Krishnendu Chatterjee, Tom Henzinger, and Vinayak Prabhu. Trading infinite memory for uniform randomness in timed games. Technical Report UCB/EECS-2008-4, EECS Department University of California, Berkeley, January 2008.

[7] Tom Henzinger, Krishnendu Chatterjee, and Vinayak Prabhu. Timed parity games: Complexity and robustness. In FORMATS: Formal Modeling and Analysis of Timed Systems, 2008. To appear.

[8] Krishnendu Chatterjee, Tom Henzinger, and Vinayak Prabhu. Trading infinite memory for uniform randomness in timed games. In HSCC: Hybrid Systems – Computation and Control, 2008.

[9] Thomas Brihaye, Tom Henzinger, Vinayak Prabhu, and Jean-Franois Raskin. Minimum-time reachability in timed games. In ICALP 2007 Automata, Languages and Programming, pages 825–837, July 2007.

[10] Krishnendu Chatterjee, Tom Henzinger, and Rupak Majumdar. Controller synthesis with budget constraints. In Hybrid Systems: Computation and Control, 11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008. Proceedings, pages 72, 86, 2008.

[11] Dirk Beyer, Arindam Chakrabarti, Krishnendu Chatterjee, Luca de Alfaro, Tom Henzinger, Marcin Jurdzinski, Freddy Mang, and Cindy Song. Chic: Checking interface compatibility, November 2007.

[12] Arindam Chakrabarti. A Framework for Compositional Design and Analysis of Systems. PhD thesis, UC Berkeley, December 2007.

[13] Dirk Beyer, Arindam Chakrabarti, Tom Henzinger, and Sanjit A. Seshia. An application of web-service interfaces. In IEEE International Conference on Web Services (ICWS) 2007, pages 831–838. IEEE Computer Society Press, July 2007.

[14] Arkadeb Ghosal. A Hierarchical Coordination Language for Reliable Real-Time Tasks. PhD thesis, EECS Department, University of California, Berkeley, January 2008.

[15] Abhijit Davare, Qi Zhu, Marco Di Natale, Claudio Pinello, Sri Kanajan, and Alberto Sangiovanni-Vincentelli. Period optimization for hard real-time distributed automotive systems. In Design Automation Conference, pages 278–283, June 2007.

[16] Trevor Meyerowitz. Single and Multi-CPU Performance Modeling for Embedded Systems. PhD thesis, University of California at Berkeley, April 2008.

[17] Trevor Meyerowitz, Dominik Langen, Mirko Sauermann, and Alberto Sangiovanni-Vincentelli. Source-level timing annotation and simulation for a heterogeneous multiprocessor. In Design Automation Test Europe. IEEE, March 2008.

[18] Ethan Jackson. The software engineering of domain-specific modeling languages: A survey through examples. Technical Report ISIS-07-807, Institute For Software Integrated Systems (ISIS), March 2008.

[19] Krishnendu Chatterjee, Tom Henzinger, Daniel Iercan, Christoph Kirsch, Claudio Pinello, and Alberto Sangiovanni-Vincentelli. Logical reliability of interacting real-time tasks. In Design, Automation and Test in Europe, 2008. DATE '08, pages 909–914, March 2008.

[20] Douglas Densmore, Trevor Meyerowitz, Abhijit Davare, Qi Zhu, and Guang Yang. Metro ii execution semantics for mapping. Technical Report UCB/EECS-2008-16, University of California, Berkeley, February 2008.

[21] Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, and S. Shankar Sastry. Markov set-chains as abstractions of stochastic hybrid systems. In M. Egerstedt and B. Misra, editors, Hybrid Systems: Computation and Control, 2008. Springer Verlag, 2008.

[22] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Approximation of general stochastic hybrid systems by switching diffusions with random hybrid jumps. In M. Egerstedt and B. Misra, editors, Hybrid Systems: Computation and Control, 2008, pages 598–601. Springer Verlag, 2008.

[23] Saurabh Amin, Falk Hante, and Alexandre Bayen. Exponential stability of switched hyperbolic systems in a bounded domain. Technical report, UC Berkeley, 2008.

[24] Anil Aswani and Claire Tomlin. Monotone piecewise affine systems. IEEE TAC, 2008. Submitted.

[25] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Neurodynamic programming for probabilistic reachability of stochastic hybrid systems. In Submitted, 2008.

[26] Anil Aswani and Claire Tomlin. Topology based control of biological genetic networks. In CDC, 2008. Submitted.

[27] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica, 2008. To appear.

[28] Alessandro D'Innocenzo, Alessandro Abate, Maria D. Di Benedetto, and S. Shankar Sastry. Approximate abstractions of discrete-time controlled stochastic hybrid systems. In Submitted, 2008.

[29] Saurabh Amin, Falk Hante, and Alexandre Bayen. On stability of switched linear hyperbolic conservation laws with reflecting boundaries. In Magnus Egerstedt and Bud Mishra, editors, Hybrid Systems: Computation and Control, pages 602–605. Springer-Verlag, 2008.

[30] Saurabh Amin, Alexandre Bayen, Laurent El Ghaoui, and S. Shankar Sastry. Robust feasibility for control of water flow in a canal reservoir system. In Decision and Control, 2007 46th IEEE Conference on, pages 1571–1577, December 2007.

[31] Aaron Ames, Alessandro Abate, and S. Shankar Sastry. Sufficient conditions for the existence of zeno behavior in nonlinear hybrid systems via constant approximations. In 46th IEEE Conference on Decision and Control and European Control, pages 4033–4038, December 2007.

[32] Alessandro Abate, Ashish Tiwari, and S. Shankar Sastry. The concept of box invariance for biologically-inspired dynamical systems. In 46th IEEE Conference on Decision and Control and European Control, pages 5162–5167, December 2007.

[33] Alessandro Abate. Probabilistic Reachability for Stochastic Hybrid Systems: Theory, Computations, and Applications. PhD thesis, University of California, Berkeley, November 2007.

[34] Alessandro Pinto, Luca Carloni, and Alberto Sangiovanni-Vincentelli. A communication synthesis infrastructure for heterogeneous networked control systems and its application to building automation and control. In EMSOFT 2007, October 2007.

[35] A. Abate, Y. Bai, N. Sznajder, C. Talcott, and A. Tiwari. Quantitative and probabilistic modeling in pathway logic. In Proceedings of the 7th IEEE International Conference on BioInformatics and BioEngineering, pages 922–929, October 2007.

[36] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Probabilistic safety and optimal control for survival analysis of bacillus subtilis. In Proceedings of the 2nd Conference on Foundations of Systems Biology in Engineering, pages 527–532, September 2007.

[37] Jeff Gray, Juha-Pekka Tolvanen, Steven Kelly, Anirudda Gokhale, Sandeep Neema, and Jonathan Sprinkle. Domain-specific modeling. In Paul A. Fishwick, editor, CRC Handbook of Dynamic System Modeling, chapter 7, page (in publication). CRC Press, 2007.

[38] A. Abate S. Amin, M. Prandini, J. Lygeros, and S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. In A. Bemporad A. Bicchi and G. Buttazzo, editors, Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings, volume 4416, pages 4–17. HSCC, Springer Verlag, 2007.

[39] A. Abate, A. D'Innocenzo, G. Pola, M. D. Di Benedetto, and S. S. Sastry. The concept of deadlock and livelock in hybrid control systems. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings, volume 4416, pages 628–632. Springer Verlag, 2007.

[40] Aaron Ames. Homotopy Meaningful Hybrid Model Structures, pages 121–144. American Mathematical Society, 2007.

[41] Daniel Lazaro Cuadrado. Automated Distribution Simulation in Ptolemy II. PhD thesis, Aalborg University, April 2008.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-0225610

UNIVERSITY OF CALIFORNIA, BERKELEY

June 16, 2009

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2008 – May 31, 2009

Contents

1 Participants **2**
1.1 People . . . 2
1.2 Partner Organizations: . . . 3
1.3 Collaborators: . . . 3

2 Activities and Findings **4**
2.1 Project Activities . . . 4
2.1.1 ITR Events . . . 4
2.1.2 Hybrid Systems Theory . . . 4
2.1.3 Deep Compositionality . . . 4
2.1.4 Robust Hybrid Systems . . . 4
2.1.5 Hybrid Systems and Systems Biology . . . 4
2.2 ProjectFindings . . . 15

3 Outreach **21**
3.1 Project Training and Development . . . 21
3.2 Outreach Activities . . . 21
3.2.1 Curriculum Development for Modern Systems Science (MSS) . . . 21
3.2.2 Undergrad Course Insertion and Transfer . . . 21
3.2.3 Graduate Courses . . . 23

4 Publications and Products **25**
4.1 Technical reports . . . 25
4.2 Conference papers . . . 25
4.3 Book chapters or sections . . . 26
4.4 Journal articles . . . 26
4.4.1 The 2008-2009 Chess seminar series . . . 26
4.4.2 Workshops and Invited Talks . . . 29
4.4.3 General Dissemination . . . 29
4.5 Other Specific Products . . . 29

5 Contributions **29**
5.1 Within Discipline . . . 30
5.1.1 Hybrid Systems Theory . . . 30
5.1.2 Model-Based Design . . . 30
5.1.3 Advanced Tool Architectures . . . 30
5.2 Other Disciplines . . . 30
5.3 Human Resource Development . . . 30
5.4 Integration of Research and Education . . . 30
5.5 Beyond Science and Engineering . . . 31

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)
CLAIRE TOMLIN (UC BERKELEY, EECS)

POST DOCTORAL RESEARCHER:
ALVARO C ÁRDENAS (UC BERKELEY)

GRADUATE STUDENTS:
ANIL ASWANI (UC BERKELEY, PROF. TOMLIN)
JAN BIERMEYER (UC BERKELEY, PROF. SASTRY)
YOUNG-HWAN CHANG (SUMMER) (UC BERKELEY, PROF. TOMLIN)
MILOS DREZGIC (UC BERKELEY, PROF. SASTRY)
TODD TEMPLETON (UC BERKELEY, PROF. SASTRY)

STAFF:
NICK BOYD (UC BERKELEY)
JESSICA GAMBLE (UC BERKELEY)
NEAL MASTER (UC BERKELEY)
MARY P STEWART (UC BERKELEY)

BUSINESS ADMINISTRATOR:
TRACEY RICHARDS (UC BERKELEY)

EXECUTIVE DIRECTOR:
CHRISTOPHER BROOKS (UC BERKELEY)

1.2 Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY

1.3 Collaborators:

ALESSANDRO ABATE, (STANDFORD UNIVERSITY, DELFT UNIVERSITY)
PIETER ABBEEL (UNIVERSITY OF CALIFORNIA, BERKELEY)
SAURABH AMIN (UNIVERSITY OF CALIFORNIA, BERKELEY)
CHRISTIAN BERGER (RWTH AACHEN, GERMANY)
ANNARITA GIANI (UNIVERSITY OF CALIFORNIA, BERKELEY)
ANDREW HINES (UNIVERSITY OF BRITISH COLUMBIA)
CHI-YEN HUANG (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)
YU-LUN HUANG (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)
ZONG-SYUN LIN (NATIONAL CHIAO TUNG UNIVERSITY, TAIWAN)
CHRISTOS H. PAPADIMITRIOU (UNIVERSITY OF CALIFORNIA, BERKELEY)
ADRIAN PERRIG (CARNEGIE MELLON UNIVERSITY)
ANUPAM PRAKSH (UNIVERSITY OF CALIFORNIA, BERKELEY)
TANYA ROOSTA (UNIVERSITY OF CALIFORNIA, BERKELEY)
MOHAN SAROVAR (UNIVERSITY OF CALIFORNIA, BERKELEY)
BRUNO SINOPOLI (CARNEGIE MELLON UNIVERSITY)
JONATHAN SPRINKLE (UNIVERSITY OF ARIZONA)
HSINYI TSAI (NATIONAL CHIAO TUNG UNIVERSITY)

2 Activities and Findings

2.1 Project Activities

This is the seventh Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software." This year was a no-cost extension for certain researchers at the University of California, Berkeley (Center for Hybrid and Embedded Systems and Software (CHESS), http://chess.eecs.berkeley.edu. Research at the other CHESS partners: 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 ended before the period covered by this report.
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.
The CHESS ITR grant has been instrumental in supporting the launch of Tomlin's new Hybrid Systems Laboratory in Cory Hall. Specifically, the grant continues to support several new directions in systems biology, centered on the development of hybrid systems models and analysis tools for the analysis and deeper understanding of several protein regulatory networks. The grant has supported Tomlin, her PhD student Anil Aswani, and a Berkeley undergraduate, Nicholas Boyd. Two additional Berkeley undergraduates, Harendra Guturu and Eugene Li, have worked on the project though have been supported by external fellowships. The research experience obtained by these undergraduates has been instrumental in helping them decide their next steps: Guturu was accepted and is currently starting the PhD program in Electrical Engineering at Stanford, and Li has been accepted into the 5th year Masters program at Berkeley and will continue working on the project this year and next. Boyd will continue working on the project as an undergraduate this year.

2.1.1 ITR Events

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

- A weekly Chess seminar was held at Berkeley. The speakers and topics are listed in Section 4.4.1, presentations for the seminar 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. As year seven was a no-cost extension, we include only thrust areas funded by the no-cost extension.

2.1.2 Hybrid Systems Theory

2.1.3 Deep Compositionality

2.1.4 Robust Hybrid Systems

2.1.5 Hybrid Systems and Systems Biology

The CHESS ITR has enabled a new collaboration, between Tomlin's group and a group of developmental biologists at Lawrence Berkeley Labs and the Department of Molecular and Cell Biology at Berkeley. This group, led by Dr. Mark Biggin and Professor Mike Eisen, are

studying the early Drosophila development. They have developed state of the art tools for RNA and protein data collection, and have collaborated with computer vision researchers to develop a "virtual embryo", visualizing all data at once on a 3D representation of the Drosophila embryo. We have begun a collaboration with their group to design dynamic models of this system: modeling RNA and protein concentrations to try to uncover the detailed interactions between these gene products that are key in fly development. We are developing continuous and hybrid models to represent the dynamics of this system.

Early patterning in the Drosophila melanogaster embryo occurs through a complicated network of interactions involving proteins and mRNA. One such system is the pattern of hunchback mRNA in the presence of Bicoid and Kruppel protein. This system is well-studied, but there is disagreement amongst biologists between two general models. Our aim is to provide evidence to support one of the two models in contention, and we do this through system identification methods. Our general approach is to do nonlinear regression on a parametric, nonlinear partial differential equation model which incorporates transcription, diffusion, and degradation. We perform the nonlinear regression and analyze the results of the nonlinear regression. We interpret the results in the biological context, and we also compare our results to previous work on this system.

In terms of hybrid model development, we have focused in particular on the relationship between a particular class of hybrid systems, known as piecewise affine (PWA) systems, and monotone systems (which have certain properties making them amenable to stability analysis). Monotone systems are order-preserving systems: given a partial order on any two initial conditions, the trajectories of the monotone system preserve this partial order through time. There is a rich theory of strong results about the dynamics and stability of monotone systems with continuous vector fields. These existing results do not apply to piecewise affine (PWA) systems, which have discontinuous vector fields. Though the previous work on monotone systems has largely been theoretical, there is growing interest in monotone systems due to the realization that many systems in biology are monotone. Our work considers the relationship between monotone and PWA systems, which have found applications in biology. Understanding which conditions are sufficient for a PWA system to be monotone is useful, both for understanding the dynamics as well as for designing controllers. In our work, we characterize monotonicity of PWA systems. Then, we prove analogs of the Kamke-Muller and related graph theoretical theorems, both of which provide sufficient conditions for a system with continuous vector field to be monotone. Our analogs give sufficient conditions for a PWA system to be a monotone system.

More generally, we have been studying the topology of graphs representing biological influence models, and investigating the development of a corresponding "control theory" for these graphs. The traditional control scheme has been to input a signal into a plant, where the signal is derived from either an open-loop or a closed-loop. This control strategy requires that the plant be able to accept inputs or can be modified to do so. However, this situation is not always true in biological genetic networks; in these systems, there is often no input or obvious modification to allow inputs. We believe that they require a new paradigm for control. Biotechnology techniques are such that it is easier to make topological changes to a genetic network than it is to either change the states of the pathway or add more elements to the pathway. Thus, for such genetic networks it is important to develop a theory of control based on making large scale changes (e.g. genetic mutations) to the topology of the network; we provide steps towards such a theory. We highlight some useful results from monotone and hybrid systems theory, and show how these results can be used for such a topological

control scheme. We consider the cancer-related p53 pathway as an example; we analyze this system using control theory and devise a controller.

Embedded Software for National and Homeland Security

Autonomous Ground Vehicles

Ground vehicle research, although more stable than airborne vehicles, presents the subtle problem of providing intelligent behavior which operates in real-time, executes safely, and yet provides a "smooth" reaction to stimuli—i.e., the software behavior is somewhat humanized. Our application is the DARPA Urban Challenge, and we are using the ground vehicle testbed to show the performance of various algorithms and advancements in theory of switched systems, model-predictive control, parameter identification, time-triggered distributed components, and computer vision, as well as how these components work in real-time with one another. Additional concerns which this project addresses are distributed software testing, component-based design, and vehicle/sensor health monitoring. We are proving many of the theories and algorithms which have been developed in the last four years of the ITR.

Real-time Computer Vision

The goal of this task is to detect moving objects using a stereo camera system mounted on a car. This information will be used to detect, and estimate the trajectories of (possibly) mobile obstacles such as other vehicles. Note that, in this scenario, we wish to detect both objects in close proximity to our vehicle (for example, the vehicle in front of us), and also objects that are farther away (for example, oncoming vehicles), so we cannot make assumptions about whether height errors at adjacent pixels appear to be close to planar. Our solution must also be able to run in real time. In this application, we are interested in determining 3D motion in the scene, rather than the more-studied case of 2D motion in the image. In particular, we want to find the motion of objects in the scene relative to each other; if we have vehicle state data available, or if we can assume that the majority of the scene is not moving, we can determine which motion-segmented region corresponds to the background and remove its perceived motion from the perceived motion of the other objects, obtaining the motion of the other objects in a global frame.

Todd Templeton's focus in the past year has primarily been hardware and software infrastructure for the autonomous / assisted-driving vehicle testbed. This effort has culminated in external time synchronization for all car sensors, which is essential for perception algorithms that utilize multiple sensors, as well as a reusable and extensible library of communication and sensing software components that can run on the variety of hardware platforms carried in the car. In addition to a complement of four half-2U mini-ITX Linux PCs, the car carries Coldfire-based NetBurner embedded processor boards for sensors that do not support external triggering and time-synchronization; the cameras are the only sensors on the car that natively support an external trigger. A hardware interrupt on each microprocessor board is wired to the same (square wave) trigger signal as the cameras, through a solid-state relay for voltage conversion. The master external trigger (generated by a network-controllable signal generator in the car) is enabled after all of the microprocessor boards are powered on, and after the camera driver has initialized the cameras. The first falling edge of the trigger signal causes the first frame to be captured by each camera, and also provides a time-zero reference to the microprocessor boards. Subsequent falling edges of the trigger signal cause subsequent frames to be captured by the cameras, and also provide a time reference from which the microprocessor boards calibrate their internal clocks. Once the master trigger is enabled and a microprocessor board has performed the initial calibration of its clock (after

receiving the second falling edge), it begins time-stamping data from the sensor (such as an INS or LADAR) to which it is attached and sending it over the car's Ethernet network using the Spread multicast messaging service. All perception algorithms, which run on the Linux PCs, use the time-stamp on each piece of sensor data, instead of the current system time, in their calculations.

The embedded Coldfire processors in the car present a challenge to software portability. To this end, Templeton has developed an architecture compatibility library that emulates the required subset of the POSIX C standard library on top of the embedded OS, embedded C library, and NetBurner system libraries. This compatibility library also provides standard I/O over a network socket, and functionality that runs before the program's main() to initialize the system clock against the external trigger.

The software collection for this platform, hopefully to be expanded to other robotic platforms within the group and open-sourced to the larger robotics community, is called the Intelligent Robotics Toolkit (IRT). It is primarily based on software used in a previous vision-based helicopter mapping and landing project, expanded to support distributed computing across a diverse hardware and software environment. We currently have a software engineering doctoral student from RWTH Aachen University in Germany helping us to strengthen the software engineering foundations of this toolkit, and to expand its simulation and visualization capabilities.

Recently, we have achieved time synchronization / triggering across all car sensors, and taken the car out on the roads near Richmond Field Station to capture sensor data from which to formulate and tune perception algorithms.

Templeton expects to graduate in December 2009: in his final contributions, he will focus on the problem of moving object segmentation while driving, using camera data collected from the above-mentioned data-gathering trip. Moving object detection and segmentation is important for both autonomous and assisted driving for two primary reasons: to estimate the trajectories and future positions of mobile obstacles, and to remove points on mobile obstacles from the map of the static portion of the scene (without which mobile obstacles become 3D walls as they move across the scene). Hence, the motion planning process becomes similar to that in a static environment (using the static scene map), with the addition of a list of mobile obstacles and their estimated trajectories and current positions.

The proposed motion segmentation algorithm is non-parametric in that it does not assume a particular shape or appearance of mobile obstacles, which allows it to detect moving obstacles such as bicyclists and pedestrians, and to not detect stationary objects that look like cars (such as parked cars)—these static objects will be handled like any other static part of the scene by the motion planner.

Verification of Driver Augmentation Systems

There has been an explosive growth in the use of embedded systems in cars. By some accounts, it is widely expected that by the end of the decade over 50% of the cost of a car will be vehicular electronics (or veitronics). Beyond the fundamental functionality of an automobile, such as driving, stopping, and turning, a modern car provides additional features for more passenger safety, better comfort, and lower environmental impact. Thus, the number of embedded processors used in a vehicle today is in the order of 80 for luxury cars, and is expected to grow further by some accounts. A contemporary car therefore is a networked embedded system, in which subsystems need to process data and communicate over special networks such as CAN, LIN or Flexray, often within hard real-time constraints. While there has been a great deal of attention paid to hybrid and electric cars and the veitronics for their drive trains, it is our contention that Advanced Driving Assistance Systems (denoted as ADAS) present a big opportunity to apply research that can potentially have a big impact on the whole automotive industry. A key issue with the introduction of these safety critical technologies is the need to have them be verified, validated and in

some cases certified. By verification, we mean formal results guaranteeing the performance (properties such as safety, liveness or non-blocking stability) of models of intelligent software systems embedded with the physical hardware on a car; and by validation we mean testing of the theoretical verified proofs of performance on hardware. Certification usually follows formal specifications laid down by regulatory/ insurance authorities and is accompanied by verification/ validation.

Automotive OEMs and tier 1 suppliers are currently suffering from the high cost of verification and validation of the kinds of features that are being demanded by customers. While the software itself to be used in automotive systems has to undergo rigorous processes for design, implementation, audit and test, using standards such as IEEE-610 or SEI CMMI, limited tools exist for verification of automotive ADAS. These systems provide an additional order of complexity, since they are not only semi-autonomous, but also interaction with a human operator for safety critical decisions is necessary.

The state-of-the art to the extend of our knowledge is a statistical approach: High-precision driving robots or automated mini-series cars such as the VW Golf GTi "53 plus 1" are used to rigorously repeat driving situations within sub-centimeter precision. If statistical measures such as MTTF or MTBF suggest safe operation, the system is deemed deployable in road traffic.

However, there are several problems with such an approach: Foremost, repeated execution with varying initialization parameters is only feasible for limited cases, such as a Parking Assistant. Further, the testing has to be rigorous and thus is costly in monetary terms and drastically increases time-to-market.

It is therefore our strong belief, that a formal verification method for ADAS is needed, to increase car functionality, safety and energy efficiency, as well as decrease development costs and time to market. We are thus developing a new theoretical framework and algorithms for the design of mathematically verified and validated cyber-human systems from beginning with ADAS as a specific case. We are exploring the next generation of cyber-human systems, which will employ much "better models of human cognition", and hence, better assist humans in autonomous or semi-autonomous ways. This research is enabled by recent major advances in the sub-areas of computing and communication, such as stochastic hybrid models, learning methods, signal-to-symbol transformation, distributed decision making and dynamic resource allocation in geographically distributed systems without communications infrastructure.

Over the last 40 years, the fields of knowledge representation, perception, robotics, control, and learning have evolved in their separate ways. The grand vision of cyber-human systems, in which these fields combine into complete agents, has all but disappeared. However, with new results and partial reunifications of these individual sub- disciplines, the time has come to propose a reconstruction of the science of cyber-human systems. The focus of the present research is developing integrated agent designs capable of performing simple tasks reliably in unstructured environments and generating purposive activity over an unbounded period. Such a goal requires dealing with perceptual input, noisy, and partially known dynamics, real-time requirements, and complex environments containing many objects and agents.

The goal is to demonstrate our work on two examples: Automatic Parking and Highway Cruise Control with linear and lateral track, as well as negative and positive acceleration, authority. We chose the parking example, since several luxury cars already deploy such systems, and our approach can readily be compared with existing solutions. The highway example, however, is on the road map of all major OEMs, yet we argue that existing verifi-

cation approaches will fail here. This is where our major contribution will be.

Automated Driver Automation Systems (ADAS) are required to perform safety-critical tasks with hard-real time guarantees. Simulation and validation based methods alone cannot guarantee that all specifications are met and that undesired behaviors are not executed. Formal methods offer a rigorous framework to prove that a mathematical abstraction of embedded software satisfies the required correctness properties. The mathematical abstraction of ADAS embedded software can be completely behavioral in that it closely describes all possible executions or it could be simply prescriptive in that it only specifies what the system should do. In both cases, the mathematical abstraction should specify evolution of the physical environment of the system such as the effect of forces, actions of the driver, presence of external entities like obstructions, other vehicles etc. In addition, the ADAS models should also include description of maneuvers such as cruise mode, lane changing, parking, and overtaking. The specification of required correctness properties can be generally translated into safety, i.e., the system never performs a bad execution; and liveness, i.e., the system eventually performs a good execution and does not deadlock.

The modeling of ADAS requires hierarchies of different models of computation, some discrete and others continuous. Such system models are referred to as hybrid system models. These models are especially suited to modeling compound behaviors arising from composition and interaction between heterogeneous sub-systems in automotive systems. A prototypical example is the hierarchical layering of finite state machines and nonlinear continuous time differential equations.

However, modeling uncertainty associated with human behavior in verification of ADAS raises new issues and challenges: First, the human-ADAS interface determines the mode and extent of information the user has about the behavior of the system. Secondly, the modes of interaction that are not proven to be correct can raise important safety issues of such systems. In particular, the human-machine interaction can lead to unpredictable behaviors if the assumption of human user does not match with the guarantee that the ADAS can provide. As an example, if the human user gives control of a safety- critical task to the ADAS but assumes a wrong model of the system's behavior, the ADAS may not be able to successfully execute the task. Lastly, such systems also raise an important modeling question: Can human cognitive limits be reasonably modeled and incorporated in ADAS verification framework? We plan to address these research challenges by formalizing probabilistic specifications of human-ADAS interaction. We are inspired by developments in modern cognitive science which holds the viewpoint that human mind is a computational system. Prior research has also demonstrated that significant headway can be made in understanding how a human's cognitive resources can be integrated most effectively into the problem when mental models of the human, and information about the human's cognitive limits, are incorporated into the design of the control architecture in the very early stages of design specification. We will build a framework in which successive refinements of models of human cognitive functions will be checked for correctness against formal models of ADAS. Our research will benefit from the efforts to model interaction between human operators and air- traffic management systems by accounting for the physical constraints that come from applications. Horvitz at Microsoft Research has pioneered the use of Bayesian networks in the design of models of human interaction with automation: NASA operators with varying levels of expertise, interacting with the fuel control systems on board the Space Shuttle. Using these models to indicate the context and timing of the information to display to the operator, the expected utility of the operator's decisions can be greatly enhanced.

In our approach to verification, we use a multi-world semantics hierarchical system. We break up the verification task into individual components, and verify each in its own, and its interface to neighboring components in the hierarchy. An expression at each level is interpreted at the same level. Therefore, checking the truth-claim at that level is performed in its own declarative world. Relating these disconnected worlds together is a nontrivial task; however, we are collaborating with Jonathan Sprinkle from University of Arizona on using metamodels for the interface design between components. This both enables better analyzability as well as potential for using components developed by different research groups. Instead of semantic flattening, where an expression has to be both syntactically translated and semantic interpreted, we are following human reasoning and cognition, and propose the use of high-level expressions, which will be compiled into idealized lower-level expressions and then interpreted. Invariants will be used to show that higher-level truth-claims now become conditional lower level truth- claims. In this setting, higher-level truth-claims become necessary conditions. This is contrary to one-world semantics, in which higher- level claims are sufficient conditions.

Suppose one-world interpretation leads to falsification of higher- level claims that are true in multi-world semantics. This can happen, when lower-level faults are not accounted for in higher-level descriptions. Then, multi-world semantics must be split into multiple- frameworks, each dealing with the identified faults.

Our proposed framework is kept general enough to apply to various systems. From the sensory point-of-view, these levels have increasingly coarse representations of the world as one moves toward higher levels, while from the actuation point-of-view, the tasks become more strategic, supervisory and planning-oriented at the higher levels starting from "reactive" ones at lower levels.

The highest level, the Mission Planner, is the human operator in the automotive context. The output of this level could be in the form of "go to San Francisco". The Route Planner then takes this input and transforms it into an actual plan, in our context it would be an A* implementation detailing which roads to take. Next, the Maneuver Control layer takes these roads and transforms them into driving decisions, considering current sensor readings. From there, the Low- level Control transforms these driving decisions into actual control input for the physical vehicle.

This model also has what we call in analogy to database design a "roll- back mechanism": If a component cannot perform a task given from a higher-level component, it has to communicate the reasons why it failed at execution. The higher level then has to roll-back: to revert to a more feasible strategy or pre-defined back-up plan.

There are several challenges associated with the verification of such hierarchical architecture. One has to establish such a hierarchy in a formal way with different models appropriate at the respective levels: define the semantics of interaction between different levels for operational and emergency goals, define assumptions about the model of interactions between the levels, the levels of service each level can provide, and define the set of services each level must accept in order to provide requisite services.

Within this hierarchy, we are also interested in knowing how uncertainty and fault propagates between different levels and their criticality to the goals.

Our mathematical framework for the hierarchy is currently under constant change. We defined a series of increasingly complex examples of (semi)-autonomous system that we try to capture in our framework, and apply our framework to these. If properties of the system cannot be described with the current definitions, these are altered or amended. Our examples

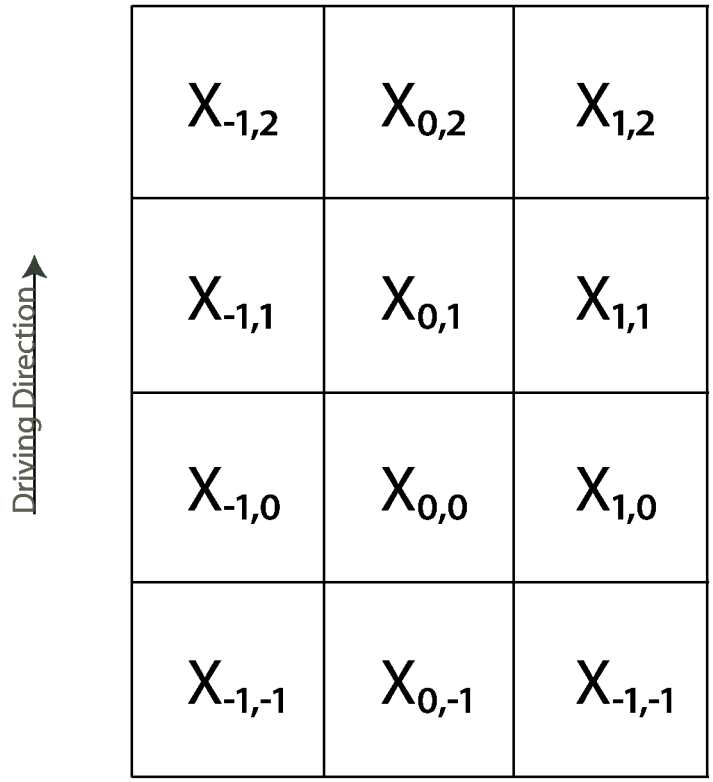

Figure 1: Matrix representation

range from the common water tank problem in hybrid systems theory via agents that have two simultaneous missions or objectives such as to traverse an area and to remain stealthy, to a collection of heterogeneous agents which each have unique sets of capabilities and have to work together towards a goal declared by the Mission Planner.

We are about to finalize this framework and intend to publish a detailed white paper in the upcoming Fall on our findings to invite feedback from the research community.

While the research effort described above is mainly aimed at verifying a (Semi-) autonomous system, an ADAS capable of making autonomous driving decisions also has to incorporate knowledge and assumptions about other traffic participants. We therefore propose the investigation into using an Algorithmic Game Theory (AGT) approach. While game theory itself is only descriptive, AGT however is prescriptive, and therefore could be used in this effort.

Clearly, interaction of cars (normally) cannot be described as an adversial game. However, it is also not really a collaborative effort, since one driver has no incentive to help another driver to achieve his goal. We are thus modeling traffic as an implicit collaboration:

Ideally, human drivers not only care for their own safety and goals, but also attempt to not inconvenience other traffic participants. Recent research conducted by our partners from the University of Aachen in co-operation with Volkswagen suggests that it suffices to use a 3x3 or 3x4 matrix to make informed LTT driving decisions. The car being the center in the matrix (X_{00} in Figure 1), its local neighborhood can be divided into segments to its sides, in front and behind, and on the diagonals. If there is any car detected in a segment, it's estimated speed is entered into the matrix. In the case of a sensor that can see further ahead than up to the next car (such as

Same advanced automotive radar), one column in driving direction extends the matrix. Each cell of the matrix contains the speed and the distance of a car, or Null if there is no car detected.

To model the interaction between cars, we are using coordination games (CG). Coordination

Figure 2: 2008 Ford Escape Hybrid ByWire XGV

games model situations in which all agents can realize mutual gains as long as they make mutually consistent decisions. For such games, there typically exist several pure strategy Nash Equilibria, so that the agents have to chose the same or corresponding strategies. However, often the coordination has to be done without any ability to communicate. Finally, some equilibria may give higher payoffs, be more fair, or, most important to the application at hand, may be safer, thus occasionally leading to interest conflicts.

We argue that this fits the interaction of cars on a highway well. We do not want to force cars to be able to communicate with each other, for two major reasons: security and deployment. Communication between cars would have to be done wirelessly, and thus provides an easy point of attack for malicious parties. Also, for such a system, all traffic participants would have to be equipped with such communication facilities, rendering it unusable for the next decade. Further, just for the interaction of two cars and one possible disturbance, we can foresee several equilibria, and cars should settle for a common equilibrium that maximizes overall welfare.

In parallel to our verification efforts, we also created the Berkeley DRIVR Lab, in close collaboration with CHESS, with the intend to showcase successfully verified algorithms. After participating in the DARPA Urban Challenge, we published a paper in AAET 2008, outlining our ideas on how to apply lessons learned from autonomous driving towards ADAS. Discussions about our paper with other researchers in the field and OEMs led to our insight on the necessity of formal ADAS verification techniques. However, other lessons learned were on the need for a robust and scalable hardware platform, as well as test bed, and a lightweight software toolkit.

We have outfitted a fully automated 2008 Ford Escape Hybrid ByWire XGV as testbed for (semi-) autonomous driving (Figure 2). Actuation is done transparently via a commercial toolkit (Torc Technologies) that directly communicates with the car's ECU. A cluster of customized quad- core mini-ITX computers is used to process data and make driving decisions, communicating among each other and with the car via Gigabit Ethernet. Various built-in screens and a KVM switch, as well as the possibility to connect notebook computers enables in-car debugging, while wireless network access enabled remote debugging, data streaming and offline analysis. All sensors, such as visual light cameras, thermal infrared cameras, laser scanners, radar, inertial measurement units and GPS are connected to time-synchronized embedded processors which trigger synchronized data-acquisition and timestamp data before relaying it to the computation cluster via Gigabit Ethernet for pre-processing, fusion and influencing driving decision making.

Since there is no robotics software framework available that satisfies all our demands for robustness, ease of use, reliability, platform independence, lightweight communication and interfaces to existing standards such as JAUS among others, we opted for implementation of our own.
Our software and computational approach is based on the following principles:

- Sensor drivers and communication methods must be abstracted as much as possible, to enable maximum flexibility in choosing different sensors and communication methods in the future.
- Transparent timestamping and synchronization facilities must be available at the sensor input level. To enable this, sensor drivers must be lightweight enough to run on an embedded processor; in particular, this means minimal or no threading.
- Transparent multicast, logging, and replaying facilities must be available at all levels (including the sensor-input level), in order to encourage the modularization of software, and to enable the independent development and verification of individual modules.
- Software modules must be able to be written in a variety of languages and run on a variety of platforms, with minimal invasiveness on the part of the infrastructure.
- Software modules should make optimal use of the computing hardware, from embedded processors via multi-core CPUs to manycore GPUS by utilizing frameworks such as Intel Thread Building Blocks, OpenMP and OpenCL or CUDA.
- Although the infrastructure is written in C/C++ due to its efficiency and ease of use, it is written with multi-platform compatibility in mind, and its interface is simple and can be easily wrapped for other programming languages, such as MATLAB.
- Where available, existing software components will be used, such as scientific and numerical libraries (BLAS, LAPACK), or vision libraries such OpenCV or the NASA Vision Workbench.

The software toolkit is currently undergoing final evaluation and testing, both internally as well as in collaboration with other research groups. We intend to release it with a free BSD license in Fall 2009. Given its ease of use, portability and free license, we are content for a broad acceptance among both academic research groups as well as commercial R&D facilities.

Control of Communication Networks

In a series of papers Abate and co-authors have continued to explore 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 (see).

Cybersecurity of Embedded Systems

2.2 ProjectFindings

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 4 below, including publications representing findings that were reported in the previous annual report.

- [1]Anil Aswani, Peter Bickel, Claire Tomlin. "Statistics for Sparse, High-Dimensional, and Nonparametric System Identification," IEEE ICRA 2009, 2133-2138, 15, May, 2009.

  Local linearization techniques are an important class of nonparametric system identification. Identifying local linearizations in practice involves solving a linear regression problem that is ill-posed. The problem can be ill-posed either if the dynamics of the system lie on a manifold of lower dimension than the ambient space or if there are not enough measurements of all the modes of the dynamics of the system. We describe a set of linear regression estimators that can handle data lying on a lower-dimension manifold. These estimators differ from previous estimators, because these estimators are able to improve estimator performance by exploiting the sparsity of the system - the existence of direct interconnections between only some of the states - and can work in the "large p, small n"' setting in which the number of states is comparable to the number of data points. We describe our system identification procedure, which consists of a presmoothing step and a regression step, and then we apply this procedure to data taken from a quadrotor helicopter. We use this data set to compare our procedure with existing procedures.

- [2]Alvaro Cárdenas, Tanya Roosta, S. Shankar Sastry. Ad Hoc Networks, "Rethinking Security Properties, Threat Models and the Design Space in Sensor Networks : A case study in SCADA systems," pp. in publication, May 2009.

  In recent years we have witnessed the emergence and establishment of research in sensor network security. The majority of the literature has focused on discovering numerous vulnerabilities and attacks against sensor networks, along with suggestions for corresponding countermeasures. However, there has been little guidance for understanding the holistic nature of sensor network security for practical deployments. In this paper, we discuss these concerns and propose a taxonomy composed of the security properties of the sensor network, the threat model, and the security design space. In particular, we try to understand the application-layer goals of a sensor network, and provide a guide to research challenges that need to be addressed in order to prioritize our defenses against threats to application-layer goals.

- [3]Saurabh Amin, Alvaro Cárdenas, S. Shankar Sastry. "Safe and Secure Networked Control Systems Under Denial-of-Service Attacks.," Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 31-45, 30, April, 2009.

  We consider the problem of security constrained optimal control for discrete-time, linear dynamical systems in which control and measurement packets are transmitted over a communication network. The packets may be jammed or compromised by a malicious adversary. For a class of denial-of-service (DoS) attack models, the goal is to find an (optimal) causal feedback controller that minimizes a given objective function

subject to safety and power constraints. We present a semi-definite programming based solution for solving this problem. Our analysis also presents insights on the effect of attack models on solution of the optimal control problem.

- [4]Alvaro Cárdenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, S. Shankar Sastry. "Challenges for Securing Cyber Physical Systems," First Workshop on Cyber-physical Systems Security, DHS, submitted, 2009.

  We discuss three key challenges for securing cyberphysical systems: (1) understanding the threats, and possible consequences of attacks, (2) identifying the unique properties of cyber-physical systems and their differences from traditional IT security, and (3) discussing security mechanisms applicable to cyber-physical systems. In particular, we analyze security mechanisms for: prevention, detection and recovery, resilience and deterrence of attacks.

- [5]Zong-Syun Lin, Alvaro Cárdenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, S. Shankar Sastry. "Model-Based Detection of Attacks for Process Control Systems," 16th ACM Computer and Communications Security Conference, ACM, submitted, 2009.

  We present security analysis of process control systems (PCS) when an attacker can compromise sensor measurements that are critical for maintaining the operational goals. We present the general sensor attack model that can represent a wide variety of DoS and deception attacks. By taking example of a well studied process control system, we discuss the consequences of sensor attacks on the performance of the system and important implications for designing defense actions. We develop model-based detection methods that can be tuned to limit the false-alarm rates while detecting a large class of sensor attacks. From the attacker's viewpoint, we show that when the detection mechanisms and control system operations are understood by the attacker, it can carry stealth attacks that maximize the chance of missed detection. From the defender's viewpoint, we show that when an attack is detected, the use of model-based outputs maintains safety under compromised sensor measurements.

- [6]Anil Aswani, Harendra Guturu, Claire Tomlin. "System identification of hunchback protein patterning in early Drosophila embryogensis," CDC 2009, 2009; Submitted.

  Early patterning in the Drosophila melanogaster embryo occurs through a complicated network of interactions involving transcription factor proteins and the mRNA of their target genes. One such system is the pattern of hunchback mRNA and its regulation by Bicoid and Krüppel proteins. This system is well-studied, but there is disagreement amongst biologists on how exactly hunchback expression is regulated. We attempt here to provide evidence to distinguish between two models in contention, through system identification. Our general approach is to do nonlinear regression on a parametric, nonlinear partial differential equation model which incorporates transcription, diffusion, and degradation. We perform the regression, analyze the results and then interpret the results in the biological context. We also compare our results to previous work on this system.

- [7]YuLun Huang, Alvaro Cárdenas, Saurabh Amin, Song-Zyun Lin , Hsin-Yi Tsai, S. Shankar Sastry. International Journal of Critical Infrastructure Protection, "Under-

standing the Physical and Economic Consequences of Attacks Against Control Systems.," pp. in publication, 2009.

This paper describes an approach for developing threat models for attacks on control systems. These models are useful for analyzing the actions taken by an attacker who gains access to control system assets and for evaluating the effects of the attacker's actions on the physical process being controlled. The paper proposes models for integrity attacks and denial-of-service (DoS) attacks, and evaluates the physical and economic consequences of the attacks on a chemical reactor system. The analysis reveals two important points. First, a DoS attack does not have a significant effect when the reactor is in the steady state; however, combining the DoS attaack with a relatively innocuous integrity attack rapidly causes the reactor to move to an unsafe state. Second, an attack that seeks to increase the operational cost of the chemical reactor involves a radically different strategy than an attack on plant safety (i.e., one that seeks to shut down the reactor or cause an explosion).

- [8]Alessandro DInnocenzo, Alessandro Abate, Maria D. Di Benedetto, S. Shankar Sastry. "Approximate Abstractions of Discrete-Time Controlled Stochastic Hybrid Systems," Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, 221-226, 9, December, 2008.

  This work presents a procedure to construct a finite abstraction of a controlled discrete-time stochastic hybrid system. The state space and the control space of the original system are partitioned by finite lattices, according to some refinement parameters. The errors introduced by the abstraction procedure can be explicitly computed, over time, given some continuity assumptions on the original model. We show that the errors can be arbitrarily tuned by selecting the partition accuracy. The obtained abstraction can be interpreted as a controlled Markov set-Chain, and can be used both for verification and control design purposes. We test the proposed technique to analyze a model from systems biology.

- [9]Anil Aswani, Peter Bickel, Claire Tomlin. Annals of Statistics, "Exterior Derivative Estimation on Manifolds," 2009; Submitted.

  Collinearity and near-collinearity of predictors cause difficulties when doing nonparametric regression on manifolds. In such scenar ios, variable selection becomes untenable because of mathematical difficulties concerning the existence and numerical stability of the regression coefficients. In addition, once computed, the regression coefficients are difficult to interpret, because a gradient does not exist for functions on manifolds. Fortunately, there is an extension of the gradient to functions on manifolds; this extension is known as the exterior derivative of a function. It is the natural quantity to estimate, because it is a mathematically well-defined quantity with a geometrical interpretation. We propose a set of novel estimators using a regularization scheme for the regression problem which considers the geometrical intuition of the exterior derivative. The advantage of this regularization scheme is that it allows us to add lasso-type regularization to the regression problem, which enables lasso-type regressions in the presence of collinearities. Finally, we consider the "large p, small n" problem in our context and show the consistency and variable selection abilities of our estimators.

- [10]Anil Aswani, Claire Tomlin. "Topology Based Control of Biological Genetic Networks," Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, 781-786,

9, December, 2008.

The traditional controller scheme has been to input a signal into a plant, where the signal is derived from either an open-loop or a closed-loop. This control strategy requires that our plant is able to accept inputs or can be modified to do so. However, this situation is not always true in biological genetic networks; in these systems, there is often no input or obvious modification to allow inputs. Many genetic networks are different, and we believe that they require a new paradigm for control. Biotechnology techniques are such that it is easier to make topological changes to a genetic network than it is to either change the states of the pathway or add more elements to the pathway (i.e. changing the "circuit"). Thus, for such genetic networks it is important to develop a theory of control based on making large-scale changes (e.g. genetic mutations) to the topology of the network. We highlight some useful results from monotone and hybrid systems theory, and show how these results can be used for such a topological controller scheme. We consider the cancer-related, p53 pathway as an example. We analyze the system using control theory and devise a controller.

- [11]Jerry Ding, Jonathan Sprinkle, S. Shankar Sastry, Claire Tomlin. "Reachability Calculations for Automated Aerial Refueling," IEEE Conference on Decision and Control, IEEE, 3706-3712, December, 2008.

  This paper describes reachability calculations for a hybrid system formalism governing UAVs interacting with another vehicle in a safety-critical situation. We examine this problem to lay the foundations toward the goal of certifying certain protocols for flight critical systems. In order to pursue these goals, we describe here what mathematical foundations are necessary to inform protocol certification, as well as describe how such formalisms can be used to automatically synthesize simulations to test against certain danger areas in the protocol. This can provide a mathematical basis for the UAV to perhaps reject a command based on the known unsafe behavior of the vehicle. We describe how creating this formalism can help to refine or design protocols for multi-UAV and/or manned vehicle interaction to avoid such scenarios, or to define appropriate behaviors in those cases.

- [12]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. Automatica, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," 44(11):2724-2734, November 2008.

  In this work, probabilistic reachability over a finite horizon is investigated for a class of discrete time stochastic hybrid systems with control inputs. A suitable embedding of the reachability problem in a stochastic control framework reveals that it is amenable to two complementary interpretations, leading to dual algorithms for reachability computations. In particular, the set of initial conditions providing a certain probabilistic guarantee that the system will keep evolving within a desired 'safe' region of the state space is characterized in terms of a value function, and 'maximally safe' Markov policies are determined via dynamic programming. These results are of interest not only for safety analysis and design, but also for solving those regulation and stabilization problems that can be reinterpreted as safety problems. The temperature regulation problem presented in the paper as case study is one such case.

- [13]Jonathan Sprinkle, J. Mikael Eklund, Humberto Gonzalez, Esten Ingar Grøtli, Pannag R Sanketi, Michael Moser, S. Shankar Sastry. Technical report, "Recovering

Models of a Four-Wheel Vehicle Using Vehicular System Data," University of California, Berkeley, August, 2008.

This paper discusses efforts to parameterize the actuation models of a four-wheel automobile for the purposes of closed-loop control. As a novelty, the authors used the equipment already available or in use by the vehicle, rather than expensive equipment used solely for the purpose of system identification. After rudimentary measurements were taken of wheelbase, axle width, etc., the vehicle was driven and data were captured using a controller area network (CAN) interface. Based on this captured data, we were able to estimate the feasibility of certain closed-loop controllers, and the models they assumed (i.e., linear, or nonlinear) for control. Examples were acceleration and steering. This work served to inform the separation of differences in simulation and vehicle behavior during vehicle testing.

- [14]Alvaro Cárdenas, Saurabh Amin, S. Shankar Sastry. "Research Challenges for the Security of Control Systems.," Proceedings of the 3rd USENIX Workshop on Hot topics in security, USENIX, Article 6, 25, July, 2008.

  In this paper we attempt to answer two questions: (1) Why should we be interested in the security of control systems? And (2) What are the new and fundamentally different requirements and problems for the security of control systems? We also propose a new mathematical framework to analyze attacks against control systems. Within this framework we formulate specific research problems to (1) detect attacks, and (2) survive attacks.

- [15]Alvaro Cárdenas, Saurabh Amin, S. Shankar Sastry. "Secure Control: Towards Survivable Cyber-Physical Systems," First International Workshop on Cyber-Physical Systems, 495-500, 17, June, 2008.

  In this position paper we investigate the security of cyber-physical systems. We (1) identify and define the problem of secure control, (2) investigate the defenses that information security and control theory can provide, and (3) propose a set of challenges that need to be addressed to improve the survivability of cyber-physical systems.

- [16]Alvaro Crdenas, Saurabh Amin, S. Shankar Sastry. "Secure Control: Towards Survivable Cyber-Physical Systems.," First International Workshop on Cyber-Physical Systems (WCPS2008)., June, 2008.

  In this position paper we investigate the security of cyberphysical systems. We (1) identify and define the problem of secure control, (2) investigate the defenses that information security and control theory can provide, and (3) propose a set of challenges that need to be addressed to improve the survivability of cyber-physical systems.

- [17]Alvaro Cárdenas, Tanya Roosta, Gelareh Taban, S. Shankar Sastry. Giorgio Franceschetti and Marina Grossi (eds.), "Cyber Security: Basic Defenses and Attack Trends," 4, 73-101, First, Artech House, 2008.

  (No abstract.)

- [18]Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Markov Set-Chains as Abstractions of Stochastic Hybrid Systems," 1, Springer Verlag, 2008.

The objective of this study is to introduce an abstraction procedure that applies to a general class of dynamical systems, that is to discrete-time stochastic hybrid systems (dt-SHS). The procedure abstracts the original dt-SHS into a Markov set-chain (MSC) in two steps. First, a Markov chain (MC) is obtained by partitioning the hybrid state space, according to a controllable parameter, into non-overlapping domains and computing transition probabilities for these domains according to the dynamics of the dt-SHS. Second, explicit error bounds for the abstraction that depend on the above parameter are derived, and are associated to the computed transition probabilities of the MC, thus obtaining a MSC. We show that one can arbitrarily increase the accuracy of the abstraction by tuning the controllable parameter, albeit at an increase of the cardinality of the MSC. Resorting to a number of results from the MSC literature allows the analysis of the dynamics of the original dt-SHS. In the present work, the asymptotic behavior of the dt-SHS dynamics is assessed within the abstracted framework.

- [19]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. "Neuro-Dynamic Programming for Probabilistic Reachability of Stochastic Hybrid Systems," Submitted, 2008.

  (No abstract.)

- [20]Anil Aswani, Claire Tomlin. IEEE TAC, "Monotone Piecewise Affine Systems," 2008; Submitted.

  (No abstract.)

- [21]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Approximation of General Stochastic Hybrid Systems by Switching Diffusions with Random Hybrid Jumps," 598 - 601, Springer Verlag, 2008; Chapter in "Hybrid Systems: Computation and Control," 2008 .

  In this work we propose an approximation scheme to transform a general stochastic hybrid system (SHS) into a SHS without forced transitions due to spatial guards. Such switching mechanisms are replaced by spontaneous transitions with state-dependent transition intensities (jump rates). The resulting switching diffusion process with random hybrid jumps is shown to converge in distribution to the original stochastic hybrid system execution. The obtained approximation can be useful for various purposes such as, on the computational side, simulation and reachability analysis, as well as for the theoretical investigation of the model. More generally, it is suggested that SHS which are endowed exclusively with random jumping events are simpler than those that present spatial forcing transitions. In the opening of this work, the general SHS model is presented, a few of its basic properties are discussed, and the concept of generator is introduced. The second part of the paper describes the approximation procedure, introduces the new SHS model, and proves, under some assumptions, its weak convergence to the original system.

3 Outreach

3.1 Project Training and Development

3.2 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.

3.2.1 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.

3.2.2 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.

Course: Introduction to Embedded Systems (UCB EECS 124)
http://chess.eecs.berkeley.edu/eecs124

Instructors:
Prof. Edward A. Lee
Prof. Sanjit A. Seshia
Prof. Claire J. Tomlin

Professor Tomlin assisted in development of the undergraduate Introduction to Embedded Systems course, EECS 124. The new material was taught in the Spring Semester of 2009 by Professor Sanjit A. Seshia.

The abstract for the class is below:

EECS 149 is a new undergraduate course, first offered on a pilot basis in Spring 2008 with course number EECS 124. This class is intended to introduce students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.

A major theme of this course will be on the interplay of practical design with formal models of systems, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

This is the second offering of the course, and we are still actively engaged in developing the course content and labs. This offering is therefore advised for advanced and adventurous undergraduates.

Course: Introduction to Control Design Techniques (UCB EECS 128)
http://inst.eecs.berkeley.edu/ ee128/fa08/

Instructor:
Prof. Claire J. Tomlin

Professor Tomlin has redesigned the undergraduate control theory and engineering course, EECS 128, adding new labs and course material. The new material was taught in the Fall Semester of 2008.

The abstract for the class is below:

Root-locus and frequency response techniques for control system synthesis. State-space techniques for modeling, full-state feedback regulator design, pole placement, and observer design. Combined observer and regulator design. Lab experiments on computers connected to mechanical systems.

- Transfer function and state space models for control system analysis and syn-

thesis. Pole locations and relationship to time response. Root locus methods. Stability.

- Feedback. Review of single-input single output (SISO) analysis and control methods in the frequency domain (Bode, Nyquist).
- SISO analysis and control using state space models. The matrix exponential and its relationship to time response. Controllability and observability. Combining state feedback with observers.
- Multi-input multi-output analysis and control using state space models.
- The linear quadratic regulator.

3.2.3 Graduate Courses

As part of the no-cost extension, a course in embedded systems was taught in the area of embedded and hybrid systems, as well as systems modeling. This course is a reflection of the teaching and curriculum goals of the ITR and its affiliated faculty.

Course: Linear System Theory(UCB EE221A)
http://inst.eecs.berkeley.edu/ ee221A/fa08/

Instructor: Claire J. Tomlin

Professor Tomlin is modernizing the graduate course in linear system theory, EECS 221A, adding units in linear programming and more general optimization. The new material was taught in the Fall Semester of 2008.

The abstract for the class is below:

This course provides a comprehensive introduction to the modeling, analysis, and control of linear dynamical systems. Topics include: A review of linear algebra and matrix theory. The solutions of linear equations. Least-squares approximation and linear programming. Linear ordinary differential equations: existence and uniqueness of solutions, the state-transition matrix and matrix exponential. Input-output and internal stability; the method of Lyapunov. Controllability and observability; basic realization theory. Control and observer design: pole placement, state estimation. Linear quadratic optimal control: Riccati equation and properties of the LQ regulator. Advanced topics such as robust control and hybrid system theory will be presented based on allowable time and interest from the class.

This course provides a solid foundation for students doing research that requires the design and use of dynamic models. Students in control, circuits, signal processing, communications and networking are encouraged to take this course.

- Linear Algebra: Fields, vector spaces, subspaces, bases, dimension, range and Null spaces, linear operators, norms, inner products, adjoints.
- Matrix Theory: Eigenspaces, Jordan form, Hermitian forms, positive definiteness, singular value decomposition, functions of matrices, spectral mapping theorem, computational aspects.
- Optimization: Linear equations, least-squares approximation, linear programming.

- Differential Equations: existence and uniqueness of solutions, Lipschitz continuity, linear ordinary differential equations, the notion of state, the state-transition matrix.
- Stability: Internal stability, input-output stability, the method of Lyapunov.
- Linear Systems - open-loop aspects: controllability and observability, duality, canonical forms, the Kalman decomposition, realization theory, minimal realizations.
- Linear systems - feedback aspects: pole placement, stabilizability and detectability, observers, state estimation, the separation principle.
- Linear quadratic optimal control: least-squares control and estimation, Riccati equations, properties of the LQ regulator.
- Advanced topics: robust control, hybrid systems.

4 Publications and Products

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

4.1 Technical reports

- [13]Jonathan Sprinkle, J. Mikael Eklund, Humberto Gonzalez, Esten Ingar Grtli, Pannag R Sanketi, Michael Moser, S. Shankar Sastry. Technical report, "Recovering Models of a Four-Wheel Vehicle Using Vehicular System Data," University of California, Berkeley, August, 2008.

4.2 Conference papers

- [16]Alvaro Crdenas, Saurabh Amin, S. Shankar Sastry. "Secure Control: Towards Survivable Cyber-Physical Systems.," First International Workshop on Cyber-Physical Systems (WCPS2008)., June, 2008.

- [10]Anil Aswani, Claire Tomlin. "Topology Based Control of Biological Genetic Networks," Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, 781-786, 9, December, 2008.

- [19]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. "Neuro-Dynamic Programming for Probabilistic Reachability of Stochastic Hybrid Systems," Submitted, 2008.

- [15]Alvaro Cárdenas, Saurabh Amin, S. Shankar Sastry. "Secure Control: Towards Survivable Cyber-Physical Systems," First International Workshop on Cyber-Physical Systems, 495-500, 17, June, 2008.

- [8]Alessandro DInnocenzo, Alessandro Abate, Maria D. Di Benedetto, S. Shankar Sastry. "Approximate Abstractions of Discrete-Time Controlled Stochastic Hybrid Systems," Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, 221-226, 9, December, 2008.

- [5]Zong-Syun Lin, Alvaro Cárdenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, S. Shankar Sastry. "Model-Based Detection of Attacks for Process Control Systems," 16th ACM Computer and Communications Security Conference, ACM, submitted, 2009.

- [14]Alvaro Cárdenas, Saurabh Amin, S. Shankar Sastry. "Research Challenges for the Security of Control Systems.," Proceedings of the 3rd USENIX Workshop on Hot topics in security, USENIX, Article 6, 25, July, 2008.

- [11]Jerry Ding, Jonathan Sprinkle, S. Shankar Sastry, Claire Tomlin. "Reachability Calculations for Automated Aerial Refueling," IEEE Conference on Decision and Control, IEEE, 3706-3712, December, 2008.

- [3]Saurabh Amin, Alvaro Cárdenas, S. Shankar Sastry. "Safe and Secure Networked Control Systems Under Denial-of-Service Attacks.," Hybrid Systems: Computation

and Control, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 31-45, 30, April, 2009.

- [4]Alvaro Cárdenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, S. Shankar Sastry. "Challenges for Securing Cyber Physical Systems," First Workshop on Cyber-physical Systems Security, DHS, submitted, 2009.

4.3 Book chapters or sections

- [17]Alvaro Cárdenas, Tanya Roosta, Gelareh Taban, S. Shankar Sastry. Giorgio Franceschetti and Marina Grossi (eds.), "Cyber Security: Basic Defenses and Attack Trends," 4, 73-101, First, Artech House, 2008.

- [21]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Approximation of General Stochastic Hybrid Systems by Switching Diffusions with Random Hybrid Jumps," 598 - 601, Springer Verlag, 2008; Chapter in "Hybrid Systems: Computation and Control," 2008 .

- [18]Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, S. Shankar Sastry. M. Egerstedt and B. Misra (eds.), "Markov Set-Chains as Abstractions of Stochastic Hybrid Systems," 1, Springer Verlag, 2008.

4.4 Journal articles

- [20]Anil Aswani, Claire Tomlin. IEEE TAC, "Monotone Piecewise Affine Systems," 2008; Submitted.

- [7]YuLun Huang, Alvaro Cárdenas, Saurabh Amin, Song-Zyun Lin , Hsin-Yi Tsai, S. Shankar Sastry. International Journal of Critical Infrastructure Protection, "Understanding the Physical and Economic Consequences of Attacks Against Control Systems.," pp. in publication, 2009.

- [2]Alvaro Cárdenas, Tanya Roosta, S. Shankar Sastry. Ad Hoc Networks, "Rethinking Security Properties, Threat Models and the Design Space in Sensor Networks : A case study in SCADA systems," pp. in publication, May 2009.

- [12]Alessandro Abate, Maria Prandini, John Lygeros, S. Shankar Sastry. Automatica, "Probabilistic Reachability and Safety for Controlled Discrete Time Stochastic Hybrid Systems," 44(11):2724-2734, November 2008.

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.

4.4.1 The 2008-2009 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 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

- "Model Transformation with Hierarchical Discrete-Event Control"
  Thomas Huining Feng, University of California, Berkeley, May 6, 2009.

- "Interval-based Abstraction Refinement and Applications"
  Pritam Roy, University of California, Santa Cruz, May 5, 2009.

- "Underpinning Empirical Architecture in a Concurrent Cyber-Physical World"
  Graham Hellestrand, Embedded Systems Technology Inc., San Carlos, California, April 28, 2009.

- "Modeling, Simulation and Analysis of Integrated Building Energy and Control Systems"
  Michael Wetter, Simulation Research Group, Lawrence Berkeley National Laboratory, Berkeley, April 21, 2009.

- "On the synthesis of correct-by-design embedded control software"
  Paulo Tabuada, University of California, Los Angeles, April 17, 2009.

- "Embedded system design with the Polychronous paradigm"
  Albert Benveniste, INRIA/IRISA, Rennes, France, April 14, 2009.

- "Mobile Millennium: using smartphones to monitor traffic in privacy aware environments"
  Alexandre Bayen, University of California, Berkeley, April 7, 2009.

- "1: Design as You See FIT: System-Level Soft Error Analysis of Sequential Circuits & 2: Optimizations of an Application-Level Protocol for Enhanced Dependability in FlexRay"
  Daniel Holcomb and Wenchao Li, UC Berkeley, March 31, 2009.

- "Integrative Modeling and Heterogeneous System Simulation"
  Dr. Jonathan Sprinkle, University of Arizona, March 17, 2009.

- "Manycore Vector-Thread Architectures"
  Christopher Batten, University of California, Berkeley, March 10, 2009.

- "Uses of Synchronized Clocks in Test and Measurement Systems"
  Jeff Burch and John Eidson, Agilent Labs, March 3, 2009.

- "Implementing Synchronous Models on Distributed Execution Platforms"
  Stavros Tripakis, University of California, Berkeley, February 24, 2009.

- "The APES-LESS project: Access Point Event Simulation of Legacy Embedded Software Systems"
  Stefan Resmerita, University of Salzburg, February 17, 2009.

- "Model-Based Development of Fault-Tolerant Real-Time Systems"
  Christian Buckl, TU Munchen, February 10, 2009.
- "Modular Code Generation from Synchronous Block Diagrams: Modularity vs. Reusability vs. Code Size"
  Stavros Tripakis, University of California, Berkeley, February 3, 2009.
- "Time Portable Programming the JAviator in Tiptoe OS"
  Christoph Kirsch, University of Salzburg, January 13, 2009.
- "Synchronous Reactive Communication: Generalization, Implementation, and Optimization"
  Guoqiang (Gerald) Wang, University of California, Berkeley, December 9, 2008.
- "Service Component Architecture (SCA)"
  Luciano Resende, IBM Silicon Valley, San Jose, California, December 2, 2008.
- "Process Network in Silicon: A High-Productivity, Scalable Platform for High-Performance Embedded Computing"
  Mike Butts, Ambric, Inc., Beaverton, Oregon, November 25, 2008.
- "The Design of a Platform-Based Design Environment for Synthetic Biology"
  Douglas Densmore, UC Berkeley, Berkeley, California, November 18, 2008.
- "The Nimrod/K director for the Kepler workflow environment"
  Colin Enticott, Monash University, Australia, November 12, 2008.
- "Toward the Predictable Integration of Real-Time COTS Based Systems"
  Marco Caccamo, University of Illinois, Urbana Champaign, October 28, 2008.
- "Model Engineering"
  Edward A. Lee and the Ptolemy Pteam, University of California, Berkeley, Berkeley, October 21, 2008.
- "Verification-Guided Error Resilience"
  Prof. Sanjit Seshia, University of California, Berkeley, Berkeley, October 7, 2008.
- "Game-Theoretic Timing Analysis"
  Sanjit Seshia, UC Berkeley, Berkeley, California, September 23, 2008.
- "Integrating Timing with Data and Space Availability as Firing Rules for Actors in a Graphical Dataflow Language"
  Tim Hayles, National Instruments, Austin, Texas, September 9, 2008.
- "On Computer Science in Systems Biology"
  Oded Maler, CNRS-VERIMAG, Grenoble, France, August 22, 2008.
- "From Control Loops to Software"
  Oded Maler, CNRS-VERIMAG, Grenoble, France, August 21, 2008.
- "Timed Automata: Modeling and Analysis"
  Oded Maler, CNRS-VERIMAG, Grenoble, France, August 20, 2008.

- "Verification for Dummies: A Gentle Introduction to Formal Verification and Hybrid Systems"
  Oded Maler, CNRS-VERIMAG, Grenoble, France, August 19, 2008.
- "Modular Timed Graph Transformation"
  Hans Vangheluwe, McGill University, Quebec, Canada, August 6, 2008.
- "Stability Analysis of Switched Systems using Variational Principles"
  Michael Margaliot, Tel Aviv University, Israel, August 5, 2008.
- "Buffer Capacity Computation for Throughput Constrained Streaming Applications with Data-Dependent Inter-Task Communication"
  Maarten Wiggers, University of Twente, The Netherlands, July 31, 2008.
- "A Type System for the Automatic Distribution of Higher-order Synchronous Dataflow Programs"
  Alain Girault, INRIA, June 10, 2008
- "Heterogeneous System Design with Functional DIF"
  William Plishker, University of Maryland, June 5, 2008.
- "Advanced topics in model-based software development"
  Bernhard Rumpe, Braunschweig University of Technology, June 3rd, 2008

4.4.2 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.

-

4.4.3 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.

4.5 Other Specific Products

The following software packages have been made available during this review period on the Chess website, http://chess.eecs.berkeley.edu:

- Upcoming software release in Fall, 2009, working title "Berkeley Intelligent Robotics Toolkit", to appear on http://robotics.eecs.berkeley.edu/drive/.

5 Contributions

This section summarizes the major contributions during this reporting period.

5.1 Within Discipline

5.1.1 Hybrid Systems Theory

5.1.2 Model-Based Design

5.1.3 Advanced Tool Architectures

5.2 Other Disciplines

5.3 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.

5.4 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.

5.5 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.

References

[1] Anil Aswani, Peter Bickel, and Claire Tomlin. Statistics for sparse, high-dimensional, and nonparametric system identification. In IEEE ICRA 2009, pages 2133–2138, May 2009.

[2] Alvaro Cardenas, Tanya Roosta, and S. Shankar Sastry. Rethinking security properties, threat models and the design space in sensor networks : A case study in scada systems. Ad Hoc Networks, page in publication, May 2009.

[3] Saurabh Amin, Alvaro Cardenas, and S. Shankar Sastry. Safe and secure networked control systems under denial-of-service attacks. In Hybrid Systems: Computation and Control, pages 31–45. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, April 2009.

[4] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and S. Shankar Sastry. Challenges for securing cyber physical systems. In First Workshop on Cyber-physical Systems Security, page submitted. DHS, 2009.

[5] Zong-Syun Lin, Alvaro Cardenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, and S. Shankar Sastry. Model-based detection of attacks for process control systems. In 16th ACM Computer and Communications Security Conference, page submitted. ACM, 2009.

[6] Anil Aswani, Harendra Guturu, and Claire Tomlin. System identification of hunchback protein patterning in early drosophila embryogensis. In CDC 2009, 2009. Submitted.

[7] YuLun Huang, Alvaro Cardenas, Saurabh Amin, Song-Zyun Lin, Hsin-Yi Tsai, and S. Shankar Sastry. Understanding the physical and economic consequences of attacks against control systems. International Journal of Critical Infrastructure Protection, page in publication, 2009.

[8] Alessandro D'Innocenzo, Alessandro Abate, Maria D. Di Benedetto, and S. Shankar Sastry. Approximate abstractions of discrete-time controlled stochastic hybrid systems. In Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, pages 221–226, December 2008.

[9] Anil Aswani, Peter Bickel, and Claire Tomlin. Exterior derivative estimation on manifolds. Annals of Statistics, 2009. Submitted.

[10] Anil Aswani and Claire Tomlin. Topology based control of biological genetic networks. In Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, pages 781–786, December 2008.

[11] Jerry Ding, Jonathan Sprinkle, S. Shankar Sastry, and Claire Tomlin. Reachability calculations for automated aerial refueling. In IEEE Conference on Decision and Control, pages 3706–3712. IEEE, December 2008.

[12] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica, 44(11):2724–2734, November 2008.

[13] Jonathan Sprinkle, J. Mikael Eklund, Humberto Gonzalez, Esten Ingar Grtli, Pannag R Sanketi, Michael Moser, and S. Shankar Sastry. Recovering models of a four-wheel vehicle using vehicular system data. Technical report, University of California, Berkeley, August 2008.

[14] Alvaro Cardenas, Saurabh Amin, and S. Shankar Sastry. Research challenges for the security of control systems. In Proceedings of the 3rd USENIX Workshop on Hot topics in security, page Article 6. USENIX, July 2008.

[15] Alvaro Cardenas, Saurabh Amin, and S. Shankar Sastry. Secure control: Towards survivable cyber-physical systems. In First International Workshop on Cyber-Physical Systems, pages 495–500, June 2008.

[16] Alvaro Crdenas, Saurabh Amin, and S. Shankar Sastry. Secure control: Towards survivable cyber-physical systems. In First International Workshop on Cyber-Physical Systems (WCPS2008)., June 2008.

[17] Alvaro Cardenas, Tanya Roosta, Gelareh Taban, and S. Shankar Sastry. Cyber Security: Basic Defenses and Attack Trends, chapter 4, pages 73–101. Artech House, first edition, 2008.

[18] Alessandro Abate, Alessandro D'Innocenzo, Maria D Di Benedetto, and S. Shankar Sastry. Markov Set-Chains as Abstractions of Stochastic Hybrid Systems, volume 1, pages 1–15. Springer Verlag, 2008.

[19] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Neurodynamic programming for probabilistic reachability of stochastic hybrid systems. In Submitted, 2008.

[20] Anil Aswani and Claire Tomlin. Monotone piecewise affine systems. IEEE TAC, 2008. Submitted.

[21] Alessandro Abate, Maria Prandini, John Lygeros, and S. Shankar Sastry. Approximation of General Stochastic Hybrid Systems by Switching Diffusions with Random Hybrid Jumps, pages 598 – 601. Springer Verlag, 2008. Chapter in "Hybrid Systems: Computation and Control," 2008.

ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-0225610

UNIVERSITY OF CALIFORNIA, BERKELEY

November 27, 2010

PERIOD OF PERFORMANCE COVERED: JUNE 1, 2009 – August 31, 2010

Contents

1 Participants **3**
1.1 People . . . 3
1.2 Partner Organizations: . . . 3
1.3 Collaborators: . . . 4

2 Activities and Findings **5**
2.1 Project Activities . . . 5
2.1.1 ITR Events . . . 5
2.1.2 Hybrid Systems Theory . . . 5
2.1.3 Robust Hybrid Systems . . . 8
2.1.4 Hybrid Systems and Systems Biology . . . 8
2.1.5 Embedded Software for National and Homeland Security . . . 12
2.1.6 Control of Communication Networks . . . 20
2.2 ProjectFindings . . . 21

3 Outreach **26**
3.1 Project Training and Development . . . 26
3.2 Outreach Activities . . . 26
3.2.1 Curriculum Development for Modern Systems Science (MSS) . . . 26
3.2.2 Undergrad Course Insertion and Transfer . . . 26
3.2.3 Graduate Courses . . . 28

4 Publications and Products **30**
4.1 Technical reports . . . 30
4.2 Software . . . 30
4.3 PhD theses . . . 30
4.4 Conference papers . . . 30
4.5 Books . . . 31
4.6 Journal articles . . . 31
4.6.1 The 2009-2010 Chess seminar series . . . 31
4.6.2 Workshops and Invited Talks . . . 34
4.6.3 General Dissemination . . . 34
4.7 Other Specific Products . . . 34

5 Contributions **34**
5.1 Human Resource Development . . . 34
5.2 Integration of Research and Education . . . 35
5.3 Beyond Science and Engineering . . . 35

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)
CARLO SEQUIN (UC BERKELEY, EECS)
CLAIRE TOMLIN (UC BERKELEY, EECS)
POST DOCTORAL RESEARCHER:
ALVARO CÁRDENAS (UC BERKELEY)
GRADUATE STUDENTS:
ANIL ASWANI (UC BERKELEY, PROF. TOMLIN)
JAN BIERMEYER (UC BERKELEY, PROF. SASTRY)
YOUNG-HWAN CHANG (SUMMER) (UC BERKELEY, PROF. TOMLIN)
MILOS DREZGIC (UC BERKELEY, PROF. SASTRY)
SLEIMAN ITANI (UC BERKELEY, PROF. TOMLIN)
TODD TEMPLETON (UC BERKELEY, PROF. SASTRY)
INSOON YANG (UC BERKELEY, PROF. TOMLIN)
STAFF:
NEAL MASTER (UC BERKELEY, ASSISTANT II - TOMLIN)
ZHAOQIAN XIA (UC BERKELEY, ASSISTANT IV - TOMLIN)
JESSICA GAMBLE (UC BERKELEY, ASSISTANT II - SASTRY & TOMLIN)
MARY P STEWART (UC BERKELEY, PROGRAMMER - SASTRY & TOMLIN)
BUSINESS ADMINISTRATOR:
GLADYS KHOURY (UC BERKELEY)
EXECUTIVE DIRECTOR:
CHRISTOPHER BROOKS (UC BERKELEY)

1.2 Partner Organizations:

UNIVERSITY OF CALIFORNIA, BERKELEY

1.3 Collaborators:

SAURABH AMIN (UNIVERSITY OF CALIFORNIA, BERKELEY)
CHRISTIAN BERGER (SOFTWARE ENGINEERING, RWTH AACHEN)
PETER BICKEL (UC BERKELEY)
MARK BIGGIN (LAWRENCE BERKELEY NATIONAL LABORATORY)
JAMES BROWN (UC BERKELEY)
JOHN CHUANG (UC BERKELEY)
CHARLES C. FOWLKES (UC IRVINE)
CHRISTOPHER GEYER (UC BERKELEY, CMU, IROBOT)
JOEL GRAY (LAWRENCE BERKELEY NATIONAL LABORATORY)
JENS GROSSKLAGS (PRINCETON UNIVERSITY)
JAY GROVES (UC BERKELEY, CHEMISTRY)
CHRIS HOOFNAGLE (UC BERKELEY)
YU-LUN HUANG (NATIONAL CHIAO TONG UNIVERSITY, TAIWAN
SOILE VE KERANEN (LBNL)
DAVID W. KNOWLES (LBNL)
JAMES KORKOLA (LAWRENCE BERKELEY NATIONAL LABORATORY)
ZONG-SYUN LIN (NATIONAL CHIAO TONG UNIVERSITY, TAIWAN
GARRY P. NOLAN (STANFORD)
TANYA ROOSTA (UC BERKELEY)
BERNHARD RUMPE (SOFTWARE ENGINEERING, RWTH AACHEN)
KAREN SACHS (STANFORD)
PAUL SPELLMAN (LAWRENCE BERKELEY NATIONAL LABORATORY)
HSIN-YI TSAI (NATIONAL CHIAO TONG UNIVERSITY, TAIWAN
NICKOLAS WANG (LAWRENCE BERKELEY NATIONAL LABORATORY)
QIAN XU (UC BERKELEY, CHEMISTRY)

2 Activities and Findings

2.1 Project Activities

This is the eighth Annual Report for the NSF Large ITR on "Foundations of Hybrid and Embedded Systems and Software." This year was a no-cost extension for certain researchers at the University of California, Berkeley (Center for Hybrid and Embedded Systems and Software (CHESS), http://chess.eecs.berkeley.edu. Research at the other CHESS partners: ISIS at Vanderbilt University (Institute for Software Integrated Systems, and the Department of Mathematical Sciences, at the University of Memphis ended before the period covered by this report.
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.

2.1.1 ITR Events

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

- A weekly Chess seminar was held at Berkeley. The speakers and topics are listed in Section 4.6.1, presentations for the seminar 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. As year eight was a no-cost extension, we include only thrust areas funded by the no-cost extension.

2.1.2 Hybrid Systems Theory

Locally-linear system identification: a new statistical method based on the estimation of the exterior derivative

As part of our system biology efforts, we have developed models for pattern formation in Drosophila embryos, based on significant prior knowledge about the system being modeled. This prior knowledge is used to formulate a specific functional form of the equations with free tuning parameters that are picked using regression procedures in order to match experimental data. The advantage of this approach is that it provides models that are highly interpretable, with specific terms of the equations corresponding to specific phenomenon in the system. The disadvantage is that it does require extensive knowledge about the system.

An alternative form of system identification is the class of nonparametric techniques; they are sometimes used in the robotics and control communities. The advantage of these techniques is that they are well-suited for situations in which there is little or no prior knowledge about the system being modeled. The disadvantage of nonparametric techniques is that they are statistically not as well behaved as parametric techniques. We have developed a new statistical technique for nonparametric system identification that has better statistical performance than existing techniques.

The advantage of requiring limited prior knowledge is important in biological applications, because there is often little or unreliable knowledge about the large networks being modeled. This makes nonparametric methods particularly useful for network identification problems in biology. In such problems, we are interested in learning not only parameter values of the model, but also the interconnections between different parts of the network. In fact, for some of the drug multitarget selection tools discussed in the second part of this thesis, it is more important to know the interconnections than to know the parameter values. From an informal perspective, systems described well by ordinary differential equations (ODEs) can be nonparametrically identified by matching trajectory data of the system with an ODE whose functional form is given by a series expansion.

Local linearization techniques are an important class of nonparametric system identification. The prevalence of control tools which utilize piecewise-affine models or linear models makes this class of techniques important. Nonparametric identification techniques involve solving regression problems of the form $Y = X\beta$ where Y is a vector of noisy response variables, X is a matrix of noisy predictor variables, and β is a vector of regression coefficients. Identifying β can be challenging due to collinearity of the predictors (which can come about if the system evolves on a manifold with dimension lower than that of the state space). Of techniques that explicitly consider manifold structure, none of these can take advantage of any sparsity in the system, that is, systems which have an upper bound on the number of non-zero entries in the linearization. Dynamical systems found in engineering problems are typically sparse because they are an interconnection of preexisting, engineered components. Biological systems (especially regulatory networks) are often sparse as well because the organization of such networks is known to be hierarchical. It is important to develop identification techniques that take advantage of this special structure.

We interpret collinearities in the language of manifolds, and this provides the two contributions of our work. This interpretation allows us to develop a new method to do regression in the presence of collinearities or near-collinearities. This insight also allows us to provide a novel interpretation of regression coefficients when there is significant collinearity of the predictors. On a statistical level, our idea is to learn the manifold formed by the predictors and then use this to regularize the regression problem. This form of regularization is informed by the ideas of manifold geometry and the exterior derivative. Our idea is to learn the manifold either locally (in the case of a local, nonlinear manifold) or globally (in the case of a global, linear manifold). The regression estimator is posed as a least-squares problem with an additional term which penalizes for the regression vector lying in directions perpendicular to the manifold. Our manifold interpretation provides a new interpretation of the regression coefficients. The gradient describes how the function changes as each predictor is changed independently of other predictors. This is impossible to do when there is collinearity of the predictors, and the gradient does not exist. The exterior derivative of a function tells us how the function value changes as a predictor and its collinear terms are simultaneously changed, and it has applications in control engineering, physics and mathematics. In particular, most of our current work is in high-dimensional system identification for biological and control engineering systems. We interpret the regression coefficients in the presence of collinearities

as the exterior derivative of the function. The exterior derivative interpretation is useful because it says that the regression coefficients only give derivative information in the directions parallel to the manifold, and the regression coefficients do not give any derivative information in the directions perpendicular to the manifold. If we restrict ourselves to computing regression coefficients for only the directions parallel to the manifold, then the regression coefficients are unique and they are uniquely given by the exterior derivative. This is not entirely a new interpretation. Similar geometric interpretations are found in the literature, but our interpretation is novel because of two main reasons. The first is that it is the first time the geometry is interpreted in the manifold context, and this is important for many application domains. The other reason is that this interpretation allows us to show that existing regularization techniques are really estimates of the exterior derivative, and this has important implications for the interpretation of estimates calculated by existing techniques. Our regularization scheme can improve estimation error, and it can be easily modified to include lasso-type regularization.

This work is presented in [1], which has been accepted to the Annals of Statistics.

Topology-based Control

Control theory has traditionally focused on a core group of goals: to stabilize a plant, to improve plant performance, to robustify a plant, to track a reference, or to perform motion planning. In engineering systems, these goals have been achieved through an analysis-design flow; this flow is rarely linear: there is often a need to go back to previous steps and incorporate things that were missed in earlier attempts. Beginning with design specifications, we write mathematical models for the engineering system, analyze these models, and devise a controller. We also implement the controller on actual hardware.

The traditional control scheme has been to input a signal into a plant, using either an open-loop or a closed-loop controller. Such a control strategy is possible if the plant is able to accept inputs or can be modified to do so. However, this situation is not always true in biological genetic networks; in these systems, there is often no input or obvious modification to allow inputs. Instead of inputs, genetic networks are more easily influenced through large-scale modifications. Genetic networks are different from traditional engineering systems and require a new paradigm for control. It is often easier to change the topology of a genetic network than it is to either change the states or elements of the network. For instance, a state could be the concentration of a protein within a cell, something which is difficult to affect to within any order of precision. Additionally, it is sometimes difficult or not feasible to modify or insert pathways by adding elements. Thus, for genetic networks it is important to develop a theory of control based on making largescale changes (e.g. genetic changes or pharmaceutical drugs) to the topology of the genetic network. Fundamentally, medical treatments seek to change how a cell operates and go beyond modifying the cellular environment.

Genetic networks can be modified in a variety of ways. The most basic is the use of pharmaceutical drugs, many of which prevent certain reactions from occurring or remove a state from a network. Biotechnology techniques allow for the insertion of genetic material into bacteria, and are commonly used for alternative energy and pharmaceutical applications.

In another technique, the genetic material of a virus is replaced with useful, genetic material. Next, the host is infected with the virus, and this inserts the useful, genetic material into the host. This control technique is being studied for use in pharmaceutical applications such as cystic fibrosis. Biologists continue to develop new techniques, amongst which include the use of microRNA and single interfering RNA. Though many of these techniques are established and used in practice, there is a lack of a systematic theory or methodology to determine which modifications to make or what to target with pharmaceutical drugs. Biological research often involves the use of intuition or trial-and-error to determine which changes are or are not beneficial for the purposes of controlling a biological system. This chapter proposes the idea of abstracting the effect of pharmaceutical drugs as modifying the topology of the biological network, and it also proposes how this abstraction might be used to do control by identifying drug targets.

Piecewise-affine (PWA) hybrid systems and ordinary differential equation (ODE) models of biological systems are considered in this work. Two different types of models are used for reasons of analysis: The simpler, hybrid systems models are easier to analyze for global behavior, and the more detailed, ODE models are easier to analyze for local behavior of small components of the network. Controllers using ODE theory are defined and analyzed, and these approaches are used to analyze and build a controller for the p53 pathway: a pathway that is related to cancer. This topological control changes the topology of the network by applying a pharmaceutical drug or other chemical, and the topology remains changed only in the presence of this pharmaceutical. As soon as it degrades away, the topology of the network goes back to an uncontrolled, unchanged state. Since the control is topological, it is crucial to have a correctly identified network. The approach described is unable to deal with latent variables that are unidentified, because the presence of latent variables can drastically change the behavior of the system.

This work is presented in [2] in the IEEE Transactions on Automatic Control, as well as in [3] and [4].

2.1.3 Robust Hybrid Systems

This year we worked on the use of hybrid systems tools to understand some issues in quantum computing [5].

2.1.4 Hybrid Systems and Systems Biology

The CHESS ITR enabled significant progress in Systems Biology in 2009-2010. The existing collaboration between Tomlin's lab and Mark Biggin's group in early Drosophila development at Lawrence Berkeley Labs was strengthened, with new results and several publications. A new collaboration between Tomlin's Lab and Joe Gray's group in breast cancer at Lawrence Berkeley Labs was begun, launching several new modeling efforts. CHESS provided critical funding for the start of this cancer modeling work, for which in mid-2010 new funding from the National Cancer Institute was awarded.

Statistical System Identification in Drosophila embryogenesis

Inferring regulatory networks in animals is challenging because of the large number of genes and the presence of redundant and indirect interactions. To build the highest quality models, it will be necessary to use multiple data sets, including: gene expression, genome wide binding, and network perturbation data. However, combining multiple data types to infer regulatory networks is still an open problem. An intermediate problem is to use only gene expression data to infer regulatory networks. The relationships between the expression levels of transcription factors and target genes are used to predict which genes are regulatory. While much work has been done in this area, it is critical to understand the maximum amount of information that can be obtained about the network using this strategy.

Typical approaches for inferring regulatory networks have been to assume a model formulation and have then fitted the data to this formulation (parametric system identification). Many models have been proposed, including coexpression networks, information-theoretic, regression onto dynamical systems and graphical models (including Bayesian networks). The primary differences between these models lie in the trade-off between statistical and interpretational issues. Techniques like Bayesian networks, graphical models, and information theoretic models have protections against over-fitting (i.e., fitting models with many parameters to a small amount of experimental data); however, these techniques do not provide dynamical models which can generate new biological insights. On the other hand, techniques such as nonlinear regression networks and regression onto dynamical systems provide more biologically interpretable models, but sometimes suffer from inaccurate assumptions or over-fitting of the model to the data. There is disagreement on the necessity of dynamical as opposed to static models. Dynamical models are more philosophically pleasing because regulatory networks contain temporal characteristics: For example, a protein binds to DNA and initiates transcription, which eventually leads to transport of the mature mRNA to the cytoplasm. Yet the argument is often made that static models provide a quasi-steady-state interpretation of the network that may provide a sufficient approximation. Rigorous comparison of the two approaches is lacking. Dynamical modeling of animal regulatory networks has a long history and is a powerful approach in which researchers hypothesize a set of nonlinear, differential equations to describe the network, but it requires significant prior knowledge about the network. If there is insufficient biological knowledge about the network, then the structure of the equations can be incorrectly chosen. And if the model is not carefully chosen, it will have a large number of parameters, possibly leading to weak biological effects being erroneously identified as strong effects. Furthermore, it is sometimes shown that a wide range of different parameter values can reproduce the biological behavior of the network, which could be taken as evidence for either network robustness or over-fitting.

We have applied the local linearization system identification described in the previous section to a network found in Drosophila embryos. The particular network which is studied here is the formation of eve mRNA stripes during stage 5 of embryogenesis, and this work was in collaboration with Mark Biggin and the Berkeley Drosophila Transcription Network Project (BDTNP). The dynamic model that we designed was shown to be significantly better than static models, and new biological insights about the developmental system were derived.

Compared to other dynamical methods, our approach requires minimal information about

the mathematical structure of the ODE; it does not use qualitative descriptions of interactions within the network; and it employs our new statistical method to protect against over-fitting. It generates spatio-temporal maps of factor activity, highlighting the times and spatial locations at which different regulators might affect target gene expression levels. We identify an ODE model for eve mRNA pattern formation in the Drosophila melanogaster blastoderm and show that this reproduces the experimental patterns well. Compared to a non-dynamic, spatial-correlation model, our ODE gives 59% better agreement to the experimentally measured pattern. Our model suggests that protein factors frequently have the potential to behave as both an activator and inhibitor on the same cis-regulatory module depending on the factors' concentration, and implies different modes of activation and repression.

This work has been written up and appears in [6], in BMC Bioinformatics.

Optimization-based inference for temporally evolving networks with applications to biology

In this work, we design an optimization-based inference scheme to identify temporally evolving (Boolean) network representations of genetic networks from data. In the formulation of the optimization problem, we use an adjacency map as a priori information, and define a cost function which both drives the connectivity of the graph to match biological data as well as generates a sparse and robust network at corresponding time intervals. The main idea of our scheme lies in representing the captured relationship as a network path with a priori information (a given connectivity map) and using convex optimization techniques to find the time-varying sparsest graph consistent with experimental observations. Despite uncertainties about details for a given biological system, we often have reasonable qualitative knowledge about interactions of each gene, so we can use this information as a priori information. In this setting, the model behavior is solely based on this qualitative information which guarantees biologically reasonable behavior: robustness and sparsity in general. The ability of many biological networks to exhibit their function reliably despite noise or perturbation is often referred as functional robustness. We also note that biological regulatory networks are likely to be sparse especially at a fixed interval of time (for example, most transcription factors (TFs) do not regulate most genes). Also, there are expectations behind modeling efforts:

- Networks represent the structure of complex connections so viewing evolving networks as dynamical systems allows us to predict many of their properties analytically.
- If we can match signal propagation that drives the placement of links and nodes, then the topology or the structural elements will follow. It can help to move beyond architecture and uncover the laws that govern the underlying dynamic process.

In contrast to previous methodologies for dynamic graph analysis, we develop a convex optimization-based inference method, where we embed the dynamics of a linear time varying representation, and enforce sparsity at corresponding time intervals. We have applied the method to an example of biological network of HER2 over-expressed breast cancer.

This work has been written up [7] and is submitted to the 2011 American Control Conference.

An ODE Model for the HER2/3-AKT Signaling Pathway in Cancers that Overexpress HER2

During the 2009-2010 period, we designed new algorithms for recovering the structure of signaling pathways by using the power of perturbations. We created three versions of a dynamic model for the HER-AKT pathway in cancers that overexpress HER2. The different versions correspond to different PI3K mutants. The models showed some predictive ability and they inspired a treatment scheme that is being tested.

The Human Epidermal Growth Factor Receptor (HER) family activates signaling pathways which control crucial cellular processes such as cell division, motility and survival, and are among the best studied of all signaling networks]. HER family members are implicated in a wide variety of human cancers and are frequent targets of pharmaceutical intervention. Individual HER family members have diverse roles and may be specifically targeted in cancer therapies.

The HER2 receptor is overexpressed in 20-30% of breast cancers, in addition to some lung and other cancers. HER3 is also frequently overexpressed in cancers that overexpress HER1 or HER2. Increased expression of HER3 synergistically increases the potency of HER2 in the cancer, and the loss of HER3 removes the transforming ability of HER2. HER3 is in fact a necessary partner in HER2-mediated transformation. Additionally, cancers that overexpress HER2 have been found to be more aggressive than similar cancers that don't. Molecularly, activation of HER receptors leads to the activation of tyrosine kinases, enzymes which activate signaling molecules via phosphorylation, enabling propagation of signal along the signaling cascade. One of the targets of the HER family receptors is AKT, a well-known proliferation agent. The inhibition of AKT allows the protein BAD to be activated, and leads to cell apoptosis. This is why the selective inactivation of AKT in cancer cells is a desired therapeutic outcome.

A relatively new class of therapies for cancer is targeted drug treatments. Among these are selective tyrosine kinase inhibitors (TKIs), which have proven more effective than traditional chemotherapy in some cancers. However, their application to HER signaling has demonstrated limited anti-tumor activity, in spite of effective receptor activity inhibition. This inconsistency was resolved in an elegant study by Sergina et al (UCSF), in which it was demonstrated that HER3 and downstream pathway effectors PI(3)K/AKT evade inhibition by current HER-family TKIs due to a compensatory shift in the HER3 phosphorylation-dephosphorylation equilibrium. This shift is driven by increased membrane HER3 expression driving the phosphorylation reaction and by reduced HER3 phosphatase activity impeding the dephosphorylation reaction. These processes are driven by an AKT-mediated negative-feedback loop.

From a different point of view, the problem of creating a drug to treat a certain type of cancer can be thought of as a control engineering problem, in which the goal is to steer the state of a cancerous cells to healthy or dead. This point of view enforces the need for mathematical modeling for the behavior of cancer cells. Recently, a number of dynamical systems models of the HER2/3 pathway in various cell types has been introduced, enabling the formulation of nonintuitive hypotheses and setting the stage for potential novel thera-

peutics. Mathematical models for signaling pathways and cells are becoming more popular, because of the large amounts of data being produced by the molecular biology community and the great promises this point of view holds.

In this project, we use a compendium of known interactions of this pathway, as well as the data from the Sergina et al study, to formulate a dynamic model of HER2 and HER3 signaling in breast cancer. We aim to model the HER2/3 pathway in cancers that overexpress HER2, using data from the BT747 and SKBr3 cell lines. We create a model with ordinary differential equations (ODEs) that uses first and second order mass action kinetics and accounts for vesicular trafficking between cellular compartments. In addition to our tissue focus, our model is distinguished by its relatively large scope in terms of the number of protein species included, as well as the time span (two days, versus a time span of up to 2 hours in previous models). The extended time span is crucial to our model predictions and therapeutic application, as anti-tumor effects fail at longer time spans, even when indications appear favorable in the initial hours post drug application. With a model that accounts for extended time behavior of cancers upon treatment with TKIs, we hope to have an understanding that allows us to design better treatments for cancers that overexpress HER2.

Although the model employs first order approximations and represents a simplification of the underlying biological system, within the constraints of the measured experiment set currently available, it demonstrably captures the system behavior over time, successfully generalizes to predict unseen training data, and enables the formulation of testable hypotheses for therapeutic applications.

The data available to fit the parameters of the model is scarce at this point, and thus it might be expected that a large class of models would fit the data. Of course, this is only true if the network has a valid structure and enough degrees of freedom. Other properties of the system can be studied by making sure that they're robust over the irrelevant parameters. The idea is that if all of the models that fit the data show a certain behavior, the true model could be expected to exhibit that behavior. Additionally, a basic model can be used to determine the experiments needed to refine the model.

This work has been written up and appears in [8].

2.1.5 Embedded Software for National and Homeland Security

Autonomous Ground Vehicles

Ground vehicle research, although more stable than airborne vehicles, presents the subtle problem of providing intelligent behavior which operates in real-time, executes safely, and yet provides a "smooth" reaction to stimuli—i.e., the software behavior is somewhat humanized. Our application is the DARPA Urban Challenge, and we are using the ground vehicle testbed to show the performance of various algorithms and advancements in theory of switched systems, model-predictive control, parameter identification, time-triggered distributed components, and computer vision, as well as how these components work in real-time with one another. Additional concerns which this project addresses are distributed

software testing, component-based design, and vehicle/sensor health monitoring. We are proving many of the theories and algorithms which have been developed in the last four years of the ITR.

Real-time Computer Vision

The goal of this task is to detect moving objects using a stereo camera system mounted on a car. This information will be used to detect, and estimate the trajectories of (possibly) mobile obstacles such as other vehicles. Note that, in this scenario, we wish to detect both objects in close proximity to our vehicle (for example, the vehicle in front of us), and also objects that are farther away (for example, oncoming vehicles), so we cannot make assumptions about whether height errors at adjacent pixels appear to be close to planar. Our solution must also be able to run in real time. In this application, we are interested in determining 3D motion in the scene, rather than the more-studied case of 2D motion in the image. In particular, we want to find the motion of objects in the scene relative to each other; if we have vehicle state data available, or if we can assume that the majority of the scene is not moving, we can determine which motion-segmented region corresponds to the background and remove its perceived motion from the perceived motion of the other objects, obtaining the motion of the other objects in a global frame.

Todd Templeton's focus in the past year has primarily been hardware and software infrastructure for the autonomous / assisted-driving vehicle testbed. This effort has culminated in external time synchronization for all car sensors, which is essential for perception algorithms that utilize multiple sensors, as well as a reusable and extensible library of communication and sensing software components that can run on the variety of hardware platforms carried in the car. In addition to a complement of four half-2U mini-ITX Linux PCs, the car carries Coldfire-based NetBurner embedded processor boards for sensors that do not support external triggering and time-synchronization; the cameras are the only sensors on the car that natively support an external trigger. A hardware interrupt on each microprocessor board is wired to the same (square wave) trigger signal as the cameras, through a solid-state relay for voltage conversion. The master external trigger (generated by a network-controllable signal generator in the car) is enabled after all of the microprocessor boards are powered on, and after the camera driver has initialized the cameras. The first falling edge of the trigger signal causes the first frame to be captured by each camera, and also provides a time-zero reference to the microprocessor boards. Subsequent falling edges of the trigger signal cause subsequent frames to be captured by the cameras, and also provide a time reference from which the microprocessor boards calibrate their internal clocks. Once the master trigger is enabled and a microprocessor board has performed the initial calibration of its clock (after receiving the second falling edge), it begins time-stamping data from the sensor (such as an INS or LADAR) to which it is attached and sending it over the car's Ethernet network using the Spread multicast messaging service. All perception algorithms, which run on the Linux PCs, use the time-stamp on each piece of sensor data, instead of the current system time, in their calculations.

The embedded Coldfire processors in the car present a challenge to software portability. To this end, Templeton has developed an architecture compatibility library that emulates the

required subset of the POSIX C standard library on top of the embedded OS, embedded C library, and NetBurner system libraries. This compatibility library also provides standard I/O over a network socket, and functionality that runs before the program's main() to initialize the system clock against the external trigger.

The software collection for this platform, hopefully to be expanded to other robotic platforms within the group and open-sourced to the larger robotics community, is called the Intelligent Robotics Toolkit (IRT). It is primarily based on software used in a previous vision-based helicopter mapping and landing project, expanded to support distributed computing across a diverse hardware and software environment. We currently have a software engineering doctoral student from RWTH Aachen University in Germany helping us to strengthen the software engineering foundations of this toolkit, and to expand its simulation and visualization capabilities.

Recently, we have achieved time synchronization / triggering across all car sensors, and taken the car out on the roads near Richmond Field Station to capture sensor data from which to formulate and tune perception algorithms.

Templeton graduated in December 2009: in his final contributions, he focused on the problem of moving object segmentation while driving, using camera data collected from the above-mentioned data-gathering trip. Moving object detection and segmentation is important for both autonomous and assisted driving for two primary reasons: to estimate the trajectories and future positions of mobile obstacles, and to remove points on mobile obstacles from the map of the static portion of the scene (without which mobile obstacles become 3D walls as they move across the scene). Hence, the motion planning process becomes similar to that in a static environment (using the static scene map), with the addition of a list of mobile obstacles and their estimated trajectories and current positions.

The proposed motion segmentation algorithm is non-parametric in that it does not assume a particular shape or appearance of mobile obstacles, which allows it to detect moving obstacles such as bicyclists and pedestrians, and to not detect stationary objects that look like cars (such as parked cars)—these static objects will be handled like any other static part of the scene by the motion planner.

Verification of Driver Augmentation Systems

There has been an explosive growth in the use of embedded systems in cars. By some accounts, it is widely expected that by the end of the decade over 50% of the cost of a car will be vehicular electronics (or veitronics). Beyond the fundamental functionality of an automobile, such as driving, stopping, and turning, a modern car provides additional features for more passenger safety, better comfort, and lower environmental impact. Thus, the number of embedded processors used in a vehicle today is in the order of 80 for luxury cars, and is expected to grow further by some accounts. A contemporary car therefore is a networked embedded system, in which subsystems need to process data and communicate over special networks such as CAN, LIN or Flexray, often within hard real-time constraints. While there has been a great deal of attention paid to hybrid and electric cars and the veitronics for their drive trains, it is our contention that Advanced Driving Assistance Systems (denoted as ADAS) present a big opportunity to apply research that can potentially

have a big impact on the whole automotive industry. A key issue with the introduction of these safety critical technologies is the need to have them be verified, validated and in some cases certified. By verification, we mean formal results guaranteeing the performance (properties such as safety, liveness or non-blocking stability) of models of intelligent software systems embedded with the physical hardware on a car; and by validation we mean testing of the theoretical verified proofs of performance on hardware. Certification usually follows formal specifications laid down by regulatory/ insurance authorities and is accompanied by verification/ validation.

Automotive OEMs and tier 1 suppliers are currently suffering from the high cost of verification and validation of the kinds of features that are being demanded by customers. While the software itself to be used in automotive systems has to undergo rigorous processes for design, implementation, audit and test, using standards such as IEEE-610 or SEI CMMI, limited tools exist for verification of automotive ADAS. These systems provide an additional order of complexity, since they are not only semi-autonomous, but also interaction with a human operator for safety critical decisions is necessary.

The state-of-the art to the extend of our knowledge is a statistical approach: High-precision driving robots or automated mini-series cars such as the VW Golf GTi "53 plus 1" are used to rigorously repeat driving situations within sub-centimeter precision. If statistical measures such as MTTF or MTBF suggest safe operation, the system is deemed deployable in road traffic.

However, there are several problems with such an approach: Foremost, repeated execution with varying initialization parameters is only feasible for limited cases, such as a Parking Assistant. Further, the testing has to be rigorous and thus is costly in monetary terms and drastically increases time-to-market.

It is therefore our strong belief, that a formal verification method for ADAS is needed, to increase car functionality, safety and energy efficiency, as well as decrease development costs and time to market. We are thus developing a new theoretical framework and algorithms for the design of mathematically verified and validated cyber-human systems from beginning with ADAS as a specific case. We are exploring the next generation of cyber-human systems, which will employ much "better models of human cognition", and hence, better assist humans in autonomous or semi-autonomous ways. This research is enabled by recent major advances in the sub-areas of computing and communication, such as stochastic hybrid models, learning methods, signal-to-symbol transformation, distributed decision making and dynamic resource allocation in geographically distributed systems without communications infrastructure.

Over the last 40 years, the fields of knowledge representation, perception, robotics, control, and learning have evolved in their separate ways. The grand vision of cyber-human systems, in which these fields combine into complete agents, has all but disappeared. However, with new results and partial reunifications of these individual sub- disciplines, the time has come to propose a reconstruction of the science of cyber-human systems. The focus of the present research is developing integrated agent designs capable of performing simple tasks reliably in unstructured environments and generating purposive activity over an unbounded period.

Such a goal requires dealing with perceptual input, noisy, and partially known dynamics, real-time requirements, and complex environments containing many objects and agents.
The goal is to demonstrate our work on two examples: Automatic Parking and Highway Cruise Control with linear and lateral track, as well as negative and positive acceleration, authority. We chose the parking example, since several luxury cars already deploy such systems, and our approach can readily be compared with existing solutions. The highway example, however, is on the road map of all major OEMs, yet we argue that existing verification approaches will fail here. This is where our major contribution will be.
Automated Driver Automation Systems (ADAS) are required to perform safety-critical tasks with hard-real time guarantees. Simulation and validation based methods alone cannot guarantee that all specifications are met and that undesired behaviors are not executed. Formal methods offer a rigorous framework to prove that a mathematical abstraction of embedded software satisfies the required correctness properties. The mathematical abstraction of ADAS embedded software can be completely behavioral in that it closely describes all possible executions or it could be simply prescriptive in that it only specifies what the system should do. In both cases, the mathematical abstraction should specify evolution of the physical environment of the system such as the effect of forces, actions of the driver, presence of external entities like obstructions, other vehicles etc. In addition, the ADAS models should also include description of maneuvers such as cruise mode, lane changing, parking, and overtaking. The specification of required correctness properties can be generally translated into safety, i.e., the system never performs a bad execution; and liveness, i.e., the system eventually performs a good execution and does not deadlock.
The modeling of ADAS requires hierarchies of different models of computation, some discrete and others continuous. Such system models are referred to as hybrid system models. These models are especially suited to modeling compound behaviors arising from composition and interaction between heterogeneous sub-systems in automotive systems. A prototypical example is the hierarchical layering of finite state machines and nonlinear continuous time differential equations.
However, modeling uncertainty associated with human behavior in verification of ADAS raises new issues and challenges: First, the human-ADAS interface determines the mode and extent of information the user has about the behavior of the system. Secondly, the modes of interaction that are not proven to be correct can raise important safety issues of such systems. In particular, the human-machine interaction can lead to unpredictable behaviors if the assumption of human user does not match with the guarantee that the ADAS can provide. As an example, if the human user gives control of a safety- critical task to the ADAS but assumes a wrong model of the system's behavior, the ADAS may not be able to successfully execute the task. Lastly, such systems also raise an important modeling question: Can human cognitive limits be reasonably modeled and incorporated in ADAS verification framework? We plan to address these research challenges by formalizing probabilistic specifications of human-ADAS interaction. We are inspired by developments in modern cognitive science which holds the viewpoint that human mind is a computational system. Prior research has also demonstrated that significant headway can be made in

understanding how a human's cognitive resources can be integrated most effectively into the problem when mental models of the human, and information about the human's cognitive limits, are incorporated into the design of the control architecture in the very early stages of design specification. We will build a framework in which successive refinements of models of human cognitive functions will be checked for correctness against formal models of ADAS. Our research will benefit from the efforts to model interaction between human operators and air- traffic management systems by accounting for the physical constraints that come from applications. Horvitz at Microsoft Research has pioneered the use of Bayesian networks in the design of models of human interaction with automation: NASA operators with varying levels of expertise, interacting with the fuel control systems on board the Space Shuttle. Using these models to indicate the context and timing of the information to display to the operator, the expected utility of the operator's decisions can be greatly enhanced.

In our approach to verification, we use a multi-world semantics hierarchical system. We break up the verification task into individual components, and verify each in its own, and its interface to neighboring components in the hierarchy. An expression at each level is interpreted at the same level. Therefore, checking the truth-claim at that level is performed in its own declarative world. Relating these disconnected worlds together is a nontrivial task; however, we are collaborating with Jonathan Sprinkle from University of Arizona on using metamodels for the interface design between components. This both enables better analyzability as well as potential for using components developed by different research groups. Instead of semantic flattening, where an expression has to be both syntactically translated and semantic interpreted, we are following human reasoning and cognition, and propose the use of high-level expressions, which will be compiled into idealized lower-level expressions and then interpreted. Invariants will be used to show that higher-level truth-claims now become conditional lower level truth- claims. In this setting, higher-level truth-claims become necessary conditions. This is contrary to one-world semantics, in which higher- level claims are sufficient conditions.

Suppose one-world interpretation leads to falsification of higher- level claims that are true in multi-world semantics. This can happen, when lower-level faults are not accounted for in higher-level descriptions. Then, multi-world semantics must be split into multiple- frameworks, each dealing with the identified faults.

Our proposed framework is kept general enough to apply to various systems. From the sensory point-of-view, these levels have increasingly coarse representations of the world as one moves toward higher levels, while from the actuation point-of-view, the tasks become more strategic, supervisory and planning-oriented at the higher levels starting from "reactive" ones at lower levels.

The highest level, the Mission Planner, is the human operator in the automotive context. The output of this level could be in the form of "go to San Francisco". The Route Planner then takes this input and transforms it into an actual plan, in our context it would be an A* implementation detailing which roads to take. Next, the Maneuver Control layer takes these roads and transforms them into driving decisions, considering current sensor readings. From there, the Low- level Control transforms these driving decisions into actual control input for

the physical vehicle.
This model also has what we call in analogy to database design a "roll- back mechanism": If a component cannot perform a task given from a higher-level component, it has to communicate the reasons why it failed at execution. The higher level then has to roll-back: to revert to a more feasible strategy or pre-defined back-up plan.
There are several challenges associated with the verification of such hierarchical architecture. One has to establish such a hierarchy in a formal way with different models appropriate at the respective levels: define the semantics of interaction between different levels for operational and emergency goals, define assumptions about the model of interactions between the levels, the levels of service each level can provide, and define the set of services each level must accept in order to provide requisite services.
Within this hierarchy, we are also interested in knowing how uncertainty and fault propagates between different levels and their criticality to the goals.
Our mathematical framework for the hierarchy is currently under constant change. We defined a series of increasingly complex examples of (semi)-autonomous system that we try to capture in our framework, and apply our framework to these. If properties of the system cannot be described with the current definitions, these are altered or amended. Our examples range from the common water tank problem in hybrid systems theory via agents that have two simultaneous missions or objectives such as to traverse an area and to remain stealthy, to a collection of heterogeneous agents which each have unique sets of capabilities and have to work together towards a goal declared by the Mission Planner.
We are about to finalize this framework and intend to publish a detailed white paper in the upcoming Fall on our findings to invite feedback from the research community.
While the research effort described above is mainly aimed at verifying a (Semi-) autonomous system, an ADAS capable of making autonomous driving decisions also has to incorporate knowledge and assumptions about other traffic participants. We therefore propose the investigation into using an Algorithmic Game Theory (AGT) approach. While game theory itself is only descriptive, AGT however is prescriptive, and therefore could be used in this effort.
Clearly, interaction of cars (normally) cannot be described as an adversial game. However, it is also not really a collaborative effort, since one driver has no incentive to help another driver to achieve his goal. We are thus modeling traffic as an implicit collaboration:
Ideally, human drivers not only care for their own safety and goals, but also attempt to not inconvenience other traffic participants. Recent research conducted by our partners from the University of Aachen in co-operation with Volkswagen suggests that it suffices to use a 3x3 or 3x4 matrix to make informed LTT driving decisions. The car being the center in the matrix (X_{00} in its local neighborhood can be divided into segments to its sides, in front and behind, and on the diagonals. If there is any car detected in a segment, it's estimated speed is entered into the matrix. In the case of a sensor that can see further ahead than up to the next car (such as
Same advanced automotive radar), one column in driving direction extends the matrix. Each cell of the matrix contains the speed and the distance of a car, or Null if there is no car

detected.
To model the interaction between cars, we are using coordination games (CG). Coordination games model situations in which all agents can realize mutual gains as long as they make mutually consistent decisions. For such games, there typically exist several pure strategy Nash Equilibria, so that the agents have to chose the same or corresponding strategies. However, often the coordination has to be done without any ability to communicate. Finally, some equilibria may give higher payoffs, be more fair, or, most important to the application at hand, may be safer, thus occasionally leading to interest conflicts.
We argue that this fits the interaction of cars on a highway well. We do not want to force cars to be able to communicate with each other, for two major reasons: security and deployment. Communication between cars would have to be done wirelessly, and thus provides an easy point of attack for malicious parties. Also, for such a system, all traffic participants would have to be equipped with such communication facilities, rendering it unusable for the next decade. Further, just for the interaction of two cars and one possible disturbance, we can foresee several equilibria, and cars should settle for a common equilibrium that maximizes overall welfare.
In parallel to our verification efforts, we also created the Berkeley DRIVR Lab, in close collaboration with CHESS, with the intend to showcase successfully verified algorithms. After participating in the DARPA Urban Challenge, we published a paper in AAET 2008, outlining our ideas on how to apply lessons learned from autonomous driving towards ADAS. Discussions about our paper with other researchers in the field and OEMs led to our insight on the necessity of formal ADAS verification techniques. However, other lessons learned were on the need for a robust and scalable hardware platform, as well as test bed, and a lightweight software toolkit.
We have outfitted a fully automated 2008 Ford Escape Hybrid ByWire XGV as testbed for (semi-) autonomous driving Actuation is done transparently via a commercial toolkit (Torc Technologies) that directly communicates with the car's ECU. A cluster of customized quad-core mini-ITX computers is used to process data and make driving decisions, communicating among each other and with the car via Gigabit Ethernet. Various built-in screens and a KVM switch, as well as the possibility to connect notebook computers enables in-car debugging, while wireless network access enabled remote debugging, data streaming and offline analysis. All sensors, such as visual light cameras, thermal infrared cameras, laser scanners, radar, inertial measurement units and GPS are connected to time-synchronized embedded processors which trigger synchronized data-acquisition and timestamp data before relaying it to the computation cluster via Gigabit Ethernet for pre-processing, fusion and influencing driving decision making.
Since there is no robotics software framework available that satisfies all our demands for robustness, ease of use, reliability, platform independence, lightweight communication and interfaces to existing standards such as JAUS among others, we opted for implementation of our own.
Our software and computational approach is based on the following principles:

- Sensor drivers and communication methods must be abstracted as much as possible, to

enable maximum flexibility in choosing different sensors and communication methods in the future.

- Transparent timestamping and synchronization facilities must be available at the sensor input level. To enable this, sensor drivers must be lightweight enough to run on an embedded processor; in particular, this means minimal or no threading.
- Transparent multicast, logging, and replaying facilities must be available at all levels (including the sensor-input level), in order to encourage the modularization of software, and to enable the independent development and verification of individual modules.
- Software modules must be able to be written in a variety of languages and run on a variety of platforms, with minimal invasiveness on the part of the infrastructure.
- Software modules should make optimal use of the computing hardware, from embedded processors via multi-core CPUs to manycore GPUS by utilizing frameworks such as Intel Thread Building Blocks, OpenMP and OpenCL or CUDA.
- Although the infrastructure is written in C/C++ due to its efficiency and ease of use, it is written with multi-platform compatibility in mind, and its interface is simple and can be easily wrapped for other programming languages, such as MATLAB.
- Where available, existing software components will be used, such as scientific and numerical libraries (BLAS, LAPACK), or vision libraries such OpenCV or the NASA Vision Workbench.

2.1.6 Control of Communication Networks

In a series of papers Abate and co-authors have continued to explore 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 (see).

2.2 ProjectFindings

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 4 below, including publications representing findings that were reported in the previous annual report.

- [5]Milosh Drezgich, S. Shankar Sastry. 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, "Matrix Multiplicative Weights and Non-Zero Sum Games," 2010; Submitted.

  This article concentrates on the improvements in the vector and matrix version of a widely used multiplicative weights algorithm .The four results that we present are the following. We first present a small improvement in the existing upper bound for the matrix multiplicative weights algorithm for the particular set of two player zero sum games. Second, using the concavity property of unital positive linear maps between *-algebras, we establish the precise connection between the vector and matrix version of the multiplicative updates algorithm, that clearly reveals why the total loss for the both algorithms are the same. Third, as a corollary we present the matrix multiplicative weights algorithm with the iterative updates, that is both computationally less demanding and guarantees the same worst case performance as the conventional cumulative matrix multiplicative weight algorithm, resolving in part the open question stated at COLT 2010. Finally, unlike the vector version of the multiplicative weights algorithm, we present an evidence that the Nash equilibrium strategy for the non-zero sum Shapely game and the augmented Shapely game, can be found using matrix multiplicative weights updates algorithms.

- [3]Anil Aswani, Claire Tomlin. Unpublished article, "Computer-Aided Drug Discovery for Pathway and Genetic Diseases," 2010.

  Selecting drug targets in pathway and genetic diseases (e.g., cancer) is a difficult problem facing the medical field and pharmaceutical industry. Because of the complex interconnections and feedback found in biological pathways, it is difficult to understand the potential effects of targeting certain portions of the network. The pharmaceutical industry has avoided novel targets for drugs, largely because of the increased risk in developing such treatments. This necessitates the need for systems biology methods which can help mitigate some of the risks of identifying novel targets and also suggest further experiments to validate them. The primary goal of this paper is to introduce a mathematical framework for solving such problems, that is amenable to computational or mathematical study. The secondary goal is to suggest methods for solving problems posed in this framework. One of these methods is a heuristic which is designed to allow its computations to scale up to much bigger examples and pathways than presented here.

- [9]Milosh Drezgich, S. Shankar Sastry. Unpublished article, "On the NP in BQP," 2010.

One of the central questions of the theory of quantum computation is whether the quantum computers are able to solve, in polynomial time, the problems that are classically currently believed intractable, for example, unstructured search and 3SAT. While mostly believed unlikely, almost all current efforts toward the positive result have been utilizing the quantum adiabatic theorem in the design of the potential algorithm. We show that if the process of computation is governed by the system Hamiltonian, design of the prospective algorithm should not rely on the premise of adiabatic or ground state computation. In particular we prove that the prospective algorithm, represented by the dynamic system through the Schroedinger equation, does not fail simply because the system evolution did not satisfy adiabatic condition or drifted away from the ground state. As a corollary we both present the algorithm in $O(n^k)$ for the unique unstructured search and make conjecture on the algorithm for 3SAT.

- [10]J. Biermeyer, T. Templeton, C. Berger, H. Gonzalez, N. Naikal, B. Rumpe, S. S. Sastry. Proceedings des 11. Braunschweiger Symposiums "Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme fr Transportmittel", ITS Niedersachsen, Braunschweig, "Rapid Integration and Calibration of New Sensors Using the Berkeley Aachen Robotics Toolkit (BART)," 2010.

  After the three DARPA Grand Challenge contests many groups around the world have continued to actively research and work toward an autonomous vehicle capable of accomplishing a mission in a given context (e.g. desert, city) while following a set of prescribed rules, but none has been completely successful in uncontrolled environments, a task that many people trivially fulfill every day. We believe that, together with improving the sensors used in cars and the artificial intelligence algorithms used to process the information, the community should focus on the systems engineering aspects of the problem, i.e. the limitations of the car (in terms of space, power, or heat dissipation) and the limitations of the software development cycle. This paper explores these issues and our experiences overcoming them.

- [1]Anil Aswani, Peter Bickel, Claire Tomlin. Annals of Statistics, "Regression on Manifolds: Estimation of the Exterior Derivative," 2010; To appear.

  Collinearity and near-collinearity of predictors cause difficulties when doing nonparametric regression on manifolds. In such scenar ios, variable selection becomes untenable because of mathematical difficulties concerning the existence and numerical stability of the regression coefficients. In addition, once computed, the regression coefficients are difficult to interpret, because a gradient does not exist for functions on manifolds. Fortunately, there is an extension of the gradient to functions on manifolds; this extension is known as the exterior derivative of a function. It is the natural quantity to estimate, because it is a mathematically well-defined quantity with a geometrical interpretation. We propose a set of novel estimators using a regularization scheme for the regression problem which considers the geometrical intuition of the exterior derivative. The advantage of this regularization scheme is that it allows us to add lasso-type

regularization to the regression problem, which enables lasso-type regressions in the presence of collinearities. Finally, we consider the large p, small n problem in our context and show the consistency and variable selection abilities of our estimators.

- [11]Anil Aswani, Harendra Guturu, Claire Tomlin. "System identication of hunchback protein patterning in early Drosophila embryogenesis," Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, 7723-7728, 18, December, 2009.

  Early patterning in the Drosophila melanogaster embryo occurs through a complicated network of interactions involving transcription factor proteins and the mRNA of their target genes. One such system is the pattern of hunchback mRNA and its regulation by Bicoid and Kruppel proteins. This system is well-studied, but there is disagreement amongst biologists on how exactly hunchback expression is regulated. We attempt here to provide evidence to distinguish between two models in contention, through system identification. Our general approach is to do nonlinear regression on a parametric, nonlinear partial differential equation model which incorporates transcription, diffusion, and degradation. We perform the regression, analyze the results and then interpret the results in the biological context. We also compare our results to previous work on this system.

- [12]Todd Templeton. PhD thesis, "Accurate Real-Time Reconstruction of Distant Scenes Using Computer Vision: The Recursive Multi-Frame Planar Parallax Algorithm," University of California, Berkeley, December, 2009.

  In this dissertation, we detail the Recursive Multi-Frame Planar Parallax (RMFPP) algorithm, a recursive extension of Irani et al.'s Multi-Frame Planar Parallax (MFPP) batch algorithm that allows real-time reconstruction of distant static scenes using computer vision, with expected error that increases only linearly with depth. We present an overview and comprehensive derivation of the theoretical foundation on which the RMFPP algorithm is built, including the seminal planar-parallax work by Sawhney. We derive a recursive cost function that preserves more of the problem's nonlinearity than does the cost function in the MFPP algorithm, which allows a more accurate recursive procedure. In order to obtain a recursive algorithm, we remove the geometry-refining optimization that is present in the MFPP algorithm; however, we empirically show that our algorithm degrades gracefully in the presence of geometric error. We present results using both synthetic and real imagery that show that the RMFPP algorithm is at least as accurate as the original MFPP batch algorithm in many circumstances, is preferred to both fixed- and dynamic baseline two-frame methods, and is suitable for real-time use.

- [13]YuLun Huang, Alvaro Cardenas, Saurabh Amin, Song-Zyun Lin , Hsin-Yi Tsai, S. Shankar Sastry. International Journal of Critical Infrastructure Protection, "Understanding the Physical and Economic Consequences of Attacks Against Control Systems.," 2(3):72-83, October 2009.

This paper describes an approach for developing threat models for attacks on control systems. These models are useful for analyzing the actions taken by an attacker who gains access to control system assets and for evaluating the effects of the attacker's actions on the physical process being controlled. The paper proposes models for integrity attacks and denial-of-service (DoS) attacks, and evaluates the physical and economic consequences of the attacks on a chemical reactor system. The analysis reveals two important points. First, a DoS attack does not have a significant effect when the reactor is in the steady state; however, combining the DoS attaack with a relatively innocuous integrity attack rapidly causes the reactor to move to an unsafe state. Second, an attack that seeks to increase the operational cost of the chemical reactor involves a radically different strategy than an attack on plant safety (i.e., one that seeks to shut down the reactor or cause an explosion).

- [14]Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, S. Shankar Sastry. "Challenges for Securing Cyber Physical Systems," Workshop on Future Directions in Cyber-physical Systems Security, DHS, 23, July, 2009.

  We discuss three key challenges for securing cyberphysical systems: (1) understanding the threats, and possible consequences of attacks, (2) identifying the unique properties of cyber-physical systems and their differences from traditional IT security, and (3) discussing security mechanisms applicable to cyber-physical systems. In particular, we analyze security mechanisms for: prevention, detection and recovery, resilience and deterrence of attacks.

- [4]Anil Aswani, Nicholas Boyd, Claire Tomlin. "Graph-theoretic topological control of biological genetic networks," ACC 2009, 1700-1705, 10, June, 2009.

  The control of biological genetic networks is an important problem. If the system is abstracted into a graph, then the affect of drugs, pharmaceuticals, and gene therapy can be abstracted as changing the topology of the graph. We consider the control objective of removing the stable oscillations of the genetic network. This control is done using several theorems relating the topology of the network to the dynamics of the system. These theorems suggest that the controller should remove all the negative feedback in the networks.We prove that the problem of minimizing the edges and vertices to remove, in order to remove negative feedback, is NP-hard. In light of this result, a heuristic algorithm to solve this graph problem is presented. The algorithm is applied to several genetic networks, and it is shown that the heuristic gives reasonable results. Additionally, we consider the p53 network and show that the algorithm gives biologically relevant results.

- [15]Zong-Syun Lin, Alvaro Cardenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, S. Shankar Sastry. "Model-Based Detection of Attacks for Process Control Systems," 16th ACM Computer and Communications Security Conference, ACM, submitted, 2009.

We present security analysis of process control systems (PCS) when an attacker can compromise sensor measurements that are critical for maintaining the operational goals. We present the general sensor attack model that can represent a wide variety of DoS and deception attacks. By taking example of a well studied process control system, we discuss the consequences of sensor attacks on the performance of the system and important implications for designing defense actions. We develop model-based detection methods that can be tuned to limit the false-alarm rates while detecting a large class of sensor attacks. From the attacker's viewpoint, we show that when the detection mechanisms and control system operations are understood by the attacker, it can carry stealth attacks that maximize the chance of missed detection. From the defender's viewpoint, we show that when an attack is detected, the use of model-based outputs maintains safety under compromised sensor measurements.

3 Outreach

3.1 Project Training and Development

3.2 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.

3.2.1 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.

3.2.2 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.

Course: Introduction to Embedded Systems (UCB EECS 149)
http://chess.eecs.berkeley.edu/eecs149

Instructors:
Prof. Edward A. Lee
Prof. Sanjit A. Seshia
Prof. Claire J. Tomlin

During the previous reporting period, Professor Tomlin assisted in development of the undergraduate Introduction to Embedded Systems course, EECS 124. For the Spring 2009 semester, the course was renumbered to EECS 149. The new material was taught in the Spring Semester of 2010 by Professor Sanjit A. Seshia.

The abstract for the class is below:

EECS 149 introduces students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.

A major theme of this course will be on the interplay of practical design with models of systems, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

This course is still under active development. This offering is therefore advised for advanced and adventurous undergraduates.

Course: Introduction to Control Design Techniques (UCB EECS 128)
http://inst.eecs.berkeley.edu/ ee128/fa08/

Instructor:
Prof. Claire J. Tomlin

During the previous reporting period Professor Tomlin has redesigned the undergrad-

uate control theory and engineering course, EECS 128, adding new labs and course material. The new material was taught in the Fall Semesters of 2008 and 2009.
The abstract for the class is below:
Root-locus and frequency response techniques for control system synthesis. State-space techniques for modeling, full-state feedback regulator design, pole placement, and observer design. Combined observer and regulator design. Lab experiments on computers connected to mechanical systems.

- Transfer function and state space models for control system analysis and synthesis. Pole locations and relationship to time response. Root locus methods. Stability.
- Feedback. Review of single-input single output (SISO) analysis and control methods in the frequency domain (Bode, Nyquist).
- SISO analysis and control using state space models. The matrix exponential and its relationship to time response. Controllability and observability. Combining state feedback with observers.
- Multi-input multi-output analysis and control using state space models.
- The linear quadratic regulator.

3.2.3 Graduate Courses

As part of the no-cost extension, a course in embedded systems was taught in the area of embedded and hybrid systems, as well as systems modeling. This course is a reflection of the teaching and curriculum goals of the ITR and its affiliated faculty.
Course: Linear System Theory(UCB EE221A)
http://inst.eecs.berkeley.edu/ ee221A/fa09/
Instructor: Claire J. Tomlin

Professor Tomlin is modernizing the graduate course in linear system theory, EECS 221A, adding units in linear programming and more general optimization. The new material was taught in the Fall Semesters of 2008, 2009 and now in 2010.
The abstract for the class is below:
This course provides a comprehensive introduction to the modeling, analysis, and control of linear dynamical systems. Topics include: A review of linear algebra and matrix theory. The solutions of linear equations. Least-squares approximation and linear programming. Linear ordinary differential equations: existence and uniqueness of solutions, the state-transition matrix and matrix exponential. Input-output and internal stability; the method of Lyapunov. Controllability and observability; basic realization theory. Control and observer design: pole placement, state estimation. Linear quadratic optimal control: Riccati equation and properties of the LQ regulator.

Advanced topics such as robust control and hybrid system theory will be presented based on allowable time and interest from the class.
This course provides a solid foundation for students doing research that requires the design and use of dynamic models. Students in control, circuits, signal processing, communications and networking are encouraged to take this course.

- Linear Algebra: Fields, vector spaces, subspaces, bases, dimension, range and Null spaces, linear operators, norms, inner products, adjoints.
- Matrix Theory: Eigenspaces, Jordan form, Hermitian forms, positive definiteness, singular value decomposition, functions of matrices, spectral mapping theorem, computational aspects.
- Optimization: Linear equations, least-squares approximation, linear programming.
- Differential Equations: existence and uniqueness of solutions, Lipschitz continuity, linear ordinary differential equations, the notion of state, the state-transition matrix.
- Stability: Internal stability, input-output stability, the method of Lyapunov.
- Linear Systems - open-loop aspects: controllability and observability, duality, canonical forms, the Kalman decomposition, realization theory, minimal realizations.
- Linear systems - feedback aspects: pole placement, stabilizability and detectability, observers, state estimation, the separation principle.
- Linear quadratic optimal control: least-squares control and estimation, Riccati equations, properties of the LQ regulator.
- Advanced topics: robust control, hybrid systems.

4 Publications and Products

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

4.1 Technical reports

- None.

4.2 Software

- [16]Anil Aswani. "NODE model for eve mRNA modeling in Drosophila melanogaster," University of California, Berkeley, 2010.
- [17]Anil Aswani. "NEDE Estimator," University of California, Berkeley, 26, April, 2010.

4.3 PhD theses

- [12]Todd Templeton. PhD thesis, "Accurate Real-Time Reconstruction of Distant Scenes Using Computer Vision: The Recursive Multi-Frame Planar Parallax Algorithm," University of California, Berkeley, December, 2009.

4.4 Conference papers

- [14]Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, S. Shankar Sastry. "Challenges for Securing Cyber Physical Systems," Workshop on Future Directions in Cyber-physical Systems Security, DHS, 23, July, 2009.
- [4]Anil Aswani, Nicholas Boyd, Claire Tomlin. "Graph-theoretic topological control of biological genetic networks," ACC 2009, 17001705, 10, June, 2009.
- [15]Zong-Syun Lin, Alvaro Cardenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, S. Shankar Sastry. "Model-Based Detection of Attacks for Process Control Systems," 16th ACM Computer and Communications Security Conference, ACM, submitted, 2009.
- [11]Anil Aswani, Harendra Guturu, Claire Tomlin. "System identication of hunchback protein patterning in early Drosophila embryogenesis," Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, 7723-7728, 18, December, 2009.
- [4]Anil Aswani, Nicholas Boyd, Claire Tomlin. "Graph-theoretic topological control of biological genetic networks," ACC 2009, 17001705, 10, June, 2009.

4.5 Books

- None.

4.6 Journal articles

- [13]YuLun Huang, Alvaro Cardenas, Saurabh Amin, Song-Zyun Lin , Hsin-Yi Tsai, S. Shankar Sastry. International Journal of Critical Infrastructure Protection, "Understanding the Physical and Economic Consequences of Attacks Against Control Systems.," 2(3):72-83, October 2009.

- [10]J. Biermeyer, T. Templeton, C. Berger, H. Gonzalez, N. Naikal, B. Rumpe, S. S. Sastry. Proceedings des 11. Braunschweiger Symposiums "Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme fr Transportmittel", ITS Niedersachsen, Braunschweig, "Rapid Integration and Calibration of New Sensors Using the Berkeley Aachen Robotics Toolkit (BART)," 2010.

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.

4.6.1 The 2009-2010 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 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

- "Bipedal Robotic Walking: Motivating the Study of Hybrid Phenomena" Professor Aaron D. Ames, Texas A&M University, May, 4, 2010.

- "Toward a Theory of Secure Networked Control Systems"
  Saurabh Amin, UC Berkeley, April 27, 2010.

- "Saving Energy by Reducing Traffic Flow Instabilities"
  Berthold K. Horn, MIT, April 26, 2010.

- "Planning and Learning in Information Space"
  Nicholas Roy, MIT, April 20, 2010.

- "SILVER: Synthesis Using Integrated Learning and Verification"
  Susmit Jha, University of California Berkeley, April 08, 2010.

- "Advances in Embedded Software Synthesis from Dataflow Models"
  Soheil Ghiasi, University of California Davis, April 06, 2010.
- "A Model-Based End-to-End Tool Chain for the Probabilistic Analysis of Complex Systems"
  Alessandro Pinto, United Technologies Research Center , March 30, 2010.
- "An Algebra of Pareto Points"
  Marc Geilen, Eindhoven University of Technology, March 16, 2010.
- "Robust Uncertainty Management Methods Applied to UAV Search and Tracking Problems"
  Andrzej Banaszuk, United Technologies Research Center, March 09, 2010.
- "Embedded Convex Optimization"
  Jacob Mattingley, Stanford University, March 02, 2010.
- "Software Failure Avoidance Using Discrete Control Theory"
  Terence Kelly & Yin Wang, HP Labs, February 23, 2010.
- "Simulation and Hybrid Control of Robotic Marionettes"
  Professor Todd Murphey, Northwestern University, February 16, 2010.
- "A Natural-Language-Based Approach to System ModelingFrom the Use of Numbers to the Use of Word"
  Professor Lotfi A. Zadeh, UC Berkeley, February 09, 2010.
- "On Relational Interfaces"
  Stavros Tripakis, UC Berkeley, February 02, 2010.
- "Modeling, Planning, and Control for Robot-Assisted Medical Interventions"
  Allison Okamura, Johns Hopkins, January 28, 2010.
- "Residential Load Management using Autonomous Auctions"
  William Burke, UC Berkeley, January 19, 2010.
- "From High-Level Component-Based Models to Distributed Implementations"
  Borzoo Bonakdarpour, Verimag Laboratory, January 12, 2010.
- "Extending Ptolemy to Support Software Model Variant and Configuration Management"
  Charles Shelton, Robert Bosch Research and Technology Center, December 15, 2009.
- "Active Traffic Management using Aurora Road Network Modeler"
  Alex A. Kurzhanskiy, University of California at Berkeley, December 08, 2009.

- "Distributed Coverage Control with On-Line Learning"
  Mac Schwager, Computer Science and Artificial Intelligence Lab, MIT, December 1, 2009.

- "Mobile Floating Sensor Network Placement using the Saint-Venant 1D Equation"
  Andrew Tinka, University of California Berkeley, November 24, 2009.

- "High-Level Tasks to Correct Low-Level Robot Control"
  Hadas Kress-Gazit, Cornell University, November 17, 2009.

- "Mobile Device Insights - Virtualization and Device Interoperability"
  Jorg Brakensiek, Nokia Research Center, November 10, 2009.

- "Understanding the Genome of Data Centers"
  Jie Liu, Microsoft Research, November 03, 2009.

- "Robust Control via Sign Definite Decomposition"
  Shankar P. Bhattacharyya, Department of Electrical Engineering, Texas A&M University, October 29, 2009.

- "Mathematical Equations as Executable Models of Mechanical Systems"
  Walid Taha, Rice University, October 22, 2009.

- "Data-Driven Modeling of Dynamical Systems: Optimal Excitation Signals and Structured Systems"
  Bo Wahlberg, Automatic Control Lab and ACCESS, KTH, Stockholm, Sweden , October 13, 2009.

- "Simulating Print Service Provider Using Ptolemy II"
  Jun Zeng, Hewlett-Packard Laboratories, October 6, 2009.

- "Robust Distributed Task Planning for Networked Agents"
  Han-Lim Choi, Massachusetts Institute of Technology, September 29, 2009.

- "User-generated 3-D Content: Personalizing Immersive Connected Experiences"
  Yimin Zhang, Intel Labs China, September 25, 2009.

- "The Relation of Spike Timing to Large-Scale LFP Patterns"
  Ryan Canolty, University of California, Berkeley, September 22, 2009.

- "Concurrency and Scalability versus Fragmentation and Compaction with Compact-fit"
  Hannes Payer, University of Salzburg, September 15, 2009.

- "Avoiding Unbounded Priority Inversion in Barrier Protocols Using Gang Priority Management"
  Harald Roeck, University of Salzburg, September 15, 2009.

- "The Design, Modeling, and Control of Mechatronic Systems with Emphasis on Betterment of Quality of Human Life"
  Kyoungchul Kong, University of California, Berkeley, September 8, 2009.
- "Large Monitoring Systems: Data Analysis, Design and Deployment"
  Ram Rajagopal, University of California, Berkeley, September 3, 2009.
- "Reasoning about Online Algorithms with Weighted Automata"
  Orna Kupferman, Hebrew University, August 25, 2009.

4.6.2 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.

- C. Tomlin, Short courses on hybrid systems, Royal Institute of Technology, Stockholm, Spring 2010
- S. Sastry, Short course on Generalized Principal Component Analysis, Royal Institute of Technology, Stockholm, Spring 2010.

4.6.3 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.

4.7 Other Specific Products

The following software packages have been made available during this review period on the Chess website, http://chess.eecs.berkeley.edu:

- http://bdtnp.lbl.gov/Fly-Net/bioimaging.jsp?w=node. Implements NODE model for eve mRNA modeling in Drosophila melanogaster. Author: Anil Aswani
- http://hybrid.eecs.berkeley.edu/ NEDE/EDE_Code.zip Implements NEDE estimator for regression on manifolds or with significant data collinearity. Author: Anil Aswani

5 Contributions

5.1 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.

5.2 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.

5.3 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 improv-

ing 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.

References

[1] Anil Aswani, Peter Bickel, and Claire Tomlin. Regression on manifolds: Estimation of the exterior derivative. Annals of Statistics, 2010. To appear.

[2] A. Aswani and C. J. Tomlin. Monotone piecewise affine systems. IEEE Transactions on Automatic Control, 54(8), 2009.

[3] Anil Aswani and Claire Tomlin. Computer-aided drug discovery for pathway and genetic diseases. Unpublished, 2010.

[4] Anil Aswani, Nicholas Boyd, and Claire Tomlin. Graph-theoretic topological control of biological genetic networks. In ACC 2009, page 17001705, June 2009.

[5] Milosh Drezgich and S. Shankar Sastry. Matrix multiplicative weights and non-zero sum games. 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, 2010. Submitted.

[6] A. Aswani, S. Keranan, J. Brown, C. Fowlkes, D. Knowles, M. Biggin, P. Bickel, and C. J. Tomlin. Nonparametric identification of regulatory interactions from spatial and temporal gene expression data. BMC Bioinformatics, 11(413), 2010.

[7] Y. H. chang, J. Gray, and C. J. Tomlin. Optimization-based inference for temporally evolving networks with applications to biology. In Proceedings of the American Control Conference, July 2011. submitted.

[8] S. Itani, J. Gray, and C. J. Tomlin. An ode model for the her2/3-akt signaling pathways in cancers that overexpress her2. In Proceedings of the American Control Conference, July 2010.

[9] Milosh Drezgich and S. Shankar Sastry. On the np in bqp. unpublished, 2010.

[10] J. Biermeyer, T. Templeton, C. Berger, H. Gonzalez, N. Naikal, B. Rumpe, and S. S. Sastry. Rapid integration and calibration of new sensors using the berkeley aachen robotics toolkit (bart). Proceedings des 11. Braunschweiger Symposiums "Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme fur Transportmittel", ITS Niedersachsen, Braunschweig, 2010.

[11] Anil Aswani, Harendra Guturu, and Claire Tomlin. System identication of hunchback protein patterning in early drosophila embryogenesis. In Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, pages 7723–7728, December 2009.

[12] Todd Templeton. Accurate Real-Time Reconstruction of Distant Scenes Using Computer Vision: The Recursive Multi-Frame Planar Parallax Algorithm. PhD thesis, University of California, Berkeley, December 2009.

[13] YuLun Huang, Alvaro Cardenas, Saurabh Amin, Song-Zyun Lin, Hsin-Yi Tsai, and S. Shankar Sastry. Understanding the physical and economic consequences of attacks against control systems. International Journal of Critical Infrastructure Protection, 2(3):72–83, October 2009.

[14] Alvaro Cardenas, Saurabh Amin, Bruno Sinopoli, Annarita Giani, Adrian Perrig, and S. Shankar Sastry. Challenges for securing cyber physical systems. In Workshop on Future Directions in Cyber-physical Systems Security. DHS, July 2009.

[15] Zong-Syun Lin, Alvaro Cardenas, Saurabh Amin, Yu-Lun Huang, Chi-Yen Huang, and S. Shankar Sastry. Model-based detection of attacks for process control systems. In 16th ACM Computer and Communications Security Conference, page submitted. ACM, 2009.

[16] Anil Aswani. Node model for eve mrna modeling in drosophila melanogaster, 2010. Software.

[17] Anil Aswani. Nede estimator, April 2010. Software.