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

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.