ANNUAL REPORT

FOUNDATIONS OF HYBRID AND EMBEDDED SYSTEMS AND SOFTWARE

NSF/ITR PROJECT – AWARD NUMBER: CCR-00225610

UNIVERSITY OF CALIFORNIA AT BERKELEY
VANDERBILT UNIVERSITY
UNIVERSITY OF MEMPHIS

MAY 31, 2004

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

Contents

1. Participants

1.1. People

PRINCIPAL INVESTIGATORS:

THOMAS HENZINGER, (UC BERKELEY, EECS)
EDWARD A. LEE, (UC BERKELEY, EECS)
ALBERTO SANGIOVANNI-VINCENTELLI, (UC BERKELEY, EECS)
SHANKAR SASTRY, (UC BERKELEY, EECS)
JANOS SZTIPANOVITS, (VANDERBILT, ELECTRICAL AND COMPUTER ENGINEERING)

FACULTY INVESTIGATORS:

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

POST DOCTORAL RESEARCHERS:

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

GRADUATE STUDENTS:

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

Andrew D. Dixon (Vanderbilt)
Matthew J. Emerson (Vanderbilt)
Joyti Gandhe (Vanderbilt)
Matthew Harren (UC Berkeley)
Ethan Jackson (Vanderbilt)
Farinaz Koushanfar (UC Berkeley)
Narayanan Krishnan (UC Berkeley)
Alexander Kurzhanskiy(UC Berkeley)
Gabor Madl (Vanderbilt)
Xiaojun Liu (UC Berkeley)
David P. Mandelin (UC Berkeley)
Elefterious Matsikoudis (UC Berkeley)
Stephen Neuendorffer (UC Berkeley)
Songhwai Oh (UC Berkeley)
William Plishker (UC Berkeley)
Kaushik Ravindran (UC Berkeley)
Pannag Sanketi (UC Berkeley)
Peter Schmidt (Vanderbilt)
Tivadar Szemethy (Vanderbilt)
Guang Yang (UC Berkeley)
Yang Zhao (UC Berkeley)
Haiyang Zheng (UC Berkeley)
Ye Zhou (UC Berkeley)

Undergraduate students:
Daniel Balasubramanian
Nickolia Coombs (Vanderbilt)
Rachael Dennison (Vanderbilt)
David Garcia (Vanderbilt)
Shantel Higgins (Vanderbilt)
John Kilby (Vanderbilt)
Efosa Ojomo (Vanderbilt)
Michael Rivera-Jackson (Vanderbilt)
Bina Shah (Vanderbilt)
Edwin Vargas (Vanderbilt)
Trione Vincent (Vanderbilt)

Technical Staff, Programmers:
Christopher Hylands Brooks (UC Berkeley)
Nathan Jew (UC Berkeley)

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

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

1.2. Partner Organizations

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

1.3. Collaborators

- Albert Benveniste (IRISA INRIA, Rennes)
- Hermann Kopetz (Technical University of Vienna, Austria)
- Manfred Morari (ETH, Zurich, Switzerland)
- Gabor Peceli (Technical University of Budapest, Hungary)
- Joseph Sifakis (CNRS VERIMAG, Grenoble, France)
- Kim Larsen (University of Aalborg, Aalborg, Denmark)
- Henrik Christensen (Royal Institute of Technology, Stockholm, Sweden)

2. Activities and Findings

2.1. Project Activities

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

The web address for the overall ITR project is:

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

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

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

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

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

2.1.1. Hybrid Systems Theory

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

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

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

identified discounting as a paradigm for achieving robustness in discrete and hybrid models, and in the past year we developed model checking algorithms for discounted properties. We also studied affine hybrid systems as an approach to robust modeling.

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

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

2.1.1.a. Deep Compositionality

Non-Zero-Sum Games as Compositional System Models

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

Hybrid Systems Modeling Tools: a Survey

In a collaborative project with the European Columbus project, we evaluated a set of tools, languages and formalisms for the simulation, verification and specification of hybrid systems. In this survey we evaluated the following languages and tools: Simulink, Modelica, HyVisual, Scicos, Charon, CheckMate, Masaccio, SHIFT, HSIF, Metropolis and Hysdel. We described their syntax and semantics and we showed their modeling capabilities through simple examples. The goal of this survey is to identify, among all the languages, a potential interchange format for hybrid systems, or to define a new language which has all the necessary semantic and syntactic properties to describe hybrid systems. The survey results are reported in [32].

Composing Quantitative Aspects of Systems

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

2.1.1.b. Robust Hybrid Systems

Design and Verification of Robust System Models

In previous work, we had identified discounting as a mechanism for moving from a discrete, brittle paradigm of boolean-valued property satisfaction to a continuous, robust paradigm of real-valued property estimation. In this last year, we developed algorithms for computing the real value of discounted properties expressed in temporal logic over state transition systems. This work is reported in [38].

Affine Hybrid Systems

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

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

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

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

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

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

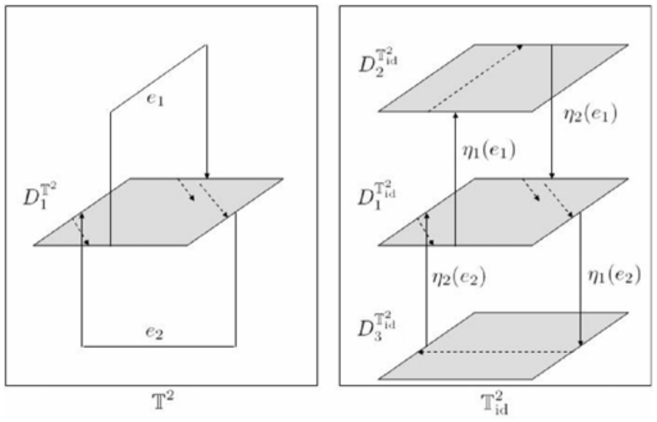

e_1
$D_1^{\mathbb{T}^2}$
e_2
$\mathbb{T}^2$
$D_2^{\mathbb{T}^2_{\mathrm{id}}}$
$\eta_2(e_1)$
$\eta_1(e_1)$
$D_1^{\mathbb{T}^2_{\mathrm{id}}}$
$\eta_2(e_2)$
$\eta_1(e_2)$
$D_3^{\mathbb{T}^2_{\mathrm{id}}}$
$\mathbb{T}^2_{\mathrm{id}}$

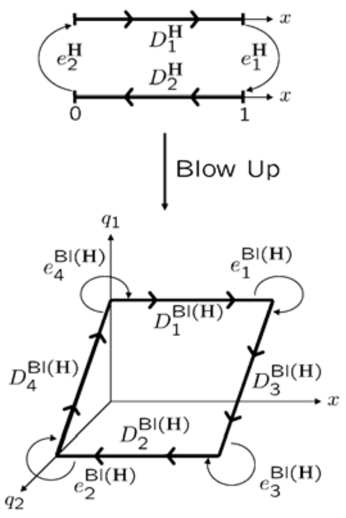
Figure 1. Left: A graphical representation of the hybrid system modeling the two-torus. Right: A graphical representation of the hybrid system that is spatially equivalent to the two-torus but has identity transition maps.

Blowing Up Affine Hybrid Systems

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

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

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



x
$D_1^{\mathbf{H}}$
$e_2^{\mathbf{H}}$
$e_1^{\mathbf{H}}$
$D_2^{\mathbf{H}}$
x
0
1
Blow Up
q_1
$e_4^{\mathsf{Bl}(\mathbf{H})}$
$e_1^{\mathsf{Bl}(\mathbf{H})}$
$D_1^{\mathsf{Bl}(\mathbf{H})}$
$D_4^{\mathsf{Bl}(\mathbf{H})}$
$D_3^{\mathsf{Bl}(\mathbf{H})}$
x
$D_2^{\mathsf{Bl}(\mathbf{H})}$
$e_2^{\mathsf{Bl}(\mathbf{H})}$
$e_3^{\mathsf{Bl}(\mathbf{H})}$
q_2

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

2.1.1.c. Computational Hybrid Systems

Algorithms for the Control of Stochastic Systems

The problem of designing a controller can be viewed as the problem of finding a winning strategy of the control player in a game against the plant player. We improved on the best known algorithms for finding such strategies in the case that there is also a probabilistic source of uncertainty in the game. This work is reported in [36].

Reach Set Calculations using Ellipsoidal Approximations

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

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

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

A Deterministic Operational Semantics for Hybrid System Simulations

We have developed a deterministic operational semantics for hybrid system simulations. We have implemented this semantics in HyVisual, a domain-specific hybrid system modeling framework built on Ptolemy II. HyVisual includes a simulator that gives a well-defined execution by removing unnecessary non-deterministic behaviors when dealing with the discontinuities of piecewise continuous signals and when dealing with simultaneous discrete events.

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

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

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

We are currently developing a formal model of this operational semantics, studying its relationship with those of synchronous languages and discrete-event languages, and unifying these into a general operational semantics for executing heterogeneous models.

2.1.1.d. Stochastic Hybrid Systems

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

Application of Stochastic Hybrid Systems to Biological Systems

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

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

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

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

2.1.2. Model-Based Design

Model-based design focuses on the formal representation, composition, and manipulation of models during the design process. It addresses system specification, model transformation, synthesis of implementations, model analysis and validation, execution, and design evolution. The semantic frameworks in which these models are applied may be domain-specific, offering embedded system designers methods and syntaxes that are closer to their application domain. The project team started off with three different notions for embedded system and software design: platform-based design (developed by Sangiovanni-Vincentelli's group), actor-based design (investigated by Lee's group) and model-based design (advocated by Sztipanovits' group). These approaches emphasize different, complementary aspects of the design process. Platform-based design focuses on the creation of abstraction layers in the design flow and investigates the semantic properties of mapping across these layers. Actor-based design investigates component interaction semantics and theories of composition on different layers of abstractions. Model-based design focuses on the specification and composition of domain-specific modeling languages (DSML-s) and model transformations via metamodeling. As a result of interaction among the research groups a synergistic view is emerging:

- Abstractions and design constraints play central role in the definition of platforms. DSML-s offer a formal way for capturing these abstractions and constraints in metamodels. The required semantic clarity for expressing the mapping across platforms challenges the DSML technology with the need of expanding the abstract syntax oriented metamodeling toward explicit representation of formal semantics.

- A core concept in actor-based design is component interaction semantics defined by models of computation (MoC). New results have been achieved in the semantic foundations for heterogeneous systems, which will be the underpinning for the safe composition of heterogeneous reactive systems.
- Mapping across platforms has fundamental role in platform-based design flow. A new development in the technology of model transformations will contribute to the platform-based design vision.

Our extensive experimental work on networked embedded systems have revealed new challenges in extending model-based design to distributed models of embedded systems. We have reached significant progress in developing a new theory for the design of this system category.

2.1.2.a. Composition of Domain Specific Modeling Languages

Metamodeling

The modeling languages in which models are expressed are domain-specific, offering embedded system designers modeling constructs and syntax that are closer to their application domain. Domain-specific modeling languages (DSMLs) must capture the structural and behavioral aspects of embedded software and systems. Their semantics must emphasize concurrency, communication abstractions, temporal and other physical properties. For example, a DSML framework (i.e. a set of related modeling aspects) for embedded systems might represent physical processes using ordinary differential equations, signal processing using dataflow models, decision logic using finite-state machines, and resource management using synchronous models.

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

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

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

The languages that are used for defining components of DSMLs are called *meta-languages* and the formal specifications of DSMLs are called *metamodels*. The specification of the abstract syntax of DSMLs requires a meta-language that can express concepts, relationships, and integrity constraints. The specification of the semantic domain and semantic mapping is more complicated, because models might have different interesting interpretations; therefore DSMLs might have several semantic domains and semantic mappings associated with them. For example, the *structural semantics* of a modeling language describes the meaning of the models in terms of the structure of model instances: all of the possible sets of components and their relationships, which are consistent with the well-formedness rules in defined by the abstract syntax. Accordingly, the semantic domain for structural semantics is defined by a *set-valued semantics*. The *behavioral semantics* may describe the evolution of the state of the modeled artifact along some time model. Hence, the behavioral semantics is formally captured by a mathematical framework representing the appropriate form of dynamics.

The specification of the abstract syntax of DSMLs requires a meta-language that can express concepts, relationships, and integrity constraints. In our work in Model-Integrated Computing (MIC), we first adopted UML class diagrams and the Object Constraint Language (OCL) as meta-language. This selection was consistent with UML's four layer meta-modeling architecture, which uses UML class diagrams and OCL as meta-language for the abstract syntax specification of UML. Last year, we have developed a MOF-based metamodeling approach and developed a new metamodeling environment using our GME tool suite [44]. Our work on MIC helped to clarify technical details on the Model Driven Architecture (MDA) concept of OMG and we have started up a meaningful interaction with the OMG MDA community [66][7][68].

Compositional Metamodeling

The GME-based metamodeling environment provides support for specifying DSML-s via metamodel composition [67]. There are three characteristics of the GME that make it a valuable tool for the construction of domain-specific modeling environments. First, the GME provides generic modeling primitives that assist an environment designer in the specification of new graphical modeling environments. Second, these generic primitives are specialized to create the domain-specific modeling concepts through meta-modeling. The meta-models explicitly support composition enabling the creation of composite modeling languages supporting multiple paradigms. Third, several ideas from prototype-based programming languages have been integrated with the inherent model containment hierarchy, which gives the domain expert the ability to clone graphical models. Currently, we are exploring characteristics of the new MOF-based metamodeling environment for DSML composition.

Integration of Metamodeling with Hybrid Systems

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

Semantic Foundations for Heterogeneous Systems

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

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

While the framework of Agent Algebra is general enough to encompass a variety of models of computation, the common structure is sufficient to prove interesting results that apply to all models. In particular, we focus on the problem of characterizing the specification of a component of a system given the global specification for the system and the context surrounding the component. This technique, called Local Specification Synthesis, can be applied to solve synthesis and optimization problems in a number of different application areas. The results include sufficient conditions to be met by the definitions of system composition and system refinement for constructing such characterizations. The local specification synthesis technique is also demonstrated through its application to the problem of protocol conversion.

This work is reported in [85].

Generalized causality analysis

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

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

2.1.2.b. Extensions to Distributed Models of Embedded systems

Compositional Theory of Heterogeneous Reactive Systems

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

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

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

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

2.1.2.c. Model Transformation

Meta Generator Technology

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

Pattern-Based Model Synthesis

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

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

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

2.1.2.d. Real-Time Programming Models

Event-Triggered Programming

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

2.1.3. Advanced Tool Architectures

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

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

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

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

Conceptual concurrence between our frameworks is evolving. Whereas previously GME had mastered meta modeling of abstract syntactic properties of actor-oriented models, today it is moving aggressively towards meta modeling of their semantic properties, focusing initially on the the concurrent models of computation that have been implemented in Ptolemy II. Metropolis and Ptolemy II are both seeking to abstract these semantic properties in a somewhat different (and complementary) way than meta modeling. They both define an "abstract semantics" that represents the common features of families of models of computation, and they both realize models of computation through specialization (concretization) of this abstract semantics. They do so, however, in different ways, and by pursuing these different ways, the team is gaining an understanding of the fundamentals behind the use of such abstract semantics in frameworks.

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

2.1.3.a. Syntax and Semantics

Modularity Mechanisms in Actor-Oriented Design

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

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

Code Generation from Actor-Oriented Models

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

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

Metropolis Framework

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

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

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

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

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

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

2.1.3.b. Interface Theories

A Component Model for Heterogeneous Systems

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

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

Interface Modeling and Models of Computation

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

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

2.1.3.c. Virtual Machine Architectures

Types for Real-Time Programs

We developed a type system for Embedded Machine code, which is assembly-like hard real-time code, with the property that well-typed programs are efficiently schedulable. A report has been submitted for publication [57].

Separating Reactivity from Schedulability

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

Implementing Event Scoping

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

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

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

2.1.3.d. Components for Embedded Systems

Mapping Network Applications to Multiprocessor Embedded Platforms

We formulated and solved the task allocation problem for a popular multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor. This method proves to be computationally efficient and produces results that are within 5% of aggregate egress bandwidths achieved by hand-tuned implementations on two representative applications: IPv4 Forwarding and Differentiated Services. The results are reported in the paper "Automated Task Allocation on Single Chip, Hardware Multithreaded, Multiprocessor Systems" at WEPA-1 2004 [1]. We are currently exploring extensions to this work by considering multiple target platforms: a reconfigurable multiprocessor system on the Xilinx Virtex-II Pro and the IXP2400.

Prospector: Code Assistant based on Jungloid Mining

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

2.1.3.e Verification of Embedded Software

Model Checking Quantitative Properties of Systems

We developed model checking algorithms for automata whose states are not labeled with boolean-valued propositions, but with natural-number valued quantities. These numbers might express, for example, power consumption or memory usage. A report on this work has been submitted for publication [34].

Run-Time Error Handling

It is difficult to write programs that behave correctly in the presence of run-time errors. Existing programming language features often provide poor support for executing clean-up ode and for restoring invariants in such exceptional situations. e present a data flow analysis for finding a certain class of error-handling mistakes: those that arise from a failure to release resources or to clean up properly along all paths. Many real-world programs violate such resource safety policies because of incorrect error handling. Our flow-sensitive analysis keeps track of outstanding obligations along program paths and does a precise modeling of control flow in the presence of exceptions. Using it, we have found over 800 error handling mistakes almost 4 million lines of Java code [94]. Among the systems that we have debugged with our tool is the Ptolemy software.

Memory Safety Enforcement:

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

2.1.4. Experimental Research

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

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

2.1.4.a. Embedded Control Systems

Conflict Detection for Aircraft

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

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

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

2.1.4.b. Embedded Software for National and Homeland Security

Soft Walls: Restricting Navigable Airspace

In the last year, we have studied several methods for the Soft Walls controller, looking for a method which will work for a more realistic method of the aircraft. We have looked at a discrete-time formulation of the game theory formulation. Unfortunately, the first theoretical result did not lend itself to a practical, computable algorithm. With George Pappas of the University of Pennsylvania, we have begun to investigate collision avoidance methods based on computational geometry methods. Interim results are reported in [33].

2.1.4.c. Networks of Distributed Sensors

VisualSense: Visual Editor and Simulator for Wireless Sensor Network Systems

Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. VisualSense is a software framework designed to support a component-based construction of such models. It is intended to enable the research community to share models of disjoint aspects of the sensor nets problem and to build models that include sophisticated elements from several aspects. VisualSense can be downloaded from http://ptolemy.eecs.berkeley.edu/visualsense, and is reported in [13].

Large Scale Sensor Networks

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

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

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

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

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

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

Distributed Control in Smart Structures

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

Programming Models for Sensor Networks

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

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

Programming by Sketching for Bitstream Programs

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

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

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

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

Elder Care

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

Networks of Distributed Sensors

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

2.1.4.d. Fault-Driven Applications

Distributed Diagnosis of Complex Physical Systems

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

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

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

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

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

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

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

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

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

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

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

On-line Hierarchical Fault Adaptive Control

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

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

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

where k is the time index, $x(k) \subseteq \Re^n$ is the sampled form of the continuous state vector at time k, $u(k) \subseteq \Re^m$ is the discrete valued input vector at time k, and $q(k) \in Q$ is the mode (discrete state) at time k. Q is a finite set of discrete states that the system can be in. δ is the (partial) transition relation. We use X and U to denote the state space and the finite input set for the system, respectively. For each mode, $q \in Q$ the function Φ is continuous in X and meets the conditions for existence and uniqueness of solutions for a set of initial states $X_o \subseteq X$. The above model is general enough to describe a wide class of hybrid systems, including nonlinear systems and piecewise linear systems. The requirement that the input set is finite is not uncommon in practical computer-controlled systems, where the control inputs are usually discrete and take values from a finite set.

Controller specifications are classified into two categories. The first is set-point specification and the second is performance specifications. The objective of the control structure is to achieve the desired level of the set-point specifications in "reasonable" time, maintain the system stable at the desired value, and optimize the given performance function. Note that, due to the nature of the system environment, it is common that the variables used to optimize the performance functions are evaluated over a quantized finite domain. In certain situations, the optimal operation point can be computed at design time, and used as a set-point objective for the system controller. In this case, the performance function can be translated into a linear or integer programming problem. We assume that optimal points for performance functions can be computed; therefore, the specification is given as one or more set-points, or a state-space region. The specifications may change during operation, and the proposed approach can accommodate the changes.

In the online control approach, the controller explores only a limited forward horizon in the system state space and selects the next event based on the available information. For set-point specification, the selection of the next step is based on a distance map that defines how close the current state is to the desired set point. The distance map can be defined for each state $x \subseteq \Re^n$ as $D(x) = ||x - x_s||$, where $||.||$ is a proper norm for $\Re^n$. In the case of performance specification, the input that minimizes (maximizes) a given multi-attribute utility function, $\sum_i V_i(P_i)$, where each V_i corresponds to a value function associated with performance parameter, P_i. The parameters, p_i, can be continuous or discrete-valued, and they are derived from the system state variables, i.e.,

$P_i(t) = p_i(x(t))$. The value functions employed have been simple weighted functions of the form $V_i(P_i) = w_i P_i$, where the weights take on values in the interval [−1 1], and represent the importance of the parameter in the overall operation of the system. The supervisory controller uses the system model to predict possible behaviors corresponding to different action sequences for a finite forward time horizon, and then selects the action (i.e., control input) that maximizes the utility function. This process is then repeated for the next time step, and so on.

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

Safety-Critical Distributed Applications

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

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

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

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

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

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

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

We have pushed an industrial design example, the Picture-in-Picture (PiP) subsystem for digital television, through the entire Metropolis flow. Functional modeling, architectural modeling, communication and architecture refinement, mapping, and simulation have been carried out for the PiP design within the Metropolis framework. We have also demonstrated that design space exploration is greatly facilitated by clearly separating various aspects of the design process. We are currently involved in further exercising and extending the capabilities of Metropolis by evaluating new applications and architectures, and by adding new verification, refinement and optimization capabilities to Metropolis.

2.2. Project Findings

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

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

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

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

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

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

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

[3] A stability criterion for Stochastic Hybrid Systems

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

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

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

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

Alessandro Abate, Laurent El Ghaoui, submitted to CDC04.

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

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

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

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

[6] Reusable Idioms and Patterns in Graph Transformation Languages

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

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

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

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

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

[8] Affine Hybrid Systems

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

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

[9] Blowing Up Affine Hybrid Systems

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

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

[10] Debugging Temporal Specifications with Concept Analysis

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

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

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

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

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

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

[12] Modeling of Sensor Nets in Ptolemy II

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

Abstract: This paper describes a modeling and simulation framework called VisualSense for wireless sensor networks that builds on and leverages Ptolemy II. This framework supports actor-oriented definition of sensor nodes, wireless communication channels, physical media such as acoustic channels, and wired subsystems. The software architecture consists of a set of base classes for defining channels and sensor nodes, a library of subclasses that provide certain specific channel models and node models, and an extensible visualization framework. Custom nodes can be defined by subclassing the base classes and defining the behavior in Java or by creating composite models using any of several Ptolemy II modeling environments. Custom channels can be defined by subclassing the WirelessChannel base class and by attaching functionality defined in Ptolemy II models.

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

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

Abstract: This paper describes a modeling and simulation framework called VisualSense for wireless sensor networks that builds on and leverages Ptolemy II. This framework supports actor-oriented definition of sensor nodes, wireless communication channels, physical media such as acoustic channels, and wired subsystems. The software architecture consists of a set of base classes for defining channels and sensor nodes, a library of subclasses that provide certain specific channel models and node models, and an extensible visualization framework. Custom nodes can be defined by subclassing the base classes and defining the behavior in Java or by creating composite models using any of several Ptolemy II modeling environments. Custom channels can be defined by subclassing the WirelessChannel base class and by attaching functionality defined in Ptolemy II models.

[14] Continuum Percolation with Steps in an Annulus

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

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

[15] Critical Probabilities in Continuum Percolation

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

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

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

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

[16] Random transceiver networks

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

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

[17] Fast Transmission in Ad Hoc Networks

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

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

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

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

[18] Connectivity of Random Geometric Graphs

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[23] Degree distribution of the FKP network model

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

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

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

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

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

[25] Toward Distributed Diagnosis of Complex Physical Systems

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[29] Ptolemy II Coding Style

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

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

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

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

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

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

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

- Desynchronization motivated by software implementations, where processes that compose the large scale system are locally implemented synchronously while their communication is implemented in an asynchronous style. This approach allows also to run each of the process at its own speed. By using the Lee and Sangiovanni-Vincentelli (LSV) tagged-signal model as a common framework, we offer a comparative exposition of these approaches and we show their precise relationship. In doing so, we also provide some insight on the role of signal absence in synchronous, asynchronous, and globally-asynchronous locally-synchronous (GALS) design styles.

[31] HyVisual: A Hybrid System Visual Modeler

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

Abstract: The Hybrid System Visual Modeler (HyVisual) is a block-diagram editor and simulator for continuous-time dynamical systems and hybrid systems. Hybrid systems mix continuous-time dynamics, discrete events, and discrete mode changes. This visual modeler supports construction of hierarchical hybrid systems. It uses a block-diagram representation of ordinary differential equations (ODEs) to define continuous dynamics, and allows mixing of continuous-time signals with events that are discrete in time. It uses a bubble-and-arc diagram representation of finite state machines to define discrete behavior driven by mode transitions.

In this document, we describe how to graphically construct models and how to interpret the resulting models. HyVisual provides a sophisticated numerical solver that simulates the continuous-time dynamics, and effective use of the system requires at least a rudimentary understanding of the properties of the solver. This document provides a tutorial that will enable the reader to construct elaborate models and to have confidence in the results of a simulation of those models. We begin by explaining how to describe continuous-time models of classical dynamical systems, and then progress to the construction of mixed signal and hybrid systems.

The intended audience for this document is an engineer with at least a rudimentary understanding of the theory of continuous-time dynamical systems (ordinary differential equations and Laplace transform representations), who wishes to build models of such systems, and who wishes to learn about hybrid systems and build models of hybrid systems.

HyVisual is built on top of Ptolemy II, a framework supporting the construction of such domain-specific tools. See Ptolemy II for information.

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

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

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

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

[33] Control Algorithms for Soft Walls

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

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

[34] A Natural Extension of Automata

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

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

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

[35] Games with Secure Equilibria

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

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

[36] Quantitative Stochastic Parity Games

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

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

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

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

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

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

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

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

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

[39] Model Checking Discounted Temporal Properties

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

Abstract: Temporal logic is two-valued: a property is either true or false. When applied to the analysis of stochastic systems, or systems with imprecise formal models, temporal logic is therefore fragile: even small changes in the model can lead to opposite truth values for a specification. We present a generalization of the branching-time logic CTL which achieves robustness with respect to model perturbations by giving a quantitative interpretation to predicates and logical operators, and by discounting the importance of events according to how late they occur. In every state, the value of a formula is a real number in the interval [0,1], where 1 corresponds to truth and 0 to falsehood. The boolean operators and and or are replaced by min and max, the path quantifiers E and A determine sup and inf over all paths from a given state, and the temporal operators F and G specify sup and inf over a given path; a new operator averages all values along a path. Furthermore, all path operators are discounted by a parameter that can be chosen to give more weight to states that are closer to the beginning of the path. We interpret the resulting logic DCTL over transition systems, Markov chains, and Markov decision processes. We present two semantics for DCTL: a path semantics, inspired by the standard interpretation of state and path formulas in CTL, and a fixpoint semantics, inspired by the mu-calculus evaluation of CTL formulas. We show that, while these semantics coincide for CTL, they differ for DCTL, and we provide model-checking algorithms for both semantics.

[40] Approximate Composition

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

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

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

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

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

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

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

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

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

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

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

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

[44] MIC, MDA and MOF

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

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

[45] Using Interaction Costs for Microarchitectural Bottleneck Analysis

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

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

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

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

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

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

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

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

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

[48] Stochastic Rays Pulse Propagation

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

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

[49] A Random Walk Model Of Wave Propagation

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

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

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

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

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

[51] Continuum Percolation With Unreliable And Spread Out Connections

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

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

[52] A Random Walk Model Of Wave Propagation

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

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

[53] Event-Driven Programming With Logical Execution Times

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

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

[54] xGiotto and the Embedded Virtual Machine

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

Abstract: xGiotto is a domain-specific language for the implementation of embedded software applications with hard real-time constraints. The language is an extension of the original Giotto language. In this paper we present the xGiotto tool chain, consisting of the compiler and a specialized virtual machine, the Embedded Virtual Machine (EVM). The compiler checks for determinism (absence of races) and time safety (schedulability within logical execution times) and generates code for the EVM. The EVM integrates an event filter (which handles aperiodic, asynchronous events and event scoping) and a modified Embedded Machine. We also extend the expressiveness of xGiotto by introducing an event calculus. The paper concludes with a case study of implementing an automotive engine controller.

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

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

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

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

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

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

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

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

Abstract: We present a type system for E code, which is an assembly language that manages the release, interaction, and termination of real-time tasks. E code specifies a deadline for each task, and the type system ensures that the deadlines are path-insensitive. We show that typed E programs allow, for given worst-case execution times of tasks, a simple schedulability analysis. Moreover, the real-time programming language Giotto can be compiled into typed E code. This shows that typed E code identifies an easily schedulable yet expressive class of real-time programs. We have extended the Giotto compiler to generate typed E code, and enabled the runtime system for E code to perform a type and schedulability check before executing the code.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

FSM and Giotto, followed by the threaded domains. Within this grouping, the domains are ordered alphabetically (which is an arbitrary choice).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Abstract: The Generic Modeling Environment (GME) is a configurable tool suite that facilitates the rapid creation of domain-specific model-integrated program synthesis environments. There are three characteristics of the GME that make it a valuable tool for the construction of domain-specific modeling environments. First, the GME provides generic modeling primitives that assist an environment designer in the specification of new graphical modeling environments. Second, these generic primitives are specialized to create the domain-specific modeling concepts through meta-modeling. The meta-models explicitly support composition enabling the creation of composite modeling languages supporting multiple paradigms. Third, several ideas from prototype-based programming languages have been integrated with the inherent model containment hierarchy, which gives the domain expert the ability to clone graphical models. This paper explores the details of these three ideas and their implications.

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

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

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

[69] Tool Integration Patterns

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

Abstract: Design tool integration is a highly relevant area of software engineering, which can greatly improve the efficiency of development processes. Design patterns have been widely recognized as important contributors to the success of software systems. This paper describes and compares two large-grain, architectural design patterns that solve specific design tool integration problems. Both patterns have been implemented and used in real-life engineering processes.

[70] Cache Aware Scheduling for Synchronous Dataflow Programs

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

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

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

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

[71] Classes and Subclasses in Actor-Oriented Design

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

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

[72] Actor-oriented Models for Codesign

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

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

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

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

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

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

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

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

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

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

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

Abstract: Most current software engineering is deeply rooted in procedural abstractions. These say little about concurrency, temporal properties, and assumptions and guarantees in the face of dynamic system structure. Actor-oriented design contrasts with (and complements) object-oriented design by emphasizing concurrency and communication between components. Components called *actors* execute and communicate with other actors. While interfaces in object-oriented design (methods, principally) mediate transfer of the locus of control, interfaces in actor-oriented design (which we call ports) mediate communication. But the communication is not assumed to involve a transfer of control. This paper explores the use of behavioral type systems in actor-oriented design.

[76] Soft Walls: Frequently Asked Questions

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

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

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

[77] Overview of the Ptolemy Project

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

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

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

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

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

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

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

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

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

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

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

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

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

[81] Model-Integrated Computing for Heterogeneous Systems

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

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

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

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

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

[83] Hierarchical Reconfiguration of Dataflow Models

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

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

[84] Implementation Issues in Hybrid Embedded Systems

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

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

[85] Semantic Foundations for Heterogeneous Systems

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

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

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

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

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

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

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

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

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

Abstract: The mapping of application functionality onto multiple multithreaded processing elements of a high performance embedded system is currently a slow and arduous task for application developers. Previous attempts at automation have either ignored hardware support for multithreading and focused on scheduling, or have overlooked the architectural peculiarities of these systems. This work attempts to fill the void by formulating and solving the mapping problem for these architectures. In particular, the task allocation problem for a popular multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor, is encoded into a 0-1 Integer Linear Programming problem. This method proves to be computationally efficient and produces results that are within 5% of aggregate egress bandwidths achieved by hand-tuned implementations on two representative applications: IPv4 Forwarding and Differentiated Services.

[88] A Programmable Microkernel for Real-Time Systems

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

Abstract: We present a new software system architecture for the implementation of hard real-time applications. The core of the system is a microkernel whose reactivity (interrupt handling) and proactivity (task scheduling) are fully programmable. The microkernel,

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

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

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

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

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

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

Abstract: Network processors have emerged as prominent examples of multiprocessor application-specific programmable architectures. While there have been significant architectural developments in this field, widespread adoption will be predicated on productively programming high performance applications on these architectures. This paper presents a case study of two programming environments for a common network processor, the Intel IXP1200. We compare the development process, achievable performance, and resource usage of the final implementations using these two programming approaches and draw conclusions regarding the advantages and disadvantages of these approaches.

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

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

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

[92] Kalman Filtering With Intermittent Observations

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

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

[93] Templating Transformations for Bitstream Programs,

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

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

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

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

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

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

[95] Platform modeling and model transformations for analysis

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

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

[96] Model-Integrated Computing Infrastructure for Fault Management

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

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

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

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

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

[98] Model-Integrated Computing for Automotive Applications

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

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

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

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

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

[100] Towards Generation of High-performance Transformations

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

Abstract: In this paper we introduce a graph rewriting language, called Graph Rewriting and Transformation (GReAT), and a code generator tool, which together provide a programming framework for the specification and efficient realization of graph rewriting systems. We argue that the performance problems frequently associated with the implementation of the transformation can be significantly reduced by adopting language and algorithmic optimizations and partial evaluation.

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

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

Abstract: It is difficult to write programs that behave correctly in the presence of run-time errors. Existing programming language features often provide poor support for executing clean-up code and for restoring invariants in such exceptional situations. We present a data flow analysis for finding a certain class of error-handling mistakes: those that arise from a failure to release resources or to clean up properly along all paths. Many real-world programs violate such resource safety policies because of incorrect error handling. Our flow-sensitive analysis keeps track of outstanding obligations along program paths and does a precise modeling of control flow in the presence of exceptions. Using it, we have found over 800 error handling mistakes almost 4 million lines of Java code. The analysis is unsound and produces false positives, but a few simple filtering rules suffice to remove them in practice. The remaining mistakes were manually verified. These mistakes cause sockets, files and database handles to be leaked along some paths. We present a characterization of the most common causes of those errors and discuss the limitations of exception handling, finalizers and destructors in addressing them. Based on those errors, we propose a programming language feature that keeps track of obligations at run time and ensures that they are discharged. Finally, we present case studies to demonstrate that this feature is natural, efficient, and can improve reliability; for example, retrofitting a 34kLOC program with it resulted in a 0.5% code size decrease, a surprising 17% speed increase (from correctly deallocating resources in the presence of exceptions), and more consistent behavior.

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

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

Abstract: Debuggers have been proven indispensable in improving software reliability. Unfortunately, on most real-life software, debuggers fail to deliver their most essential feature - a faithful replay of the execution. The reason is non-determinism caused by multithreading and nonrepeatable inputs. A common solution to faithful replay has been to record the non-deterministic execution. Existing recorders, however, either work only for data-race-free programs or have prohibitive overhead. As a step towards powerful debugging, we develop a practical low-overhead hardware recorder for cache coherent multiprocessors, called Flight Data Recorder (FDR). Like an aircraft flight data recorder, FDR continuously records the execution, even on deployed systems, logging the execution for post-mortem analysis. FDR is practical because it piggybacks on the cache coherence hardware and logs nearly the minimal thread ordering information necessary to faithfully replay the multiprocessor execution. Our studies, based on simulating a four-processor server with commercial workloads, show that when allocated less than 7% of system's physical memory, our FDR design can capture the last one second of the execution at modest (less than 2%) slowdown.

[103] Image and Video Processing Libraries in Ptolemy II

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

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

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

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

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

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

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

[105] Communication Systems Modeling in Ptolemy II

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

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

2.3. Project Training and Development

2.4. Outreach Activities

Our agenda is to build a modern systems science (MSS) with profound implications on the nature and scope of computer science and engineering research, the structure of computer science and electrical engineering curricula, and future industrial practice. This new systems science must pervade engineering education throughout the undergraduate and graduate levels. Embedded software and systems represent a major departure from the current, separated structure of computer science (CS), computer engineering (CE), and electrical engineering (EE). In fact, the new, emerging systems science reintegrates information and physical sciences. The impact of this change on teaching is profound, and cannot be confined to graduate level. Based on the ongoing effort at UCB, we have set out to rearchitect and retool undergraduate teaching at the participating institutions, and to make the results widely available to encourage critical discussion and facilitate adoption. In addition, have recruited new undergraduate students (mostly juniors) from minority institutions through the established REU programs SUPERB-IT at UCB and SURGE at VU to participate in the research of the project.

2.4.1. Curriculum Development for Modern Systems Science (MSS)

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

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

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

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

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

2.4.2. SUPERB-IT Program

The Summer Undergraduate Program in Engineering Research at Berkeley - Information Technology (SUPERB-IT) in the Electrical Engineering and Computer Sciences (EECS) Department offers a group of talented undergraduate engineering students the opportunity to gain research experience. The program's objective is to increase diversity in the graduate school pipeline by affirming students' motivation for graduate study and strengthening their qualifications.

SUPERB-IT participants spent eight weeks at UC Berkeley during the summer of 2003 and have begun the summer of 2004 working on exciting ongoing research projects in information technology with EECS faculty mentors and graduate students. Students who participate in this research apprenticeship explore options for graduate study, gain exposure to a large research-oriented department, and are motivated to pursue graduate study. Additional information about the program can be obtained at:

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

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

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

The students supported at Berkeley in 2003 were Antonio Yordan-Nones, Ismael Sarmiento, Rakesh Reddy, Colin Cochran, Mike Okyere, and Philip Baldwin. The six students built a suite of applications and infrastructure for embedded systems design using the Ptolemy II software infrastructure. Their eight week projects began with an intensive group training in the Chess software lab that familiarized them with the use a CVS code repository, the Eclipse integrated development environment, construction of applications in Ptolemy II, and design and construction of actors for Ptolemy II. They were guided to use an Extreme Programming (XP) software engineering style, which includes pair programming, extensive use of automated test suites, and design and code reviews. All but one of the students had sufficient experience with Java programming that little time was required for familiarization with Java. The one student with little Java experience (Philip Baldwin) focused on building models of distributed, wireless, real-time systems in Ptolemy II, a project that eventually led to VisualSense (see above and [13]). These models used infrastructure built in Java by the other students. In the XP context, he functioned as the 'customer.' Three graduate student mentors facilitated the process, and the operation was coordinated and directed by Professor Edward Lee. Although the students worked together and interacting extensively, each was responsible for an individual project, as described below. Their project posters and reports are available at http://chess.eecs.berkeley.edu/projects/ITR/2003/superb.htm.

Project: Interactive embedded systems showcase

Student: Antonio Yordan-Nones

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

Project: Wireless systems modeling

Student: Philip Baldwin

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

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

Student: Ismael Sarmiento

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

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

Student: Rakesh Reddy

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

Project: Actor-oriented design of smart-home systems

Student: Colin Cochran

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

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

Student: Mike Okyere

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

Project: Platform Based Reconfigurable Hardware Exploration via Boolean Constraints

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

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

Plans for 2004

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

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

Framework: Hyper

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

Project: Ordinary Differential Equation Solver

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

Project: Switched System Simulator

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

Project: Visualization interfaces

Visualization of the simulation, verification, validation, and actual model entry are important for the success of the framework. This student will describe and devise visualization schemes, and how they should be used by machine-produced results (e.g., the execution trace of a switched system may provide multiple visualizations of these switches based on the simulated behavior: a real-time display of the states and transitions, or a time-axis plot of values in the system). The student will likely do prototyping using MATLAB as the visualization tool, and follow up with Python implemented visualizations, to avoid requiring MATLAB as a runtime element for the framework.

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

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

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

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

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

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

Student: Bina P. Shah

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

Student: Edwin Vargas

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

Student: Trione Vincent

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

Project: Visual Tracking

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

Student: Rachael A. Dennison

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

Student: David Garcia

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

Student: Daniel Balasubramanian

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

Project: LEGO Mindstorm Robot Control

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

Student: Michael J. Rivera-Jackson

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

Student: Nickolia S. Coombs

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

Project: TAB Robot Control

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

Student: Shantel Higgins

> Shantel was a rising senior at Vanderbilt University majoring in Electrical Engineering. Shantel is originally from Sugarland, Texas, a small suburb of Houston. She was the treasurer of her sorority, Delta Sigma Theta and was the chair of their First Annual Aids Awareness Walk. She was the community service chair for the Black Student Alliance at Vanderbilt as well as a volunteer for the local YMCA in Nashville. Shantel was an active member of the National Society of Black Engineers and the Society of Women Engineers, as well as a coach for an intramural women's basketball team. Shantel has received honors from the School of Engineering at Vanderbilt and she is the recipient of the Sam McCleskey Engineering Honor Scholarship. Shantel has enjoyed her time at Vanderbilt and she credits her interest and enthusiasm in research and technology from her Vanderbilt curriculum. Shantel's interest is in the field of wireless communications, sparked by her semiconductors class and integrated circuit design and fabrication class. She hopes to pursue a master's and doctoral degrees in this area.

Project: Control of Adaptive Structures

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

3. Publications and Products

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

3.1. Journal Publications

- Bollobás, B. and O. Riordan, “Robustness and vulnerability of scale-free random graphs,” has appeared as the first paper in the first issue of *Internet Mathematics*, 2004, 1—31

- Edwards, Stephen A. and Edward A. Lee, "The Semantics and Execution of a Synchronous Block-Diagram Language," *Science of Computer Programming*, Vol. 48, no. 1, July 2003.

- Franceschetti, M., J. Bruck, and L. Schulman, "A Random Walk Model Of Wave Propagation," *IEEE Trans. on Antennas and Propagation*, 52(5), May 2004.

- Karsai, G. A. Agrawal, F. Shi, J. Sprinkle, "On the Use of Graph Transformation in the Formal Specification of Model Interpreters," *Journal of Universal Computer Science*, Volume 9, Issue 11, 2003

- Karsai, G. Maroti, M., Ledeczi, A. Gray, J., Sztipanovits, J. "Composition and Cloning in Modeling and Meta-Modeling," *IEEE Transactions on Control Systems Technology*, March 2004, pp 263-278, Volume 12, Issue: 2

- Lee, Edward A., Stephen Neuendorffer and Michael J. Wirthlin, "Actor-Oriented Design of Embedded Hardware and Software Systems," Invited paper, *Journal of Circuits, Systems, and Computers*, Vol. 12, No. 3 pp. 231-260, 2003.

- Mosterman, P. J., and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Systems, Man, and Cybernetics – A*, vol. 29, no. 6, pp. 554-565, 1999

- Mosterman, P., Sztipanovits, J., Engell, S., "Computer-Automated Multi-Paradigm Modeling in Control Systems Technology," *IEEE Transactions on Control System Technology* Vol. 12, pp. 223-234, March 2004

- Schmidt, P., I. Amundson and K. D. Frampton, "A Distributed Algorithm for Acoustic Localization Using a Distributed Sensor Network," *Journal of the Acoustical Society of America*, Vol. 115, No. 5, Pt. 2, pp. 2578, 2004. Presented at *147th Meeting of the Acoustical Society of America*, New York, May 24-28, 2004

3.2. Conference Papers

- Abdelwahed, S., G. Karsai and G. Biswas, "Online Safety Control of a Class of Hybrid Systems." *41st IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 1988-1990

- Abdelwahed, S., J. Wu, G. Biswas, J. W. Ramirez, and E. J. Manders, "Online Hierarchical Fault-Adaptive Control for Advanced Life Support Systems," *International Conference On Environmental Systems*, Denver, CO, July 2004

- Agrawal A., Karsai G., Ledeczi A., "An End-to-End Domain-Driven Software Development Framework," *18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, Anaheim, California, October 26, 2003.

- Agrawal, A., Simon, G. Karsai, G., "Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT," *Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT)* 2004. To appear in *Electronic Notes on Theoretical Computer Science*, Elsevier

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

- Ammons, Glenn, David Mandelin, Rastislav Bodik, James Larus, "Debugging Temporal Specifications with Concept Analysis," ACM SIGPLAN Conference on Programming Language Design and Implementation, San Diego, CA, June 2003.

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

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

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

- Biswas, G., G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, G. Karsai, "A robust method for hybrid diagnosis of complex systems," *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS),* Washington, D. C., pp. 1125-1130, June 2003

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

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

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

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

- Carloni, L.P. and A.L. Sangiovanni-Vincentelli, "A Formal Modeling Framework for Deploying Synchronous Designs on Distributed Architectures," *First International Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures* (FMGALS 2003).

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

- Emerson, M. J., Sztipanovits, J. and Bapty, T., "MIC, MDA, and MOF," *IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS* 2004

- Fields, Brian, Rastislav Bodik, Mark D. Hill, Chris J. Newburn, "Using Interaction Costs for Microarchitectural Bottleneck Analysis," *The 36th Annual IEEE/ACM International Symposium on Microarchitecture*, San Diego, CA, December 2003.

- Franceschetti, M. O. Dousse, D. Tse, and P. Thiran, "Closing The Gap In The Capacity Of Random Wireless Networks," *IEEE International Symposium on Information Theory* (ISIT '04), Chicago, Illinois.

- Franceschetti, M., "Power Delay Profile In A Cluttered Environment," *IEEE International Conference on Communications* (ICC '04), Paris, France.

- Franceschetti, M., "Stochastic Rays: The Cluttered Environment," *International Conference on Electromagnetics in Advanced Applications* (ICEAA '03), Turin, Italy.

- Franceschetti, M., "Phase Transitions, An Engineering Perspective," *Allerton Conference on Communication Computing and Control* (Allerton '03), Monticello, Illinois.

- Franceschetti, M., "Stochastic Rays Propagation," *Allerton Conference on Communication Computing and Control* (Allerton '03), Monticello, Illinois.

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

- Henzinger , Thomas A., Christoph M. Kirsch, and Slobodan Matic, "Schedule Carrying Code," *EMSOFT*, Philadelphia, PA October 12-15, 2003.

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

- Karsai, G., "Automotive Software: A Challenge and Opportunity for Model-based Software Development," to appear in *LNCS* volume on *Automotive Software Development Workshop*, San Diego, January, 2004

- Karsai, G., Agrawal A., Ledeczi, A., "A Metamodel-Driven MDA Process and its Tools," *WISME, UML 2003 Conference*, San Francisco, CA, October 2003

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

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

- Lee, Edward A., "Model-Driven Development - From Object-Oriented Design to Actor-Oriented Design," extended abstract of an invited presentation at *Workshop on Software Engineering for Embedded Systems: From Requirements to Implementation* (a.k.a. The Monterey Workshop) Chicago Sept.24, 2003.

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

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

- Pinello, C., L.P. Carloni, and A.L. Sangiovanni-Vincentelli, "Fault-Tolerant Deployment of Embedded Software for Cost-Sensitive Real-Time Feedback-Control Applications," The *Proceedings of the Conference on Design, Automation and Test in Europe* (DATE), 2004.

- Plishker, William, Kaushik Ravindran, Niraj Shah, and Kurt Keutzer, "Automated Task Allocation on Single Chip, Hardware Multithreaded, Multiprocessor Systems," *Workshop on Embedded Parallel Architectures* (WEPA-1), February, 2004.

- Schmidt, P., I. Amundson and K. D. Frampton, "A Distributed Algorithm for Acoustic Localization Using a Distributed Sensor Network," *Journal of the Acoustical Society of America*, Vol. 115, No. 5, Pt. 2, pp. 2578, 2004. To be presented at the *147th Meeting of the Acoustical Society of America*, New York, May 24-28, 2004

- Shah, Niraj, William Plishker, Kurt Keutzer. Comparing Network Processor, "Programming Environments: A Case Study," *2004 Workshop on Productivity and Performance in High-End Computing* (P-PHEC), February, 2004.

- Sinopoli, B., L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE International Conference on Control Decision and Systems* (CDC '03), Maui, Hawaii.

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

- Szemethy, T., G. Karsai, "Platform modeling and model transformations for analysis," *IEEE TC-ECBS and IFIP WG10.1: 4th Joint Workshop on Formal Specifications of Computer-Based Systems, FSCBS* 2004

- Sztipanovits, J., "Design Space Construction and Exploration: A Model Integrated Computing Approach," *MPSOC 2003*, Chamonix, France, June 8, 2003 (presentation in Workshop Proceedings)

- Sztipanovits, J., "Model-Integrated Computing Infrastructure for Fault Management," in *Proc. Of DX-14, 14th International Workshop on Principals of Diagnosis*, Washington DC, June 12, 2003

- Sztipanovits, J., Neema, S., Chen, K., "Model-Integrated Computing for Automotive Applications," *Automotive Software Workshop* San Diego, CA, January 2004, *LNCS*, to appear

- Xu, Min, Rastislav Bodik, Mark Hill, "A "Flight Data Recorder" for Enabling Full-system Multiprocessor Deterministic Replay," The 30th International Symposium on Computer Architecture, San Diego, CA, June 2003.

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

- Agrawal, A., Simon, G. Karsai, G., "Semantic Translation of Simulink/Stateflow models to Hybrid Automata using GReAT," *Proceedings of International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT)* 2004. To appear in *Electronic Notes on Theoretical Computer Science*, Elsevier

- Baldwin, Philip, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, and Yang Zhao, "VisualSense: Visual Modeling for Wireless and Sensor Network Systems," Technical Memorandum UCB/ERL M04/08, University of California, Berkeley, CA 94720, USA, April 23, 2004.

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

- Carloni, Luca, Maria Domenica DiBenedetto, Alessandro Pinto and Alberto Sangiovanni-Vincentelli, “Modeling Techniques, Programming Languages, and Design Toolsets for Hybrid Systems,” Columbus Report IST-2001-38314 WPHS.

- Cataldo, A., C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, H. Zheng "HyVisual: A Hybrid System Visual Modeler," Technical Memorandum UCB/ERL M03/30, University of California, Berkeley, CA 94720, July 17, 2003 (earlier version, January, 2003).

- Cataldo, Adam, "Control Algorithms for Soft Walls," Master's Report, Technical Memorandum UCB/ERL M03/42, University of California, Berkeley, CA 94720, January 21, 2004.

- Cheong, Elaine and Jie Liu, "galsC: A Language for Event-Driven Embedded Systems," Technical Memorandum UCB/ERL M04/7, University of California, Berkeley, CA 94720, USA, 20 April 2004.

- Eker, Johan and Jorn W. Janneck, " CAL Language Report: Specification of the CAL actor language," Technical Memorandum No. UCB/ERL M03/48, University of California, Berkeley, CA, 94720, USA, December 1, 2003.

- Gray, J., J. Sztipanovits, T. Bapty and S. Neema, “Two-level Aspect Weaving to Support Evolution in Model-Driven Software,” in *Aspect-Oriented Programming*

- Harren, Matthew, and George Necula, "Lightweight Wrappers for Interfacing with Binary code in Ccured," In Proceedings of the 3rd International Symposium on Software Security (ISSS03), Tokyo, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II) ," Technical Memorandum UCB/ERL M03/27, University of California, Berkeley, CA USA 94720, July 16, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng, (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture) ," Technical Memorandum UCB/ERL M03/28, University of California, Berkeley, CA USA 94720, July 16, 2003.

- Hylands, C., E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, Y. Xiong, H. Zheng (eds.), "Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains)," TechnicalMemorandum UCB/ERL M03/29, University of California, Berkeley, CA USA 94720, July 16, 2003.

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

- Karsai, G., Agrawal, A. “Graph Transformations in OMG’s Model-Driven Architecture,” to appear in *Lecture Notes in Computer* Science volume on *Applications of Graph Transformation with Industrial Relevance*, 2003.

- Karsai, G., S. Neema, D. Sharp, “Model-Driven Architecture for Embedded Software: A Synopsis and an Example,” To appear in *Science of Computer Programming* (Elsevier) on *Model Driven Architecture: Foundations and Applications Model Driven Architecture*.

- Kohli, Sanjeev, "Cache Aware Scheduling for Synchronous Dataflow Programs," Master's Report, Technical Memorandum UCB/ERL M04/03, University of California, Berkeley, CA 94720, February 23, 2004.

- Lee, Edward A., "Soft Walls: Frequently Asked Questions," Technical Memorandum UCB/ERL M03/31, University of California, Berkeley, CA 94720, July 21, 2003.

- Lee, Edward A., "Overview of the Ptolemy Project," Technical Memorandum No. UCB/ERL M03/25, University of California, Berkeley, CA, 94720, USA, July 2, 2003.

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

- Neuendorffer, Stephen and Edward A. Lee, "Hierarchical Reconfiguration of Dataflow Models," Technical Memorandum UCB/ERL M04/2, University of California, Berkeley, CA 94720, USA, January 2004.

- Neuendorffer, Stephen, "Implementation Issues in Hybrid Embedded Systems," Technical Memorandum No. UCB/ERL M03/22, University of California, Berkeley, CA, 94720, USA, June 24, 2003.

- Roberto Passerone, *Semantic Foundations for Heterogeneous Systems*, PhD dissertation, Department of EECS, University of California at Berkeley, May 2004.

- Weimer, Westley, and George Necula, "Finding and Preventing Run-Time Error Handling Mistakes", In Proceedings of the Object-Oriented Programming Systems, Languages and Applications (OOPSLA04), Vancouver, 2004.

- Yeh, James, " Image and Video Processing Libraries in Ptolemy II," Master's Report, Technical Memorandum No. UCB/ERL M03/52, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

- Zhao, Yang, " A Model of Computation with Push and Pull Processing," Master's Report, Technical Memorandum No. UCB/ERL M03/51, University of California, Berkeley, CA, 94720, USA, December 16, 2003.

- Zhou, Ye, "Communication Systems Modeling in Ptolemy II," Master's Report, Technical Memorandum No. UCB/ERL M03/53, University of California, Berkeley, CA, 94720, USA, December 18, 2003.

3.4. Dissemination

Although this is a long term project focused on foundations, we are actively working to set up effective technology transfer mechanisms for dissemination of the research results. A major part of this is expected to occur through the open dissemination of software tools.

Making these software tools useful and usable outside the research community is a significant issue. Towards this end, we have cooperated with the formation of the Escher consortium, which has begun operating (www.escherinstitute.org) . Escher has negotiated with both Berkeley and Vanderbilt specific priorities for "industrial hardening" of research tools from this project. In particular, at Berkeley, top priority will be placed on Giotto, xGiotto, and Ptolemy II in the near term. At Vanderbilt, top priority will be placed on GME, Desert, and GReAT. General Motors, Raytheon, and Boeing are signed up as charter industrial partners in Escher, and more companies are expected.

An important, emerging forum for dissemination of information and influencing industrial practice is the recently form Model-Integrated Computing Special Interest Group (MIC PSIG) of OMG. (http://mic.omg.org/) This forum is run by industry and its primary goal is the preparation and management of standardization activities related to various aspects model-based design in embedded systems. The ISIS and CHESS teams are very much involved these activities. The White Paper for a standard Open Tool Integration Framework (OTIF) is based on the work of researcher at ISIS [69].

The Chess website, http://chess.eecs.berkeley.edu, includes publications and software distributions. In addition, as part of the outreach effort, the UC Berkeley introductory signals systems course, which introduces hybrid systems, is available at http://ptolemy.eecs.berkeley.edu/eecs20/ and Ptolemy II software is available at http://ptolemy.eecs.berkeley.edu.

The ISIS website, http://www.isis.vanderbilt.edu, makes publications and software available.

3.5. Other Specific Product

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

- HyVisual 3.0 and 4.0-alpha, beta, a block-diagram editor and simulator for continuous-time and hybrid systems. This visual modeler supports construction of hierarchical hybrid systems. It uses a block-diagram representation of ordinary differential equations (ODEs) to define continuous dynamics. It uses a bubble-and-arc diagram representation of finite state machines to define discrete behavior. HyVisual is a packaged subset of Ptolemy II.

- The Giotto and xGiotto systems are a programming methodology for embedded control systems running on possibly distributed platforms. Giotto and xGiotto are programming languages that allow the implementation of deterministic and analyzable real-time systems. While Giotto restricts the programming model to be time-triggered, xGiotto allows it to be event driven. Both the languages use the logical execution time paradigm for tasks to achieve program determinism. This paradigm restricts the execution time of task to be deterministic and platform independent. Giotto software can be accessed at: http://www-cad.eecs.berkeley.edu/~giotto/.

- VisualSense 4.0-alpha, beta: a visual editor and simulator for wireless sensor network system. Modeling of wireless sensor networks requires sophisticated modeling of communication channels, sensor channels, ad-hoc networking protocols, localization strategies, media access control protocols, energy consumption in sensor nodes, etc. This modeling framework is designed to support a component-based construction of such models. It is intended to enable the research community to share models of disjoint aspects of the sensor nets problem and to build models that include sophisticated elements from several aspects. VisualSense is a packaged subset of Ptolemy II.

- Ptolemy II 3.0.2, and 4.0-alpha, beta. Ptolemy II is a set of Java packages supporting heterogeneous, concurrent modeling and design. Its kernel package supports clustered hierarchical graphs, which are collections of entities and relations between those entities. Its actor package extends the kernel so that entities have functionality and can communicate via the relations. Its domains extend the actor package by imposing models of computation on the interaction between entities. Examples of models of computation include discrete-event systems, dataflow, process networks, synchronous/reactive systems, and communicating sequential processes. Ptolemy II includes a number of support packages, such as graph, providing graph-theoretic manipulations, math, providing matrix and vector math and signal processing functions, plot, providing visual display of data, data, providing a type system, data encapsulation and an expression parser. Ptolemy II is available at http://ptolemy.eecs.berkeley.edu/ptolemyII.

- Chic 1.1, an interface compatibility checking framework for hardware and software components interacting with each other and an environment. Chic supports component-based specification of behavioral assumptions and guarantees with respect to, for example, static and/or dynamic constraints on input and output values, method call patterns, resource consumption, etc. The tool then computes the minimal restrictions that must be followed by the environment to allow the known components to functions correctly with respect to some given safety properties. If no such environment exists, the tool correctly concludes that the given set of components is not mutually consistent, or compatible. This framework thus provides a static methodology for early design-time error detection for component-based systems. Chic is available at http://www.eecs.berkeley.edu/~tah/Chic/.

- GalsC, a language and compiler for use with the TinyGALS programming model, which uses TinyOS as the underlying component model. GalsC is compatible with TinyOS 1.x and nesC 1.1.1. You can use the galsC compiler to compile all your existing TinyOS programs. TinyGALS is a globally asynchronous, locally synchronous model for programming event-driven embedded systems, especially sensor networks. At the local level, software components communicate with each other synchronously via method calls. Components are composed to form actors. At the global level, actors communicate with each other asynchronously via message passing, which separates the flow of control between actors. A complementary model called TinyGUYS is a guarded yet synchronous model designed to allow thread-safe sharing of global state between actors without explicitly passing messages. The TinyGALS programming model is structured such that code for all inter-actor communication, actor triggering mechanisms, and access to guarded global variables can be automatically generated from a high level specification. By raising concurrency concerns above the level of TinyOS components, the TinyGALS programming model allows programmers to focus on the main tasks that the application must execute. Programs developed using this task-oriented model are thread safe and easy to debug.

- Nc2momllib: This tool is an extension of the nesC compiler. nesC is "an extension to the C programming language designed to embody the structuring concepts and execution model of TinyOS. TinyOS is an event-driven operating system designed for sensor network nodes that have very limited resources (e.g., 8K bytes of program memory, 512 bytes of RAM)." TinyOS, described at http://webs.cs.berkeley.edu/tos/, is used, for example, on the Berkeley MICA "motes," which are small wireless sensor nodes. Nc2momllib is used to convert nesC files (.nc) into MoML files (.xml). This will create the Ptolemy II libraries of components that are used to assemble models. TinyOS provides a rich library of nesC components.

- GME is a metaprogrammable visual modeling environment that allows the metamodeling of new domain-specific modeling languages (DSML-s). The metamodeling is done using UML/OCL. Once the metamodels are compiled, they can be used to instantiate the GME tool to support the abstract syntax and static semantics of the specified DSML. GME is available via http://www.isis.vanderbilt.edu/Projects/gme/.

- GREAT is a toolsuite for building model transformation tools using graph transformation techniques. The transformations are expressed in the GREAT language, which supports the visual specification in terms of the metamodel of the input and the output of the transformations. The tool suite includes a (GME-based) modeling language, a virtual machine that directly executes the transformations, a code generator that compiles transformations into C++ code, and a debugger that allows debugging of transformation programs. GREAT is available from http://www.isis.vanderbilt.edu/Projects/mobies/downloads.asp#GREAT

- SMOLES: As reported in a paper for the FSCBS 2004 workshop, we have developed a technique for modeling platforms of embedded systems and using these models to perform analysis on the models. We have designed a simple modeling language for embedded systems (SMOLES) that supports a version of the dataflow paradigm. The language has been used in the 2003 SIPHER program by undergraduates to develop small-scale embedded applications. Next we have developed a technique for transforming SMOLES models into timed automata (T/A) models that are equivalent representations for the applications. These T/A models can then be analyzed using a standard analysis tool (Uppaal) to determine the timing properties of the application. The main contribution of this work was the generative approach to create the model of the run-time (dataflow) kernel using graph transformation techniques

4. Contributions

This section summarizes the major contributions during this reporting period.

4.1. Within Discipline

4.1.1. Hybrid Systems Theory

- We introduced a game-theoretic view of system models that is compositional, and interestingly, not a zero-sum game. We have identified and studied a special kind of Nash equilibrium for such games.
- We have collaborated with our European colleagues to conduct a systematic and detailed survey of hybrid systems modeling tools.
- We have developed a theory for composing robust models of systems where instead of traces having a hard distinction between “possible” and “impossible,” the distinction is graduated, indicating a “distance” to a possible trace.
- We have developed algorithms for computing the real value of discounted properties, which are continuous values that replace discrete, brittle, Boolean-valued property satisfaction, expressed in temporal logic over state transition systems.
- We have developed a theory of affine hybrid systems.
- We have improved on the best known algorithms for finding strategies for the control of stochastic hybrid systems.
- We have constructed a toolbox using ellipsoidal methods to calculate reach sets for linear dynamic systems.
- We have developed a deterministic operational semantics for hybrid systems simulations that deals systematically with zero-time events, simultaneous events, and discontinuities in piecewise continuous signals.
- We have obtained a stability result for stochastic hybrid systems and have applied them to studying biological systems.

4.1.2. Model-Based Design

- Applying our ongoing work on metamodeling, we have developed a metamodel for the abstract syntax of the Hybrid System Interchange Format (HSIF) and have used it in developing a translator between HSIF and Simulink/Stateflow.

- We have developed agent algebras as a formal framework for uniformly representing and reasoning about models of computation used in the design of hybrid and embedded software systems.
- We have developed a theory and compositional framework for reasoning about causality in components that composed under concurrent models of computation.
- We have extended our previously developed tagged-signal model for concurrent models of computation to represent the semantics of globally asynchronous, locally synchronous systems built upon loosely time-triggered architectures.
- We have developed a language and a suite of supporting tools for the specification of model transformations based on graph rewriting.
- We have developed an approach to model synthesis based on patterns specified formally as meta models.
- We have extended the principles of the Giotto language beyond periodic time-driven systems to aperiodic event-driven systems, and we have developed a language called xGiotto supporting this approach.

4.1.3. Advanced Tool Architectures

- We have defined and prototyped modularity mechanisms (classes, subclasses, inheritance, interfaces) for actor-oriented design, complementing the prevailing object-oriented methods.
- We have further developed the code generation approach based on component specialization by developing a formal framework for reasoning about reconfiguration in embedded software.
- We have improved the performance and feature set of the Metropolis framework.
- We have further developed our notion of interface theories to support reasoning about heterogeneous component composition and about the dynamics of models of computation.
- We have introduced a strong type system into our embedded virtual machine, and have separated scheduling functionality into a separate virtual machine architecture called the S machine.
- We have introduced the idea of event scoping in embedded software, and have defined an extended embedded virtual machine that supports it.
- We have implemented our embedded virtual machine architectures on KURT Linux and have developed a concept demonstration prototype of code generation from Ptolemy II for this target.

- We have formulated and solved the task allocation problem for a popular, multithreaded, multiprocessor embedded system, the Intel IXP1200 network processor.
- We have developed an interactive tool called Prospector that is an extension of Eclipse supporting navigating through complex framework documentation.
- We have developed model checking algorithms for automata where states are labeled with natural-number valued quantities rather than Booleans, expressing for example power consumption and memory usage.
- We have developed a flow-sensitive static analysis tool that identifies failures of software to deal with outstanding obligations, for example closing open files when exception conditions occur.
- We have improved the Ccured static analysis tool for C programs to deal with library functions where source code is not available.

4.1.4. Experimental Research

- We have made progress on models for conflict detection for aircraft.
- We have improved our means for modifying the control software in fly-by-wire aircraft to restrict the airspace that an aircraft will fly into. The scheme is called Soft Walls.
- We have developed a modeling environment for wireless sensor networks.
- We have fundamental results on the connectivity of large-scale sensor networks.
- We have developed methods for dealing with safety critical distributed applications that include novel models of computation to capture safety specifications and synthesis algorithms that map optimally the requirements on a redundant architecture.
- We have deployed the Metropolis platform-based design methodology and the Metropolis environment for the solution of a Picture-in-Picture subsystem for HDTV as specified by one of our industrial partners.
- We have developed new programming models for sensor networks that build on the popular TinyOS models.
- We have developed programming models for bit streaming applications that use sketching of strategies with code generation.
- We have been adapting sensor networks technology to address elder care problems.

4.2. Other Disciplines

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

4.3. Human Resource Development

Several panels in important conferences and workshops pertinent to embedded systems (e.g., DAC, ICCAD, HSCC, EMSOFT, CASES, and RTSS) have pointed out the necessity of upgrading the talents of the engineering community to cope with the challenges posed by the next generation embedded system technology. Our research program has touched many graduate students in our institutions and several visiting researchers from industry and other Universities so that they now have a deep understanding of embedded system software issues and techniques to address them. The industrial affiliates to our research program are increasing and we hope to be able to export in their environments a modern view of system design. Preliminary feedback from our partners has underlined the importance of this process to develop the professional talent pool.

4.4. Research and Education

In this report, we have touched multiple times on research and education especially in the outreach section. In addition, there has been a strong activity in the continued update of the undergraduate course taught at Berkeley on the foundations of embedded system design. The graduate program at Berkeley and at Vanderbilt has greatly benefited from the research work in the ITR. EE249 at Berkeley has incorporated the most important results thus far obtained in the research program. EE 290 A and C, advanced courses for PhD students, have featured hybrid system and the interface theories developed under this project. EE219C, a course on formal verification, has used results from the hybrid theory verification work in the program. Finally, many final projects in these graduate courses have resulted in papers and reports listed in this document.

4.5. Beyond Science and Engineering

Embedded systems are part of our everyday life and will be much more so in the future. In particular, wireless sensor networks will provide a framework for much better environmental monitoring, energy conservation programs, defense and health care. Already in the application chapter, we can see the impact of our work on these themes. Elder care, soft walls and distributed diagnosis are a few examples of this impact. In the domain of transportation systems, our research is improving safety in cars. Future applications of hybrid system technology will involve biological systems to a much larger extent showing that our approach can be exported to other field of knowledge ranging from economics to biology and medicine. At Berkeley, the Center for Information Technology Research in the Interest of Society is demonstrating the potential of our research in fields that touch all aspects of our life.