YEAR 5
and
FINAL REPORT
Berkeley Portion

High-Confidence Design of Adaptive, Distributed Embedded Control Systems (HCDDES)

PERIOD OF PERFORMANCE COVERED:
June 1, 2010 – August 31, 2011

Contents

1 Berkeley Participants

1.1 People

PRINCIPAL INVESTIGATOR:

Claire Tomlin

CO-PRINCIPAL INVESTIGATORS:

Edward A. Lee (Faculty, partially funded)

S. Shankar Sastry (Faculty, funded elsewhere)

GRADUATE STUDENTS:

Maximilian Balandat (Graduate Student, partially funded)

Jerry Ding (Graduate Student, partially funded)

Humberto Gonzalez (Graduate Student, partially funded)

Maryam Kamgarpour (Graduate Student, partially funded)

Michael Peter Vitus (Graduate Student, partially funded)

ACADEMIC RESEARCHERS:

Jan Reineke (Post-doc, partially funded)

STAFF:

Christopher Brooks (Software Engineer, funded 5%)

Jessica Gamble, Research Support Staff, partially funded)

Dusan Stipanovic, visiting professor, partially funded)

Andrew. M. Sy (Undergraduate, partially funded)

Sridatta Thatipamala, (Undergraduate, partially funded (summers only))

2 High-Confidence Design for Distributed Embedded Systems (HCDDES) MURI

The Multidisciplinary University Research Initiative (MURI) project on High-Confidence Design for Distributed Embedded Systems integrate verification, validation, and test procedures throughout the complete design, development and maintenance cycle, from requirements capture to deployment and life cycle updates.

This project was funded by the Air Force Office of Scientific Research. The project started with a kickoff meeting on August 7, 2006 and on August 31, 2011.

The design of embedded systems is challenging for high-confidence aerospace systems, where technology leadership is the key to superiority. Next generation unoccupied air vehicles (UAVs) and space vehicles (SVs) are complex systems of systems involving multiple synchronous and asynchronous processes in an architecture distributed across several processors both within a single UAV (SV) and across groups of UAVs (SVs). Additionally, autonomy, fault tolerance, and resource optimization require that mixed-criticality functions are resident on the same processors in this architecture. Guaranteeing the provably correct behavior of the overall embedded system is extremely difficult, especially with respect to timing constraints and their relationship to safety of the physical systems, and performance specifications associated with mission-level objectives. Traditional software engineering methods cannot solve these problems because they do not address properties key to the interaction of physical processes with software, such as timing, mixed criticality functions, and concurrent processes.

This project developed a comprehensive approach to the design of high-confidence complex embedded systems, that is, systems that are correct-by-construction, fault tolerant, secure, and degrade gracefully under fault conditions or information attack. Our approach integrates verification, validation, and test procedures throughout the complete design, development and maintenance cycle, from requirements capture to deployment and life cycle updates. This MURI project had the following research areas:

1. Hybrid and embedded systems theory.
2. Model-based software design/verification.
3. Composable Tool Architectures.

4. Testing and Experimental Validation.

At Berkeley, the participating faculty were Claire Tomlin, S. Shankar Sastry and Edward A. Lee. In addition, during the 2009-2010 academic year, Dusan Stipanovic (Associate Prof. UIUC) spend his sabbatical year at Berkeley and was partially funded on this project.

This report contains Berkeley's year 5 input (key achievements, publications, management related data) in an overview style that lists the key results of the last year and during the project.

3 Status of Effort

4 Accomplishments and New Findings

4.1 Hybrid and Embedded Systems Theory

4.1.1 Hierarchies of Robust Hybrid and Embedded Systems (Tomlin, Krogh, Sastry)

Reachability Analysis. In safety-critical applications, the task of controller design and synthesis is often subject to hard constraints such as safe operation envelopes, target attainability requirements, and limits on the admissible control inputs. Furthermore, due to various uncertainties in the modeling phase and operation phase, the controller design must be robust to factors such as modeling error, environment disturbances, and adversarial actions.

Over the course of this MURI, we have developed several methods for addressing these problems in the context of a hybrid system abstraction, which is a powerful modeling framework encapsulating both continuous and discrete dynamics. Our approach is based upon a dynamic game formulation of reachability analysis in which the safety and target attainability objectives are encoded as reachability objectives, and the disturbance is modelled as an adversarial agent. In the following, we will briefly outline some of these methods, along with relevant application scenarios.

a) Provably Safe Maneuver Sequence Design

In the case where the sequence of transitions in a hybrid system is known ahead of time, we have developed a systematic method for designing the discrete transition conditions, as well as the continuous control laws so as to satisfy a given safety specification [1, 2]. This method is applied to the

problem of flight maneuver design for automated aerial refueling (AAR) . This scenario arises when unmanned aerial vehicles (UAVs) undertake long range missions need to be refueled mid-flight by a human-operated tanker aircraft. The refueling procedure is illustrated in Figure 1.

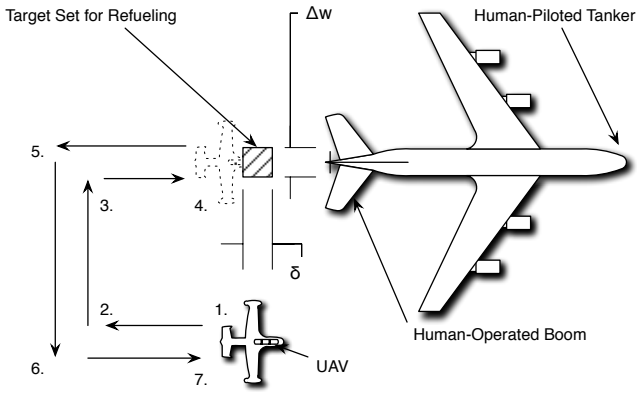


Figure 1: Automted aerial refueling sequence.

Under a hybrid system abstraction, the discrete modes are the flight maneuvers performed in transitioning between the waypoints labeled in Figure 1, while the continuous states are the relative coordinates of the tanker aircraft in the UAV reference frame, represented by a vector $x = (x_1, x_2, x_3)$ (the longitudinal, lateral, and heading coordinates, respectively). The relative motion between the tanker aircraft and UAV is then described by a kinematics model of the form $\dot{x} = f(x, u, d)$, where u is the control input (in this case the translational and angular velocities of the UAV), and d is the disturbance entering into the system (in this case the fluctuation in tanker velocity due to wind effects).

The maneuver sequence design problem involves choosing the switching conditions between the flight maneuvers, as well as the continuous control laws for the input u, so as to ensure that a collision between UAV and tanker is prevented regardless of possible environment disturbances d and command latencies. Under our proposed approach, a reachability calculation is carried out for each flight maneuver in the refueling sequence to determine 1) the capture set, which is the set of aircraft states from which a target waypoint can be attained within a finite time horizon, and 2) the collision set, which is the set of aircraft states from which the trajectory of a flight maneuver passes

through a collision zone centered on the tanker aircraft. An example of these sets computed for the Precontact maneuver (transition between waypoints 2 and 3) is shown in Figure 2.

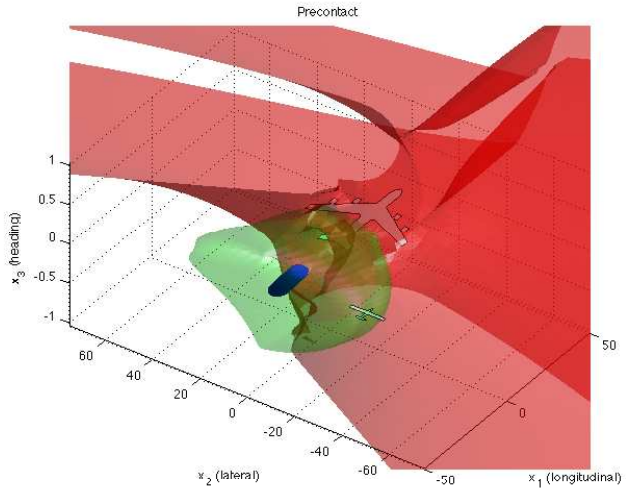


Figure 2: Capture set (light, green) and collision set (dark, red) computed for Precontact maneuver in aerial refueling sequence.

At design time, the capture sets and collision sets computed for the various maneuvers in the AAR sequence can be used to guide the choice of maneuver control laws and switching conditions to ensure that each maneuver terminates in an aircraft state that satisfies the target-attainability and safety objectives of the next maneuver (thus allowing the next maneuver to be feasibly initiated). Furthermore, through appropriate modifications of the reachability analysis, the effects of bounded disturbances and communication latency can also be taken into account. However, in such cases, the resulting design of maneuver control laws and switching conditions is in general more conservative than the case in which the robustness factors are not considered. The performance of the control laws and switching conditions for the full AAR sequence has been validated in simulation with realistic model parameters, as detailed in [2].

b) Automatic Controller Synthesis for Switched Nonlinear Systems

While in certain applications a mode sequence is specified a priori according to designer insights, there are many cases in which one is simply

given a set of controlled modes of operation and the task is to construct a mode selection policy, possibly as function of the state measurements, so as to ensure that desired control objectives are satisfied. In particular, consider a switched nonlinear system of the following form:

$$\dot{x}(t) = f_{q(t)}(x(t), u(t), d(t)) \tag{1}$$

where $\{f_q, q \in Q\}$ is a family of vector fields parametrized by a finite index set Q (for example a finite set of flight maneuvers); u is a continuous control input; and d is a disturbance input.

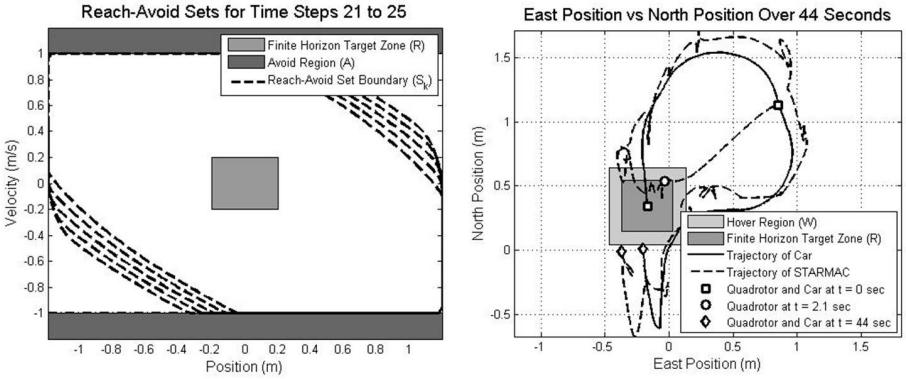
We are interested in synthesizing controllers, which involves both a choice of the discrete mode q and the continuous input u, so as to drive the continuous state x into a set of goal configurations R, while avoiding an unsafe set A, subject to system dynamics (1) and unknown but bounded, time-varying disturbances. We call this a *reach-avoid* problem. For practical considerations, the controller is alllowed to modify input selections only at regularly spaced sampling instants, $t = 0, T, ..., NT$, at which measurements of the system state are received.

In [3] and [4], a solution to this controller synthesis problem is proposed, based upon iterative reachability calculations over sampling intervals. In particular, we compute over successive iterations $k = 0, 1, ..., N$, the set of initial conditions S_k for which there exists an admissible feedback policy satisfying the reach-avoid objectives over $[0, kT]$, subject to the worst-case disturbance, using a differential game formulation of reachability analysis. The resulting set S_N then provides us with the N-step feasible set for the reach-avoid problem. Furthermore, as discussed in [3], we can derive from the representation of the sets S_k a set-valued control law which satisfies the desired control objectives over $[0, kT]$. By storing the reachable sets as lookup tables, one can then compute the appropriate control inputs in an online setting by checking set inclusions. It has also been shown that a variant of this reachability calculation can be used to address the invariance problem, in which the objective is to remain within a desired set indefinitely [4].

We have applied this controller synthesis approach in an experimental setting to a target tracking problem, in which the objective is to control a quadrotor helicopter to first hover over a stationary ground vehicle and then track the vehicle as it starts moving, while satisfying certain velocity constraints. In this case, the modes of the system are used to represent discrete choices of roll and pitch angles, which affects the quadrotor acceleration in

the x and y directions, while disturbances appear in the form of model uncertainties, as well as the movement of the ground vehicle, which is not planned ahead of time.

Using the procedure described previously, we computed the set of initial conditions in the relative position-velocity space reachable to the hover region over 2.5 seconds, with the results shown in Figure 3(a). The control policy derived from this reachability computation was then implemented onboard the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). The trajectories from an experimental run are shown in Figure 3(b). It can be seen that the quadrotor indeed enters the hover region within the time horizon of interest, and then remains within this region for the rest of the experiment, despite movements of the ground vehicle.



(*a*) (*b*)

Figure 3: Results for target tracking application: (a) reach-avoid sets in position-velocity space over 2.5 seconds ($T = 0.1$ s); (b) experimental trajectories of quadrotor and ground vehicle.

4.1.2 Optimal Control of Switched Hybrid Dynamical Systems (Sastry, Tomlin)

A natural extension of classical dynamical systems, where the state of the system is governed by a single differential equation, are switched dynamical systems, where the state of a system is governed by a finite number of differential equations, each of which can be arbitrarily chosen at an instant of time. The control parameter for such systems has a discrete component, the

sequence of modes, and two continuous components, the duration of each mode and the continuous input. Switched systems arise in numerous modeling applications [**?**, **?**].

Stemming from Branicky et al.'s seminal work that established a necessary condition for the optimal trajectory of switched systems in terms of quasi-variational inequalities [**?**], there has been growing interest in the optimal control of such systems. However, Branicky provided only limited means for the computation of the required control. Others have employed variants of dynamic programming to develop algorithms to address the special case of piecewise-linear or affine systems [**?**, **?**, **?**]. Since after each iteration of these algorithms the number of possible switches grows exponentially, the representation of the optimal value function becomes increasingly complex. These approaches focus on addressing this particular shortcoming by considering a variety of possible relaxations of the optimal value function.

Several address just the continuous component of the optimal control of an unconstrained nonlinear switched system while keeping the sequence of modes fixed. Given a fixed mode schedule, Xu et al. develop a bi-level hierarchical optimization algorithm: at the higher level, a conventional optimal control algorithm finds the optimal continuous input assuming fixed mode duration and at the lower level, a conventional optimal control algorithm finds the optimal mode duration while keeping the continuous input fixed [**?**]. Axelsson et al. consider the special case of unconstrained nonlinear autonomous switched systems (i.e. systems wherein the control input is absent) and develop a similar bi-level hierarchical algorithm: at the higher level, the algorithm updates the mode sequence by employing a single mode insertion technique, and at the lower level, the algorithm assumes a fixed mode sequence and minimizes the cost functional over the switching times [**?**, **?**].

We generalized Axelsson's approach by constructing an optimal control algorithm for constrained nonlinear switched dynamical systems [**?**, **?**]. In our first paper we developed a bi-level hierarchical algorithm that divided the problem into two nonlinear constrained optimization problems. At the lower level, our algorithm assumed a fixed modal sequence and determined the optimal mode duration and optimal continuous input. At the higher level, our algorithm employed a single mode insertion technique to construct a new lower cost sequence. The result of our approach was an algorithm that provided a sufficient condition to guarantee the local optimality of the mode duration and continuous input while decreasing the overall cost via mode

insertion. Though this was a powerful outcome given the generality of the problem under consideration, it suffered from three shortcomings which made its immediate application difficult. First, if our algorithm was initialized at an infeasible point it was unable to find a feasible lower cost trajectory. Unfortunately, initializing an optimization algorithm with a feasible point is nontrivial. Second, our algorithm did not incorporate multiple objectives into its cost function, which is useful for path planning type tasks. Finally, our algorithm did not penalize the number of hybrid jumps. In our second paper we design a new algorithm to address these three deficiencies and detail the utility of this modified approach on two examples.

We are interested in the optimal control of dynamical systems whose trajectory is governed by a differential equation of the form:

$$\dot{x}(t) = f\big(x(t), u(t), d(t)\big), \quad \forall t \geq 0, \quad x(0) = x_0, \tag{2}$$

where $u : [0, \infty) \to \mathbb{R}^m$ is continuous input and $d : [0, \infty) \to \{1, 2, \ldots, Q\}$ is the discrete input. Instead of directly optimizing over the discrete input d, we optimize over the sequence $\sigma : \mathbb{N} \to \{1, 2, \ldots, Q\}$ and $s : \mathbb{N} \to [0, \infty)$, where it must be noted that given a pair (σ, s) one can construct a discrete input recursively by, given $k \in \mathbb{N}$, applying the input $\sigma(k)$ for $s(k)$, and then repeat replacing k with $k + 1$. Also, since we want to account for several objectives as mentioned above, we introduce a extra variable $w : \{1, 2, \ldots, W\} \to \mathbb{N}$ which indexes the objectives.

Our algorithm finds a numerical solution of the following problem:

$$(OCP) \qquad \min_{\sigma,s,u,w} \int_0^T L\big(x(t), u(t), t\big)dt + \sum_{k=1}^{W} \phi_k\big(x(\tau_{w(k)})\big) + C\#\sigma, \tag{3}$$

subject to:

$$\tau_k = \sum_{i=1}^{k} s(i), \tag{4}$$

$$\dot{x}(t) = f\big(x(t), u(t), d(t)\big), \quad t \in [0, T], \quad x(0) = x_0, \tag{5}$$

$$d \text{ is constructed from the pair } (\sigma, s), \tag{6}$$

$$h_j\big(x(t)\big) \leq 0, \quad \forall t \in [0, T], \quad j \in \{1, 2, \ldots, J\}. \tag{7}$$

The problem (OCP) has both continuous as well as discrete variables, thus it is usually solved numerically using Integer Programming algorithms, which

suffer the curse of dimensionality, i.e. as the size of the discrete variable increases linearly the number of function evaluations increases exponentially. Our algorithm offers numerical solutions which are not strict minimizers (i.e. our algorithm might stop at points that are not necessarily global minimizers) but it does not suffer the curse of dimensionality.

The algorithm solves a two-level optimization problem. In the first level it fixes σ and w and it minimizes s and u using known optimal control techniques as the ones in [**?**]. The second level performs a first-order variation of the cost function after the instantaneous insertion of an extra mode at a particular time: if the variation results in a decrease of the cost, then create a new sequence σ including that insertion and repeat the process. This procedure converges to a class of points where the first-order variation produces no change in the cost, and hence no single mode insertion can further decrease the cost [**?**, **?**].

As an example, we consider the optimal control of a quadrotor helicopter in 2D using a model described in [**?**]. The evolution of the quadrotor can be defined with respect to a fixed 2D reference frame using six dimensions where the first three dimensions represent the position along a horizontal axis, the position along the vertical axis and the roll angle of the helicopter, respectively, and the last three dimensions represent the time derivative of the first three dimensions. We model the dynamics as a three mode switched system (the first mode describes the dynamics of going up, the second mode describes the dynamics of moving to the right, and the third mode describes the dynamics of moving to the left) with a single input as described in Table 1 where $L = 0.305$ meters, $M = 1.3$ kilograms, $I = 0.0605$ kilogram meters squared and $g = 9.8$ meters per second squared. The cost and input constraints are as described in Table 2 where the waypoints, $\hat{w}(i)$, are:

$$\hat{w}(1) = \begin{bmatrix} 2 \\ 5 \\ \pi \end{bmatrix} \text{ and } \hat{w}(2) = \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}, \tag{8}$$

and the time-varying desired trajectory is defined as:

$$r(t) = \begin{cases} (t, 1), & \text{if } t < 2 \\ \left(2 + \cos\left(t - 2 - \frac{\pi}{2}\right), 3 + 2\sin\left(t - 2 - \frac{\pi}{2}\right)\right), & \text{if } t < 2 + 2\pi \\ (t - 2 + 2\pi, 1), & \text{if } t < 4 + 2\pi \\ (4, 1), & \text{else} \end{cases} \tag{9}$$

$$\text{Mode 1:} \quad \ddot{x}(t) = \begin{bmatrix} \frac{\sin x_3(t)}{M}\left(u(t) + Mg\right) \\ \frac{\cos x_3(t)}{M}\left(u(t) + Mg\right) - g \\ 0 \end{bmatrix}$$

$$\text{Mode 2:} \quad \ddot{x}(t) = \begin{bmatrix} g \sin x_3(t) \\ g \cos x_3(t) - g \\ \frac{-Lu(t)}{I} \end{bmatrix}$$

$$\text{Mode 3:} \quad \ddot{x}(t) = \begin{bmatrix} g \sin x_3(t) \\ g \cos x_3(t) - g \\ \frac{Lu(t)}{I} \end{bmatrix}$$

Table 1: The dynamics of each of the modes of the quadrotor for our example.

$$L(x(t), u(t), t) = 10 \sum_{j=1}^{2} \big(x_j(t) - r_j(t)\big)^2$$

$$\phi_k\big(x(\tau)\big) = 100 \sum_{j=i}^{2} \big(x_j(\tau) - \hat{w}_j(i)\big)^2 + 1000 \left(\sin^2\left(\frac{x_3(\tau) - \hat{w}_3(i)}{2}\right)\right)$$

$$C = 1$$

Table 2: The components of the cost function, the input constraints, and the parameters of the optimality function for our example.

Figure 4: Optimal trajectories (the quadrotor is drawn in black and the normal direction to the frame is drawn in gray) on the top row and bottom left in an environment with constraints (drawn in brown) and objectives (drawn in green). The first image shows iteration 1, the second iteration 7, and the third is iteration 10.

The state is constrained to remain above the ground and outside of a box of height and width both equal to 1 centered at $(1, 1)$. The result of iterations 1, 7, and 10 of the algorithm are shown in Figure 4.

4.1.3 Hybrid and Embedded Systems Theory (Lee)

FSMs are actors whose behavior is described using a finite set of states and transitions between the states. The transitions between the states are enabled by guards, which are boolean-valued expressions that can reference inputs to the actor and parameters in scope. The transitions can produce outputs and can update the value of parameters in scope. Modal models extend FSMs by allowing states to have refinements, which are hierarchical Ptolemy II models. The refinements may themselves be FSMs, modal models, or any composite actor containing a director compatible with the domain in which the modal model is being used. On the basis of several examples, the memorandum describes the operational semantics, the practical usage, and the semantics of time in modal models.

During this period, we released our update to the modal model system as part of Ptolemy II 8.0.1. In addition we wrote a conference paper "Modal Models in Ptolemy," for the Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, held in conjunction with MODELS [5] [6], the abstract of which is reproduced below:

> "Ptolemy is an open-source and extensible modeling and simulation framework. It offers heterogeneous modeling capabilities by allowing different models of computation to be composed hierarchically in an arbitrary fashion. This paper describes modal models, which allow to hierarchically compose finite-state machines with other models of computation, both untimed and timed. The semantics of modal models in Ptolemy are defined in a modular manner."

Model models were also covered in a general Ptolemy tutorial[7] given at ESWeek 2010.

4.1.4 Autocoding Embedded Software for Safety Critical Systems (Lee)

Code Generation

During this period, we enhanced the code generator so that we can generate code that calls user-defined actors for which we do not have code generation templates. This feature allows the same user-generated code to be used in development and in deployment.

In addition, we modified the code generator so that it can handle very large models and generate code that can be successfully compiled. Our test case is a propietary model consisting of 19000 actors. Our code generated created a 96Mb Java file that we were able to compile by implementing a number of tricks such as inner classes, static imports and using arrays instead of individual variables. This work was done as part of the Extensible Modeling and Analysis Framework (EMAF) for AFRL.

During this period we explored the theoretical ramifications of interface theory. This work is critical if we are to be able to prove that that the clustering of models is correct. Marc Geilen, visiting Berkeley from Eindhoven University of Technology, coauthored a paper with Stavros Tripakis (Berkeley) and Maarten Wiggers (Berkeley), "The Earlier the Better: A Theory of Timed Actor Interfaces" [8]

> "In a component-based design context, we propose a relational interface theory for synchronous systems. A component is abstracted by its interface, which consists of input and output variables, as well as one or more contracts. A contract is a relation between input and output assignments. In the stateless case, there is a single contract that holds at every synchronous round. In the general, stateful, case, the contract may depend on the state, modeled as the history of past observations. Interfaces can be composed by connection or feedback. Parallel composition is a special case of connection. Feedback is allowed only for Moore interfaces, where the contract does not depend on the current values of the input variables that are connected (although it may depend on past values of such variables). The theory includes explicit notions of environments, pluggability and substitutability. Environments are themselves interfaces. Pluggability means that the closed-loop system formed by an interface and an environment is well-formed, that is, a state with unsatisfiable contract is unreachable. Substitutability means that an interface can replace another interface in any environment. A refinement relation between interfaces is proposed, that has two main properties: first, it is preserved by composition; second, it is equivalent to substitutability for well-formed interfaces. Shared refinement and abstraction operators, corresponding to greatest lower and least upper bounds with respect to refinement, are also

defined. Input-complete interfaces, that impose no restrictions on inputs, and deterministic interfaces, that produce a unique output for any legal input, are discussed as special cases, and an interesting duality between the two classes is exposed. A number of illustrative examples are provided, as well as algorithms to compute compositions, check refinement, and so on, for finite-state interfaces."

Stavros Tripakis, Ben Lickly, Tom Henzinger (EPLF) and Edward A. Lee's paper, "A Theory of Synchronous Relational Interfaces,"[9] was accepted to the ACM Transactions on Programming Languages and Systems (TOPLAS). The abstract for that paper is below:

"In a component-based design context, we propose a relational interface theory for synchronous systems. A component is abstracted by its interface, which consists of input and output variables, as well as one or more contracts. A contract is a relation between input and output assignments. In the stateless case, there is a single contract that holds at every synchronous round. In the general, stateful, case, the contract may depend on the state, modeled as the history of past observations. Interfaces can be composed by connection or feedback. Parallel composition is a special case of connection. Feedback is allowed only for Moore interfaces, where the contract does not depend on the current values of the input variables that are connected (although it may depend on past values of such variables). The theory includes explicit notions of environments, pluggability and substitutability. Environments are themselves interfaces. Pluggability means that the closed-loop system formed by an interface and an environment is well-formed, that is, a state with unsatisfiable contract is unreachable. Substitutability means that an interface can replace another interface in any environment. A refinement relation between interfaces is proposed, that has two main properties: first, it is preserved by composition; second, it is equivalent to substitutability for well-formed interfaces. Shared refinement and abstraction operators, corresponding to greatest lower and least upper bounds with respect to refinement, are also defined. Input-complete interfaces, that impose no restrictions

on inputs, and deterministic interfaces, that produce a unique output for any legal input, are discussed as special cases, and an interesting duality between the two classes is exposed. A number of illustrative examples are provided, as well as algorithms to compute compositions, check refinement, and so on, for finite-state interfaces."

Two other papers that relate to autocoding and model checking are:

- Dai Bui, Hiren Patel and Edward A. Lee, "Checking for Circular Dependencies in Distributed Stream Programs", [10] EECS Department, University of California, Berkeley. Technical Report No. UCB/EECS-2011-97, August 29, 2011.

- Stavros Tripakis, Christos Stergiou, Roberto Lublinerman, "Checking Non-Interference in SPMD Programs" [11] at the 2nd USENIX Workshop on Hot Topics in Parallelism (HotPar 2010).

Ontologies

The Ptolemy Hierarchical Orthogonal Multi-Attribute Solver (PtHOMAS) project (in conjunction with Bosch Research Center, Palo Alto) is focused on enhancing model-based design techniques with the ability to include in models semantic information about data (what the data means), to check consistency in the usage of data across models, and to optimize models based on inferences made about the meaning of the data.

Including semantic information in models helps to expose modeling errors early in the design process, engage a designer in a deeper understanding of the model, and standardize concepts and terminology across a development team. It is impractical, however, for model builders to manually annotate every modeling element with semantic properties. PtHOMAS demonstrates a correct, scalable and automated method to infer semantic properties using lattice-based ontologies, given relatively few manual annotations. Semantic concepts and their relationships are formalized as a lattice, and relationships within and between components are expressed as a set of constraints and acceptance criteria relative to the lattice. Our inference engine automatically infers properties wherever they are not explicitly specified. Our implementation leverages the infrastructure in the Ptolemy II type system to get efficient and scalable inference and consistency checking. We demonstrate the approach on a non-trivial Ptolemy II model of an adaptive cruise control system.

Our primary focus during this period has been making the ontology analyzer more powerful by enabling expression of more types of ontologies. There are two methods that we have pursued to allow this. The first method allows composition of existing ontologies to create new combination ontologies, while the second method allows expression of ontologies that are potentially infinite.

Composing multiple ontologies into a combination ontology allows us to create analyses that rely on interaction between multiple domains. Using this, we can do things like creating an ontology that infers both variable values and statement reachability, and uses the knowledge of variable values at conditional branches to determine that certain branches are unreachable.

Allowing infinite ontologies makes the ontologies much more powerful, in that it allows us to model much more fine-grained information within the ontology rather than just coarse abstractions. One example of an infinite ontology is one that can infer the values of variables no matter what the value is. Another completely different type of infinite ontology is one that handles structured recursive data-types, such as a record types. Another infinite ontology whose structure is very similar to that of a record is expression monotonicity. Since it is important in our analysis for all constraint expressions to be monotonic in order for our algorithm to guarantee a unique result, we have also worked on developing an analysis that determines the monotonicity of expressions. Based on the work "Static Monotonicity Analysis for λ-Definable Functions over Lattices" by Murawski and Yi[12], we have implemented this analysis itself within Ptolemy's ontology analyzer using the infinite monotonicity ontology.

4.1.5 Testing and Experimental Validation (Tomlin, Sastry, Lee, Karsai)

Controlling Timing with a deadline instruction

Executive Summary:

The Precision Timed (PRET) architecture introduces predictable timing and instruction-set extensions that provide precise timing control for hard real-time embedded systems.

We have enabled control over timing at the Instruction Set Architecture (ISA) level by introducing four so-called "deadline" instructions. As higher-level models of computation (MoC) often have a timed semantics, the deadline instructions can be used to implement such MoCs at the binary level,

thereby filling a void in the model-based development of high-confidence systems from high-level models to low-level realizations. To allow for a wide range of cost-efficient implementations of the PRET timed semantics, it can be parameterized in different architectural aspects, as for instance the sizes of different levels of the memory hierarchy. We have made significant progress in developing parametric timing analyses to support the verification of timing constraints on PRET.

On the implementation side, we have continued the development of a prototypical PRET processor called PTARM, implementing the four deadline instructions and providing predictable timing behavior. A new PRET DRAM controller provides access to large Dynamic RAMs at predictable and composable memory access times to the four hardware threads of the PTARM core. Compared with previous approaches, we improve the worst-case access latency of small requests by up to 73%.

Report:

Guaranteeing the correct behavior of embedded systems is extremely difficult, especially with respect to timing constraints and their relationship to the safety of the physical systems. The traditional verification process is faced with two major challenges regarding timing constraints: 1. The execution time of a program depends on the hardware on which it is running. Every new hardware realization requires the development of new timing analysis tools. 2. The development process of such analysis tools is becoming increasingly error-prone and time-consuming for modern, complex hardware realizations.

PRET tackles both of these challenges. By introducing a timed semantics for the instruction set architecture, timing becomes a property of programs rather than a property of programs running on particular hardware realizations. As a consequence, programs can be ported from one realization to the next without having to recertify and develop new timing analysis tools. Furthermore, due to the simplicity of the timing model associated with the ISA, precise and efficient timing analysis becomes possible. We anticipate this work to entail a paradigm shift in the development of hardware realizations for embedded systems: instead of focusing on improving performance, new hardware realizations will be developed that optimize aspects such as energy and power consumption, implementation cost, and reliability.

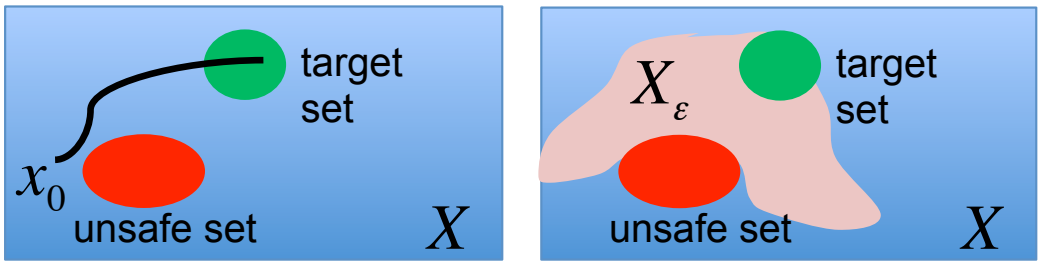
In the past year, we have taken the following steps towards the PRET vision:

- We have introduced four so-called "deadline" instructions, which introduce control over timing at the ISA level. These instructions allow to enforce upper and lower bounds on the execution time of blocks of code and provide the ability to act upon deadline misses. As higher-level models of computation (MoC) often have a timed semantics, the deadline instructions can be used to implement such MoCs at the binary level, thereby filling a void in the model-based development of high-confidence systems from high-level models to low-level realizations. This work [13] has been presented at DAC 2011.

- We have continued the development of a prototypical PRET processor called PTARM, which shall implement the four deadline instructions and provide predictable timing behavior [14].

- We have developed a new DRAM controller which provides predictable and composable memory access times to the four hardware threads of the PTARM core. Instead of viewing the DRAM device as one resource that can only be shared as a whole, our approach views it as multiple resources that can be shared between one or more clients individually. We partition the physical address space following the internal structure of the DRAM device, i.e., its ranks and banks, and interleave accesses to the blocks of this partition. This eliminates contention for shared resources within the device, making accesses temporally predictable and temporally isolated. Compared with previous approaches, we improve the worst-case access latency of small requests by up to 73%. This work [13], has been presented at CODES+ISSS 2011.

- To allow for a wide range of cost-efficient implementations of the PRET timed semantics, it can be parameterized in different architectural aspects, as for instance the sizes of different levels of the memory hierarchy and their respective latencies, or the latencies floating-point instructions. Supporting such a parameterized timed semantics requires parametric timing analysis. Parametric timing analysis derives a predicate that is satisfied for choices of parameter values that are guaranteed to meet all timing constraints. This summer, we have made significant progress in developing such parametric analyses.



(a) Reach-Avoid problem (b) Probabilistic backward reach-avoid set

Figure 5: Reach-avoid problem for stochastic hybrid systems

4.2 Embedded Systems Modeling and Deep Compositionality (Krogh, Tomlin, Sastry)

4.2.1 Verification of Stochastic Hybrid Systems

We consider a general hybrid modeling framework in which we account for stochastic disturbances in evolution of the continuous and discrete states. In addition, we account for deterministic disturbances in the model. The motivation is that while some classes of uncertainty, such as those by nature, are best modeled stochastically, some other classes of uncertainty, such as those due to presence of agents with competing objectives, are best modeled in a deterministic worst-case approach. For example, in a collision avoidance scenario between two aircraft, on the one hand, wind affects the dynamics of aircraft and uncertainties in wind forecast and measurement may be best accounted for through a stochastic framework. On the other hand, in the absence of communication between the aircraft, from the perspective of each aircraft, the trajectory must be safe in the worst-case performance of the other aircraft. Hence, a robust approach should be considered.

The reach-avoid objective in the setting of the stochastic hybrid system with two players becomes a stochastic game in which the objective of player 1 (the control) is to steer the hybrid system state into a desired target set, while avoiding a set of unsafe states, as shown in Figure 5(a). On the other hand, the objective of player 2 (the adversary) is to either steer the state into the unsafe set or prevent it from reaching the target set.

Mathematically, the problem is stated as follows: Let X denote the hybrid state space. Suppose that Borel sets $G, S \in \mathcal{B}(X)$ are given as the desired

target set and safe set, respectively, with $G \subseteq S$. Then the probability that the state trajectory $(x_0, x_1, \ldots, x_N)$ reaches G while staying inside S under fixed choices of player 1 policy $\mu \in \mathcal{M}_a$ and player 2 strategy $\gamma \in \Gamma_d$ is [15]:

$$r_{x_0}^{\mu,\gamma}(G,S) = E_{x_0}^{\mu,\gamma}\left[\mathbf{1}_G(x_0) + \sum_{j=1}^{N}\left(\prod_{i=0}^{j-1}\mathbf{1}_{S\setminus G}(x_i)\right)\mathbf{1}_G(x_j)\right],$$

where $E_{x_0}^{\mu,\gamma}$ denotes the expectation with respect to the probability measure $P_{x_0}^{\mu,\gamma}$ induced by the initial condition, $x_0 \in X$, and the players' strategies. The admissible control spaces consist of the set of Markov policies and strategies, denoted by $\mathcal{M}_a$, Γ_d, for the control and adversary, respectively.

Define the worst-case reach-avoid probability under a choice of Markov policy μ as $r_{x_0}^{\mu}(G,S) = \inf_{\gamma\in\Gamma_d} r_{x_0}^{\mu,\gamma}(G,S)$. Our objective is to maximize this worst-case probability over the set of Markov policies. Thus, we need to compute the maxmin value function $r_{x_0}^{*}(G,S) := \sup_{\mu\in\mathcal{M}_a} r_{x_0}^{\mu}(G,S)$, and find a maxmin policy $\mu^* \in \mathcal{M}_a$, such that $r_{x_0}^{*}(G,S) = r_{x_0}^{\mu^*}(G,S)$, $\forall x_0 \in X$.

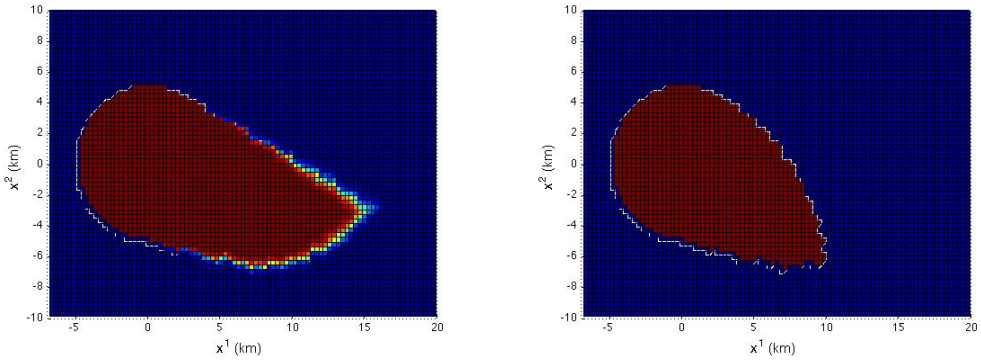
We have developed a dynamic programming algorithm for maximizing the reach-avoid probability and for synthesizing a control law that achieves this probability. In addition, from this algorithm we can find the set of initial conditions X_ϵ for which the reach-avoid probability is above $(1-\epsilon)$, $\forall\epsilon \in [0,1]$, under the worst-case adversary behavior, as shown in Figure 5(b).

The algorithm has been applied to several robust motion planning problems including a quadrotor helicopter tracking a ground vehicle and aircraft conflict detection and resolution scenarios [15].

Aircraft Conflict Detection

The scenario involves two aircraft with possibly intersecting nominal trajectories. From the perspective of the first aircraft, the task is to detect the possibility of conflict given current position of another aircraft, and design a collision avoidance trajectory in case potential conflict is detected. Motivated by wind influence on aircraft trajectories and on accuracy of conflict detection, we consider wind with a deterministic component, known through forecast or measurements, and a stochastic component to capture its uncertainties. Based on geostatistics analysis of wind data, the stochastic wind component is modeled as a time dependent random field over the $2D$ airspace.

A conflict is defined if aircraft get closer than a critical distance of R_c. Hence, the safe set in $2D$ can be defined in relative coordinates as: $S = \{(x^1, x^2) \in \mathbb{R}^2 \; s.t. \; \|(x^1, x^2)\|_2 \geq R_c\}$.For conflict detection, we assume that



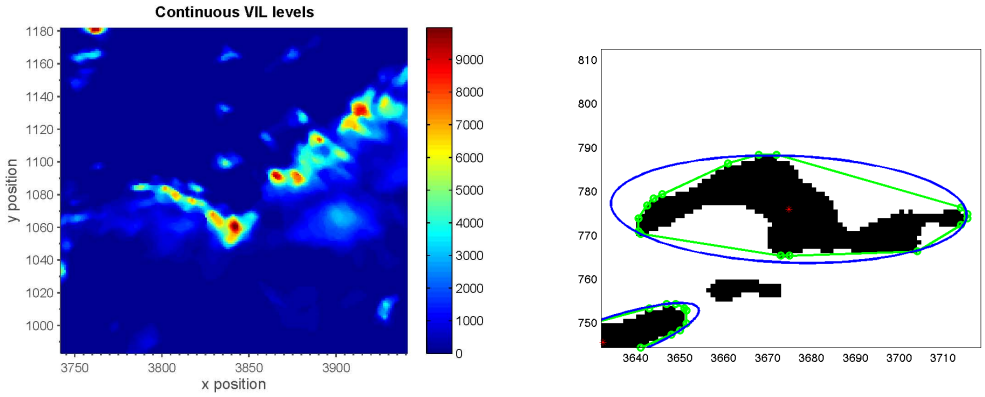
(a) Maxmin probability of safety (b) Deterministic reachable set

Figure 6: Collision detection in probabilistic and deterministic settings

the current position of each aircraft is available, for example through Automatic Dependent Surveillance-Broadcast (ADS-B) network. For the conflict resolution, we assume that the control of the two aircraft are decentralized. Furthermore, in the absence of further information on the decision algorithms, each aircraft assumes that the other aircraft could potentially make choices that endanger safety.

The aircraft kinematics is modeled as a unicycle. The input of each aircraft is its heading angle rate. Motivated by discrete maneuvers used in air traffic, we assume at any given time, each aircraft can choose to be in one of three flight maneuvers: straight, right turn, or left turn.

The maxmin probability of safety over a horizon of 2.5 minutes and with aircraft speed of 5 km per minute, is computed based on our proposed dynamic programming algorithm. Figure 6(a) illustrates this probability for the set of initial conditions with relative heading of $\frac{3\pi}{4}$ radians. The interpretation of this probability map is as follows. Consider an initial condition of (10.55 km, -6.85 km, $\frac{3\pi}{4}$ rad). From the value function we obtain the maxmin probability of safety to be 99%. This means that if aircraft 1 selects flight maneuvers according to the maxmin policy μ^* and aircraft 2 selects any maneuvers within the set of Markov strategies, the probability of collision would remain at most 1%. For comparison, the result of deterministic computation, in which wind influence is ignored, is shown in Figure 6(b). In this case, any initial condition is characterized as being safe or not.



(a) VIL measurements from forecast

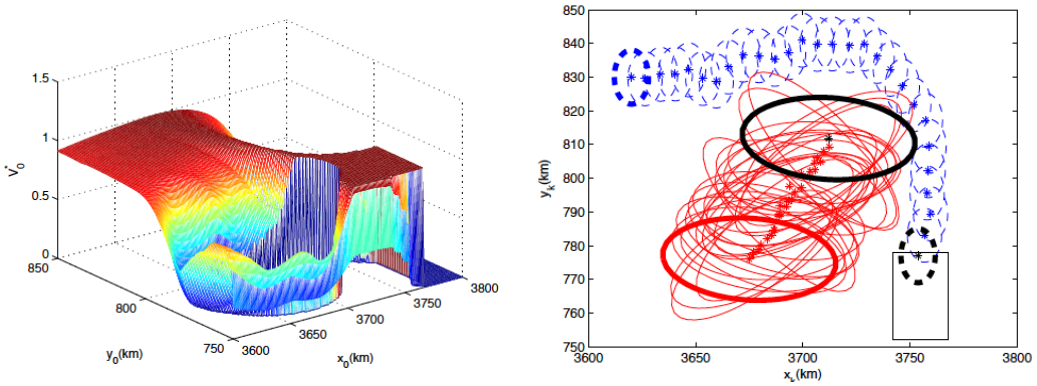
(b) No-fly zones based on forecast

Figure 7: Forecast data for Vertically Integrated Liquid level

Uncertain Safe and Target Sets

It has been shown that high values of Vertically Integrated Liquid (VIL) level indicate storm precipitation and regions with high VIL levels should be avoided by pilots. We have used MIT Lincoln lab forecast of VIL data over a gridded United States airspace to characterize obstacles in aircraft flight. Figure 7(a) shows the forecast data for 01/07/2009 near gulf coast of Florida, while Figure 7(b) shows the minimum-volume geometric shapes enclosing the regions with high VIL values in order to use them in an optimization algorithm. In [16] we used a deterministic hybrid optimal control approach to find fuel efficient aircraft trajectories which avoid these hazardous regions. Also, based on the actual and forecast data, we modeled the storm movement over time and formulated a receding horizon nonlinear program to design conflict-free aircraft trajectories which avoid moving storms [17]. Clearly, the actual weather deviates from the forecast, specially, with increasing forecast horizon. To account for such environmental uncertainties, we introduced a parametrized set-valued stochastic process model for the stochastic target and safe sets in the reach-avoid problem. In [18] we showed that the verification and control synthesis for stochastic hybrid systems with stochastic sets can be addressed by an appropriate dynamic programming algorithm.

We used our methodology to optimize the probability that the aircraft attains a rectangular region around a waypoint shown in Figure 8(b), while



(a) Probability map for $\psi_0 = -0.785$ (b) Maximally safe trajectory

Figure 8: Aircraft trajectory planning in stochastic environment

avoiding the stochastic unsafe sets, representing the storm locations, over the 30 minute horizon and to synthesize an optimal Markov policy that achieves this probability. The optimal value function, is shown in Figure 8(a) for all initial positions (x_0, y_0) in $2D$ with an initial heading angle of $\psi_0 = -0.785$ radians. An example execution of the process is shown in Figure 8(b).

5 Consultative and advisory functions to other laboratories and agencies, especially Air Force and other DoD laboratories. Provide factual information about the subject matter, institutions, locations, dates, and names(s) of principal individuals involved

5.1 Edward A. Lee

5.1.1 Air Force Research Laboratory, AFRL/RIEA, Rome, NY

Brian Romano USAF AFMC /AFRL/RIEF, Brian.Romano@rl.af.mil

The objective of the Extensible Modeling and Analysis Framework (EMAF) effort is to build on top of Ptolemy II and adapt Ptolemy II for the rapid

construction and configuration of modeling and analysis systems that incorporate disparate technologies. The purpose of this gap-filling project is to develop technologies for future incorporation into large-scale modeling and analysis systems, with specific focuses on scalable algorithm description, composition of heterogeneous components, and synthesis of efficient deployable decision-support systems that exploit multicore and distributed computing platforms. In particular, we have applied the code generation infrastructure developed under this MURI to a very large problem consisting of roughly 13000 actors. We were able to reduce the run time from roughly 10 minutes to 3 seconds.

5.1.2 US Army Research Laboratory

Scalable Composition of Subsystems (SCOS)

Chris Winslow, winslow@arl.army.mil

The Scalable Composition of Subsystems (SCOS) project, was funded by the Army Research Office in connection with the OSD Software-Intensive Systems Producibility Initiative. The mission was to research scalable techniques in software engineering based upon the concepts inherent in modelbased composition. The overarching goal was to show that these techniques will result in predictable and understandable behaviors in Systems-of-Systems (SoS) environments. The focus was on interaction between components (rather than the conventional focus on transformation of data), and on composition, which in this domain needs to be intrinsically concurrent (rather than the conventional thread-based applique of concurrency on imperative models).

Disciplined Design of Systems of Systems (DDoSoS)

Chris Winslow, winslow@arl.army.mil

The Disciplined Design of Systems of Systems (ddosos) is sponsored by the Army Research Laboratory (ARL) The project covers these areas: Multiform Models of Time, Temporal Isolation, Hybrid Models, Correct Composition, Linking Behaviors to Implementation and Design Drivers.

5.1.3 Lawrence Berkeley National Laboratory

Michael Wetter, MWetter@lbl.gov

Researchers at the Lawrence Berkeley National Laboratory have developed the BCVTB: "The Building Controls Virtual Test Bed is a software environment that allows expert users to couple different simulation programs

for co-simulation, and to couple simulation programs with actual hardware. For example, the BCVTB allows to simulate a building in EnergyPlus and the HVAC and control system in Modelica, while exchanging data between the software as they simulate. The BCVTB is based on the Ptolemy II software environment. The BCVTB allows expert users of simulation to expand the capabilities of individual programs by linking them to other programs. Due to the different programs that may be involved in distributed simulation, familiarity with configuring programs is essential."

5.1.4 Various Universities

Kepler: A System for Scientific Workflows, is a cross-project collaboration to develop open source tools for Scientific Workflow Management and is currently based on the Ptolemy II system for heterogeneous concurrent modeling and design.

The Kieler Project at the University of Kiel is "a research project about enhancing the graphical model-based design of complex systems. The basic idea is to consistently employ automatic layout to all graphical components of the diagrams within the modeling environment. This opens up new possibilities for diagram creation and editing on the one hand and also new methods for dynamic visualizations of e.g. simulation runs on the other hand. Hence the focus of this project is the pragmatics of model-based system design, which can improve comprehensibility of diagrams, shorten development and change actions, and improve the analysis of dynamic behaviour."

5.2 Claire Tomlin

- Member, Science Advisory Board, NSF Institute for Pure and Applied Mathematics, UCLA, 2009-present. Methods in control, reachability, stochastic control systems stemming from this grant.
- Member, Steering Committee for Cyber-Physical Systems, 2009-present.
- Co-Chair (with Prof. John Hansman, MIT) of the Federal Networking and Information Technology Research and Development (NITRD) High Confidence Systems and Software Committee Study on Software for Safety Critical Aviation Systems, 2005-2007

5.3 Technology Assists, Transitions, and Transfers.

5.3.1 Ptolemy II 8.0

Ptolemy II 8.0.1 [19] shipped on October, 28, 2010. New work in Ptolemy II 8.0.1 includes:

- Model Transformation - a framework for the analysis and transformation of actor models using model transformation techniques
- Ptera (Ptolemy Event Relationship Actor) Domain
- Causality Analysis: Updates to our non-conservative causality analysis for modal models within discrete-event (DE) systems
- Continuous and Modal Domains: a substantial rework of modal models and the underlying finite state machine infrastructure to make them work predictably and consistently across domains.

5.3.2 Strategic Directions in Software At Scale

, In August 2010, the Office of Information Systems and Cyber Security (ISCS) within the Office of the Director, Defense Research and Engineering (DDR&E) sponsored the Strategic Directions in Software at Scale (SaS) Workshop [20]. The SaS Workshop was hosted by the University of California, Berkeley. The goals of the workshop were to:

- Identify new ideas and promising research directions in software engineering and computer science achievable in the short-, mid-, and long-term.
- Identify opportunities for collaboration and engage in rich intellectual exchange of technical ideas.
- Create a foundation for developing a DoD roadmap for SaS.
- Begin to build a case for increasing DoD investment in software engineering and computer science research to strengthen the DoDs software technology base.

  Fifteen invited speakers gave presentations in the areas of software synthesis, robust and continuous behavior, temporal semantics, scalable

composition, and software engineering process and methodology. Each speaker advocated a particular technical approach that could be the basis for a Strategic Direction in future software research. To capture the quality and promise of the technical approaches, attendees were asked to rate each presentation with respect to six evaluation criteria.

The overall best technical approaches, as assessed by the attendees, were "Temporal Semantics in Concurrent and Distributed Software"Edward Lee, "Is Distributed Consistency Scalable?"Ken Birman, and "The Effect of Software (and Communication) Reliability and Security on Control Systems"Bruno Sinopoli.

5.3.3 Ptolemy Miniconference

The Ninth Biennial Ptolemy Miniconference was held on February 16, 2011. We had 70 attendees from around the world.

The Ptolemy Miniconference is an opportunity for research collaborators and Ptolemy users and extenders from industry, academia, and government to get together, present their work to the Ptolemy community, and hear about related research and results. Presentations and posters from the conference may be found at http://ptolemy.org/conferences/11.

5.4 Honors and Awards

Awards

- Tomlin. Charles A. Desoer Chair, College of Engineering, UC Berkeley, July 2011.
- Tomlin. Distinguished Teaching Award, Electrical Engineering, UC Berkeley, 2010-2011.
- Tomlin. IEEE Fellow, 2010.
- Tomlin. AIAA Associate Fellow, 2010.
- Tomlin. Tage Erlander Guest Professorship, Swedish Research Council, 2009-2010.

Keynotes Talks

- Edward A. Lee, "Disciplined Heterogeneous Modeling," [34] Invited Keynote Talk, ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2010), Oslo, Norway, October 6-8, 2010.

- Edward A. Lee, "An Introductory Textbook on Cyber-Physical Systems" [35] Invited Keynote Address, Workshop on Embedded Systems Education (WESE), organized as part of ESWEEK, Scottsdale, AZ, October 28, 2010.

- Edward A. Lee, "Synthesis of Reliable Distributed Real-Time Software" [36] Invited Keynote Talk, Workshop on Software Synthesis, ESWEEK 2010, Scottsdale, October 29, 2010.

- Edward A. Lee, "Synthesis of Distributed Real-Time Embedded Software," [37] Keynote talk, Electronic System Level Conference, San Diego, California, June 5-6, 2011.

- Edward A. Lee. "Computing Needs Time," [38] 2, November, 2010; Invited presentation at the Working Day on Time-oriented Embedded Systems at ENSEEIHT, Toulouse, France.

- Edward A. Lee. "Time for High-Confidence Software Systems," [39] Keynote Talk, 11th Annual Conference on High Confidence Software and Systems, Annapolis, Maryland, May 1-6, 2011.

- Claire Tomlin. Plenary Speaker, European Commission: European Future Technologies Conference and Exhibition, Budapest, May 2011.

- Claire Tomlin. Plenary Speaker, Institute for Pure and Applied Mathematics (UCLA) 10th Anniversary Symposium, November 2010.

Invited talks

- Edward A. Lee. "Predictability, Repeatability, and Models for Cyber-Physical Systems," [40] 24, October, 2010; Invited talk, Workshop on Foundations of Component Based Design (WFCD) at ESWeek 2010, Scottsdale, AZ.

- "Architecture for Precise and Repeatable Timing," Thales workshop, Palaiseau, France, November 3, 2010.

- "Programming Models for Parallel and Distributed Real-Time Systems," Thales workshop, Palaiseau, France, November 3, 2010.
- "Computing Needs Time," [41] Distinguished Lecture Series on Cyber-Physical Systems, Washington University, St. Louis, November 12, 2010.
- "Computing Needs Time," [42] The Distinguished Systems Speakers Series, Purdue University, December 6, 2010.
- "Computing Needs Time," [43] The Center for Embedded Systems - Distinguished Lecture Series, Virginia Tech, December 10, 2010.
- Claire Tomlin. Booz-Allen-Hamilton Distinguished Colloquium, ECE Department, University of Maryland. October 2011.

References

[1] Jerry Ding, Jonathan Sprinkle, S. Shankar Sastry, and Claire J. Tomlin. Reachability calculations for automated aerial refueling. In 47th IEEE Conference on Decision and Control, pages 3706–3712, Dec. 2008.

[2] Jerry Ding, Jonathan Sprinkle, Claire J. Tomlin, S. Shankar Sastry, and James L. Paunicka. Reachability calculations for vehicle safety during manned/unmanned vehicle interaction. AIAA Journal of Guidance, Control, and Dynamics, 2011. Accepted for publication.

[3] Jerry Ding and Claire J. Tomlin. Robust reach-avoid controller synthesis for switched nonlinear systems. In 49th IEEE Conference on Decision and Control, pages 6481–6486, Dec. 2010.

[4] Jerry Ding, Eugene Li, Haomiao Huang, and Claire J. Tomlin. Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 2160–2165, May 2011.

[5] Edward A. Lee and Stavros Tripakis. Modal models in Ptolemy. In Proceedings of 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (EOOLT 2010), pages 1–11, October 2010. http://chess.eecs.berkeley.edu/pubs/700.html.

[6] Edward A. Lee and Stavros Tripakis. Modal models in Ptolemy, October 2010. Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, in conjunction with MODELS. http://chess.eecs.berkeley.edu/pubs/708.html.

[7] Christopher Brooks, Edward A. Lee, and Stavros Tripakis. Exploring models of computation with Ptolemy II, October 2010. Tutorial, ESWeek 2010, Scottsdale, AZ. http://chess.eecs.berkeley.edu/pubs/712.html.

[8] Marc Geilen, Stavros Tripakis, and Maarten Wiggers. The earlier the better: A theory of timed actor interfaces. In 14th International Conference on Hybrid Systems: Computation and Control (HSCC'11), April 2011.

[9] Stavros Tripakis, Ben Lickly, Tom Henzinger, and Edward A. Lee. A theory of synchronous relational interfaces. Technical Report UCB/EECS-2010-45, UC Berkeley, April 2010. Revised version of this report to appear in ACM TOPLAS.

[10] Dai Bui, Hiren Patel, and Edward A. Lee. Checking for circular dependencies in distributed stream programs. Technical Report UCB/EECS-2011-97, EECS Department, University of California, Berkeley, Aug 2011.

[11] Stavros Tripakis, Christos Stergiou, and Roberto Lublinerman. Checking non-interference in spmd programs. In 2nd USENIX Workshop on Hot Topics in Parallelism (HotPar 2010), pages 1–6, June 2010.

[12] Andrzej S. Murawski and Kwangkeun Yi. Static monotonicity analysis for λ-definable functions over lattices. In Revised Papers from the Third International Workshop on Verification, Model Checking, and Abstract Interpretation, VMCAI '02, pages 139–153. Springer-Verlag, 2002. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=?doi=10.1.1.15.2537.

[13] Jan Reineke. Pret dram controller: Bank privatization for predictability and temporal isolation, October 2011.

[14] Isaac Liu, Jan Reineke, and Edward A. Lee. A pret architecture supporting concurrent programs with composable timing properties. In

44th Asilomar Conference on Signals, Systems, and Computers, pages 2111–2115, November 2010.

[15] M. Kamgarpour, J. Ding, S. Summers, , A. Abate, J. Lygeros, and C. Tomlin. Discrete Time Stochastic Hybrid Dynamic Games: Verification & Controller Synthesis. In Proceedings of IEEE Conference on Decision and Control, Dec 2011. to appear.

[16] M. Kamgarpour, M. Soler, C. Tomlin, A. Olivares, and J. Lygeros. Hybrid optimal control for aircraft trajectory design with a variable sequence of modes. In Proceedings of IFAC World Congress, August 2011.

[17] M. Kamgarpour, V. Dadok, and C. Tomlin. Trajectory Generation for Multiple Aircraft Subject to Dynamic Weather Uncertainty. In Proceedings of IEEE Conference on Decision and Control, 2010.

[18] Sean Summers, Maryam Kamgarpour, John Lygeros, and Claire Tomlin. A Stochastic Reach-Avoid Problem with Random Obstacles. In Hybrid Systems: Computation and Control, pages 251–260. ACM, 2011.

[19] Edward A. Lee, Christopher Brooks, Thomas Huining Feng, Jackie Man-Kit Leung, Bert Rodiers, Kyungmin Bae, Chad Berkley, Chihhong Cheng, Teale Fristoe, Shanna-Shaye Forbes, Hauke Fuhrmann, Ben Lickly, Isaac Liu, and Michael Wetter. Ptolemy II 8.0.1: An open-source software framework supporting experimentation with actor-oriented design, October 2010. http://chess.eecs.berkeley.edu/pubs/713.html.

[20] Michael J. May, Edward A. Lee, and Lindsay E. Jones. Strategic directions in software at scale. Technical Report DDRE-ISCS-2010-1, Office of Information Systems & Cyber Security (ISCS), Office of the Director, Defense Research & Engineering (DDR&E), November 2011. The presentations from the workshop may be found at http://chess.eecs.berkeley.edu/conferences/10/SDISAS/.

[21] Edward A. Lee. Ptolemy miniconferences, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[22] Jianwu Wang, Daniel Crawl, Ilkay Altintas, Chad Berkley, and Matt Jones. Distributed execution architectures in kepler, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[23] Patricia Derler, Jia Zou, Slobodan Matic, and John Eidson. Modeling distributed real-time systems with ptolemy ii, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[24] Stavros Tripakis and Edward A. Lee. Semantics of modal models in ptolemy, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[25] Charles Shelton, Elizabeth Latronico, and Ben Lickly. Static analysis using the ptolemy ii ontologies package, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[26] Jan Reineke, Isaac Liu, Gage W. Eads, Stephen A. Edwards, Sungjun Kim, and Hiren D. Patel. To meet or not to meet the deadline, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[27] Shuvra S. Bhattacharyya. The dataflow interchange format: Towards co-design of dsp-oriented dataflow models and transformations, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[28] Sven Koehler, Bertram Ludaescher, and Timothy McPhillips. Workflow recovery for different models of computation and models of provenance, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[29] Kaushik Ravindran and Murali Parthasarathy. Design, analysis, and implementation of static dataflow models for hardware targets, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[30] Gongjing Cao, Lei Dou, Quinn Hart, and Bertram Ludaescher. Kepler/g-pack: A kepler package using the google cloud for interactive scientific workflows, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[31] Anne Ngu and George Chin Jr. Context aware actors, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[32] Dai Nguyen Bui, Stavros Tripakis, Marc Geilen, Bert Rodiers, and Edward A. Lee. Modular code generation, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[33] Edward A. Lee. The ptolemy project: Advancing system design, February 2011. Presented at the Ninth Biennial Ptolemy Miniconference, Berkeley, CA.

[34] Edward A. Lee. Disciplined heterogeneous modeling. In O. Haugen D.C. Petriu, N. Rouquette, editor, Proceedings of the ACM/IEEE 13th International Conference on Model Driven Engineering, Languages, and Systems (MODELS), pages 273–287. IEEE, October 2010. http://chess.eecs.berkeley.edu/pubs/679.html.

[35] Edward A. Lee. An introductory textbook on cyber-physical systems, October 2010. Invited Keynote Address, Workshop on Embedded Systems Education (WESE), at ESWeek 2010, Scottsdale, AZ. http://chess.eecs.berkeley.edu/pubs/767.html.

[36] Edward A. Lee. Synthesis of reliable distributed real-time software, October 2010. Invited Keynote Talk, Workshop on Software Synthesis, ESWeek 2010, Scottsdale, AZ. http://chess.eecs.berkeley.edu/pubs/795.html.

[37] Edward A. Lee. Synthesis of distributed real-time embedded software, June 2011. Keynote talk, Electronic System Level Synthesis Conference, June 5-6, 2011, San Diego, California.

[38] Edward A. Lee. Computing needs time, November 2010. Invited presentation at the Working Day on Time-oriented Embedded Systems at ENSEEIHT, Toulouse, France. http://chess.eecs.berkeley.edu/pubs/792.html.

[39] Edward A. Lee. Time for high-confidence software systems, May 2011.

[40] Edward A. Lee. Predictability, repeatability, and models for cyber-physical systems, October 2010. Invited talk, Workshop on Foundations of Component Based Design (WFCD), at ESWeek 1010, Scottsdale, AZ. http://chess.eecs.berkeley.edu/pubs/798.html.

[41] Edward A. Lee. Computing needs time, November 2010. Distinguished Lecture Series on Cyber-Physical Systems, Washington University, St. Louis. http://chess.eecs.berkeley.edu/pubs/793.html.

[42] Edward A. Lee. Computing needs time, December 2010. Presented at the Distinguished Systems Speakers Series, Purdue University. http://chess.eecs.berkeley.edu/pubs/796.html.

[43] Edward A. Lee. Computing needs time, December 2010. Presented at the Center for Embedded Systems – Distinguished Lecture Series, Virginia Tech. http://chess.eecs.berkeley.edu/pubs/797.html.