ExCAPE Report of Activities (May 2012 – March 2013)

1 Overview

ExCAPE proposes a novel approach to software design—*computer augmented program engineering*, in which a programmer and an automated program-synthesis tool collaborate to generate software that meets its specification. A programmer expresses her insights about the design using synthesis artifacts of different kinds such as programs that may contain ambiguities, declarative specifications of high-level requirements, positive and negative examples of desired behaviors, and optimization criteria for selecting among alternative implementations. The synthesis tool composes these different views about the structure and functionality of the system into a unified concrete implementation using a combination of algorithmic techniques such as decision procedures for constraint-satisfaction problems, iterative schemes for abstraction and refinement, and data-driven learning. Our goal is to develop the theory and practice of the proposed approach into a transformative software design paradigm with the promise of a more reliable software at a lower cost. To achieve this goal, our team brings together expertise in theoretical foundations (computer-aided verification, control theory, program analysis), design methodology (human-computer interaction, model-based design, programming environments), and applications (concurrent programming, network protocols, robotics, system architecture).

Progress during the first year of the Expeditions is discussed in this report, which is organized along the research themes of *design methodology* (Section 2), *computational engines* (Section 3), *challenge problems* (Section 4), *tools and evaluation* (Section 5), and *education and knowledge transfer* (Section 6). The list below includes key activities that highlight the collaborative spirit of our project:

- Researchers at Cornell, Rice, UC Berkeley, and UCLA, are collaborating on synthesis for robots, advancing both the theory and tools for reactive synthesis, in the context of a specific challenge problem of a programmable robotic waiter.
- Building upon the technology of the tool Sketch for synthesis of programs (a joint project between MIT and UC Berkeley), researchers at Penn have designed the system Transit for specification and synthesis of distributed protocols, and demonstrated its benefits for design of industrial-strength cache-coherence protocols.
- Researchers at Michigan, Rice, and UC Berkeley are collaborating on identifying the synergies and differences between two different synthesis frameworks, namely, reactive synthesis and supervisory control.
- Researchers at MIT, Penn, and UC Berkeley are collaborating to formalize and standardize the core computational problem of synthesis of straight-line programs, which will be used to organize a repository of benchmarks and a synthesis competition.
- Researchers at MIT, Penn, and UC Berkeley have developed novel application of the synthesis technology to the problem of automated grading and feedback for online education, which is being deployed in classrooms.
- ExCAPE summer school, to be organized in June 2013, includes tutorials and lectures that span control theory and hybrid systems, temporal logics and automata theory, program synthesis and constraint solvers, and already has over 100 students registered.

2 Design Methodology

The first research theme, *design methodology*, is aimed at understanding how synthesis can be integrated in the software design process so as to decrease the cognitive load on the designers and the programmers. Key contributions over the past year include (1) enhancements to the Sketch synthesis tool to improve scalability and programmability, (2) a new project Rosette aimed at simplifying the task of building domain-specific synthesis tools, (3) a new collaborative effort to identify the commonalities between the subdisciplines of reactive synthesis and supervisory control in terms of methodology and computational solutions, (4) a framework for formal reasoning about transformations of programs represented in the open-source LLVM intermediate compiler representation, and (5) platform-based design methodology that allows constraints imposed by the underlying platform to be considered during the synthesis phase.

Modular Synthesis in Sketch
PI: Solar-Lezama (MIT)

In the programming approach advocated by the Sketch system, a programmer writes a partial program with incomplete details, and the synthesizer fills in the missing details using user-specified assertions as the correctness specification. Solar-Lezama's group is exploring different ways of enhancing the usability and scalability of Sketch. Recent work focuses on the problem of modular synthesis by relying on user-provided models of complex functions. The approach looks at how the user can write simple models to abstract complex functionality and how using such models can make computationally demanding synthesis problems tractable. Their research shows that introducing models can potentially interfere with the counterexample-guided-inductive-synthesis algorithm used by Sketch, and they propose a solution to overcome this challenge. Showing that an implementation is represented by a model is also tricky because it requires an extra quantifier to show that there exists a behavior of the model that matches the behavior of the implementation on a given input. However, their work shows how a few simple restrictions on the model can allow the quantifier to be eliminated without placing additional stress on the solver.

Infrastructure for Designing a Synthesizer
PI: Bodik (UC Berkeley)

Research scientist Emina Torlak, in collaboration with PI Bodik, is leading the effort on designing Rosette. Rosette is a high-level programming language with symbolic reasoning capabilities, designed to enable rapid scripting and prototyping of domain-specific synthesis tools. To prototype a synthesizer in Rosette, we first define the target programming model or an EDSL (embedded domain-specific language) by writing an interpreter for it. Next, we allow some constructs in the target language to be underspecified—that is, to produce symbolic, rather than just concrete, values. Rosette's symbolic reasoning engine then does the rest: executing a program in the target EDSL yields a symbolic encoding of the program's semantics, which is used to instantiate and discharge synthesis queries. You can think of Rosette as a lightweight Sketch with facilities for metaprogramming.

Bridging the Gap between Reactive Synthesis and Supervisory Control
PIs: Lafortune (Michigan), Tripakis (UC Berkeley), and Vardi (Rice)

Traditionally two relatively disjoint communities developed different frameworks for system synthesis: the subdiscipline of reactive synthesis and the subdiscipline of supervisory control for discrete-event systems. These frameworks solve similar problems, and this collaborative effort within the ExCAPE project tries to bridge the gap. The effort started during the ExCAPE kick-off meeting where Lafortune, Tripakis and Vardi decided to address the challenge of finding the synergies and the differences. Since then, progress has been made with remote conference calls, email exchanges and visits. Tripakis visited Lafortune for two days in September 2012 working almost exclusively on the problem. Lafortune visited Tripakis in March 2013 to continue work on this topic. An initial draft describing how the basic supervisory control problem can be reduced to a reactive synthesis problem has been prepared. A more complete technical report is under preparation. The plan is to present an initial version of this work at the upcoming ExCAPE Summer School in June 2013.

Verified LLVM Infrastructure
PIs: Martin (Penn) and Zdancewic (Penn)

PIs Steve Zdancewic and Milo Martin, along with their Ph.D. students, have developed Vellvm (verified LLVM), a framework for reasoning about programs expressed in the open-source LLVM intermediate compiler representation and program transformations that operate on it. Vellvm provides a mechanized formal semantics of LLVM's intermediate representation, its type system, and properties of its SSA form. The framework is built using the Coq interactive theorem prover. It includes multiple operational semantics and proves simulation relations among them to facilitate different reasoning styles and proof techniques.

Vellvm supports the activities of the ExCAPE project in two ways. First, it provides a foundational way of connecting the semantics of programs synthesized by high-level tools to their low-level implementations. As such, it offers the possibility of verifying the correctness of synthesized software with respect to a realistic low-level computation model – Vellvm can serve as a bridge between formal methods and main-stream compiler technology. Second, the Vellvm project itself draws on traditional *correctness by construction* techniques: verified Vellvm program transformations can be extracted directly from their proofs of correctness as expressed in Coq. This is itself a form of program synthesis, which is complementary to other approaches being explored in the ExCAPE research.

Platform-Based Design for Software Synthesis
PI: Sangiovanni-Vincentelli (UC Berkeley)

When synthesizing software from models for control applications, a number of constraints on latency, throughput, and calculation accuracy have to be satisfied. The key aspect of this problem is how to take into account the physical constraints on an abstract representation. The Turing abstraction indeed needs to be modified or even abandoned to do so. Computation time, power consumed and communication network characteristics of the platform that will be used for implementation have to be *inserted* while trying to maintain the flexibility of changing the platform later on if so demanded by the application. Platform-based design offers a framework where the abstract domain of algorithms and the physical aspects of the platform can be considered together for the task of software synthesis. Platform-based design is a meet-in-the-middle methodology where the lower level of the abstraction stack is characterized with macro parameters and the higher level of the stack is mapped onto the lower level. The lower level of the stack is parameterized with a library of components together with their interfaces and models. A platform instance is a selection of a set of components that can be composed according to appropriate rules. The selection of the components is carried out by evaluating the properties of the software mapped onto the platform. We are in the process of developing a complete flow for distributed systems with particular attention to the robotics application considered in ExCAPE.

3 Computational Engines

The research theme of *Computational Engines* explores common algorithmic foundations for formalizing and solving the different versions of the synthesis problem. A recently launched collaborative effort aims at defining the problem of *template-based synthesis modulo theories* in a rigorous and general way so that the community can share benchmarks and algorithmic solutions. ExCAPE PIs have advanced the algorithmic foundations for synthesis on multiple fronts including probabilistic systems, concurrent systems, linear control systems, hybrid systems, and quantitative analysis. Representative efforts are described in greater details below.

Template-based Synthesis Modulo Theories
PIs: Alur (Penn), Bodik (UC Berkeley), Martin (Penn), Seshia (UC Berkeley), and Solar-Lezama (MIT)

The goal of this collaborative effort is to define a logic-based core computational problem so that (1) different algorithmic strategies can be compared in an experimentally rigorous manner, (2) computational advances can be easily shared across ExCAPE tools (such as Sketch and Transit), and (3) it can form the basis of ExCAPE synthesis competition. Based on teleconferences over the last few months, we have now formalized the problem. The proposed syntax for problem specification extends the SMT-LIB2 language. Roughly speaking, the computational problem is the following: given a combination of logical theories supported by SMT solvers, the input consists of (1) a set of typed function symbols $f_1, \ldots f_k$, (2) a (first-order) formula φ that uses symbols from the underlying SMT theory along with the function symbols $f_1, \ldots f_k$, and (3) for $1 \leq i \leq k$, a context-free grammar G_i for each function symbol f_i which serves as the syntactic template for synthesizing the expression for f_i. The synthesis problem is to compute, for $1 \leq i \leq k$, an expression e_i from the set of terms generated by the grammar G_i, such that the formula obtained by substituting each function symbol f_i by e_i in φ, is valid.

Our next goal is to generate a comprehensive set of benchmarks in this proposed format using ExCAPE projects involving protocol synthesis, automatic program grading, and synthetic biology. Then, we plan to adapt and implement existing computational solutions (that are currently deeply integrated within tools such as Sketch and Transit) to this new format, and evaluate their relative merits and limits. The resulting report will form the basis of the announcement of the ExCAPE competition (to be held in Summer 2014). We believe that such a competition will motivate a lot of researchers to improve computational engines.

Synthesis of Logic for Avoidance of Concurrency Bugs
PI: Lafortune (Michigan)

PI Lafortune's group has been developing novel techniques for synthesis of deadlock-avoidance control logic for multi-threaded programs using the theory of discrete-event systems. New work on synthesis of deadlock-avoidance specifically focuses on: (i) synthesis technique that employs SAT-solvers to achieve greater scalability; and (ii) on atomicity enforcement using Petri-net-based techniques. This work is in collaboration with researchers at HP Labs, and has been presented at CDC (Conference on Decision and Control) 2012 in the paper titled *On Atomicity Enforcement in Concurrent Software via Discrete Event Systems Theory* by Wang, Liu, Kelly, Lafortune, Reveliotis, and Zhang. Work on synthesis in supervisory control problems for partially-observed systems, specifically of sensor activation policies and of opacity-enforcing insertion functions, by Lafortune and his graduate students at Michigan appears in another paper presented at CDC 2012.

Component-based Reactive Synthesis for Probabilistic Systems
PI: Vardi (Rice)

Synthesis from components is the automated construction of a composite system from a library of reusable components such that the system satisfies the given specification. This is in contrast to classical synthesis, where systems are always constructed from scratch. In the control-flow model of composition, exactly one component is in control at a given time and control is switched to another when the component reaches an exit state. The composition can then be described implicitly by a transducer, called a composer, which statically determines how the system transitions between components. Recently, it was shown that control-flow synthesis of deterministic composers from libraries of probabilistic components is decidable. In this work, we considered the more general case of probabilistic composers. We showed that probabilistic composers are more expressive than deterministic composers, and that the synthesis problem still remains decidable. In the near future, we will pursue how these theoretical results can help in improving the scalability of reactive synthesis for the robotics application domain.

Optimal Performance for Continuous-time Controllers
PI: Kress-Gazit (Cornell)

Recently, formal methods have been used to transform high-level robot tasks into correct-by-construction controllers. While correctness is guaranteed, these inherently discrete methods can often lead to behaviors that are not optimal in the continuous sense, i.e. they induce robot paths that are significantly suboptimal. This work proposes an algorithm for dynamically reordering the robot goals and connecting them with the shortest path based on the given continuous metric. The generated robot trajectories are close-to-optimal while satisfying the original task specification in a dynamic environment. This method is implemented and simulation results are shown. The results are being integrated in the tool LTLMoP to be applied to the Robotics challenge problem.

Synthesis of Controllers for Linear Systems
PI: Tabuada (UCLA)

In this work, we present and analyze a novel algorithm to synthesize controllers enforcing linear temporal logic specifications on discrete-time linear systems. The central step within this approach is the computation of the maximal controlled invariant set contained in a possibly non-convex safe set. Although it is known how to compute approximations of maximal controlled invariant sets, its exact computation remains an open problem. We provide an algorithm which computes a controlled invariant set that is guaranteed to be an under-approximation of the maximal controlled invariant set. Moreover, we guarantee that our approximation is at least as good as any invariant set whose distance to the boundary of the safe set is lower bounded. The proposed algorithm is founded on the notion of sets adapted to the dynamics and binary decision diagrams. Contrary to most controller synthesis schemes enforcing temporal logic specifications on continuous systems, we do not compute a discrete abstraction of the continuous dynamics. Instead, we abstract only the part of the continuous dynamics that is relevant for the computation of the maximal controlled invariant set. For this reason we call our approach specification guided. We describe the theoretical foundations and technical underpinnings of a preliminary implementation and report on several experiments including the synthesis of an automatic cruise controller. Our preliminary implementation handles up to five continuous dimensions and specifications containing up to 160 predicates defined as polytopes in about 30 minutes with less than 1GB memory.

Synthesis for Hybrid Systems
PIs: Kavraki (Rice) and Vardi (Rice)

We developed a novel computational method for the falsification of safety properties specified by syntactically safe linear temporal logic (LTL) formulas for hybrid systems with general nonlinear dynamics and input controls. The method is based on an effective combination of robot motion planning and model checking. Experiments on a hybrid robotic system benchmark with nonlinear dynamics show significant speedup over prior works.

Regular Functions for Quantitative Analysis
PI: Alur (Penn)

Theory of regular languages (of strings, infinite strings, and trees) has proved to be an excellent foundation for *qualitative* verification and synthesis of finite-state systems. The goal of this research thread is to develop a similarly robust foundation for *quantitative* analysis and synthesis. We propose a deterministic model for associating costs with strings that is parameterized by operations of interest (such as addition, scaling, and min), a notion of regularity that provides a yardstick to measure expressiveness, and study decision problems and theoretical properties of resulting classes of cost functions. Our definition of regularity relies on the theory of string-to-tree transducers, and allows associating costs with events that are conditional upon regular properties of future events. Our model of cost register automata allows computation of regular functions using multiple "write-only" registers whose values can be combined using the allowed set of operations. We show that classical shortest-path algorithms as well as algorithms designed for computing discounted costs, can be adopted for solving the min-cost problems for the more general classes of functions specified in our model. Cost register automata with min and increment give a deterministic model that is equivalent to weighted automata, an extensively studied nondeterministic model, and this connection results in new insights and new open problems.

4 Challenge Problems

To demonstrate the potential viability of the ExCAPE approach, and to guide the foundational research along the most promising directions, this theme focuses on representative *challenge problems*. We report on the progress on the four domains, namely, (1) multicore protocols, (2) networked systems, (3) robotic systems, and (4) development of applications for mobile platforms. Note that in the original proposal, the list of challenge problems was (1) multicore protocols, (2) networked systems, (3) robotic systems, and (4) concurrent programming. The substitution for the fourth application domain was triggered since some of the investigators identified new and exciting opportunities for applying ExCAPE synthesis techniques and tools to this new domain.

4.1 Multicore Protocols

Hardware communication and coordination protocols are the backbone of today's highly integrated Systems-on-Chip (SoC) designs, which are ubiquitous in mobile and embedded computing platforms. Even when employing the state-of-the-art design practices, designers are still called upon to create the entire design, including the most mundane but error-prone low-level aspects of the design, which arguably leads to tedious design and presence of bugs. The goal of this theme is to explore how synthesis can simplify and improve the design process for multicore protocols.

Methodology and Tool for Specifying Multicore Protocols
PIs: Alur (Penn) and Martin (Penn)

We have proposed a new way to program distributed protocols using *concolic snippets*, which are sample execution fragments that contain both concrete and symbolic values. This approach allows the programmer to describe the desired system partially using the traditional model of communicating extended finite-state-machines (EFSMs) along with high-level invariants and concrete execution fragments. Our tool derives a protocol implementation from a set of EFSM skeletons, which is analyzed using a model-checker with respect to the desired invariants. The programmer can add new concrete execution fragments to fix the counter-examples produced by the model-checker.

We show that (1) for a classical cache coherence protocol, our tool automatically generates a complete implementation from an EFSM skeleton and a few concrete examples for every transition, and (2) a published partial description of the SGI-Origin coherence protocol maps directly into symbolic examples and leads to a complete implementation in a few model-checker iterations.

This effort was inspired by the program synthesis tool Sketch and the work on programming by examples, and integrates a diverse range of ideas from ExCAPE team members. The initial report on this project, called Transit, will be presented in the upcoming meeting PLDI (ACM Conference on Programming Language Design and Implementation) 2013. We are planning to extend this tool in many ways, and use it as a case study for many ExCAPE research themes.

Synthesizing Distributed Algorithms using Transit
PIs: Alur (Penn), Martin (Penn), and Tripakis (UC Berkeley)

Distributed controller synthesis is in general undecidable both in the frameworks of reactive synthesis as well as of supervisory control. This is unfortunate as a number of interesting problems can be expressed as distributed controller synthesis problems. In particular, the problem of synthesizing the Alternating Bit Protocol (ABP), a basic communication protocol ensuring lossless communication over a lossy channel, can be stated as a distributed controller synthesis problem. The goal of this collaborative effort between PIs Alur, Martin, and Tripakis, is to study to what extent the Transit approach and tool (being developed by PIs Alur and Martin for synthesis of cache coherence protocols) can be used to synthesize a more broad class of distributed controllers. In particular, in collaboration with students Abhishek Udupa (Penn) and Antti Halme (Aalto University, Finland), we investigated to what extent Transit can be used to synthesize the ABP. The results are encouraging and suggest that this can be done as long as the number of states (alternating bit) are known in advance. A report describing this ongoing work is under preparation.

4.2 Robotic Systems

The ExCAPE postdoctoral researcher Ruediger Ehlers is coordinating various activities within ExCAPE on this theme to ensure synergistic progress. To establish a common ground for the collaborative research in the robotics domain of the ExCAPE project, we started by defining the robot waiter as a running scenario around which we center our research. The general idea of the scenario is to have a robot serving some plates to customers. We focus on the service-level aspect of the setting, that is, getting the robot to schedule the steps to be performed in an efficient and effective way, and abstract from robot arm handling, as this is an orthogonal problem that is already subject to intensive research outside of ExCAPE. We describe the tasks that the robot waiter needs to perform in a purely declarative manner and synthesize a controller for the robot.

In a series of teleconferences, we concretized the robot waiter scenario to a reference specification. At the same time, we collected the solution ideas for the robot waiter problem from the participating groups. PIs Kress-Gazit and Kavraki identified the main challenges of the problem for the current state-of-the-art in reactive synthesis technology and categorized which problems have priority as seen from the application side. Starting from this list, a couple on concrete research sub-projects were initiated as described below.

The research endeavors were complemented by work on the evaluating the current state-of-the-art in synthesis. Kai Weng Wong, a PhD student at Cornell, started to examine how far we can progress in the robot waiter scenario with currently available synthesis and motion planning tools. This work is performed in collaboration with the other members of Prof. Kress-Gazit's group. The results of this survey will be used in the next meeting of the collaborators in the ExCAPE robot application domain to identify and outline further concrete projects.

Robust Synthesis
PIs: Kress-Gazit (Cornell), Seshia (UC Berkeley), and Tabuada (UCLA)

To improve the robustness of the implementations computed by synthesis algorithms, we started a collaboration between the groups of Prof. Tabuada, Prof. Kress-Gazit, and Prof. Seshia. We established a layered methodology that keeps the complexity of the high-level decision making small, but can still achieve precision in robust actuation on a lower layer. This project is currently in the stage of evaluating the achievable performance on the lower layer. Collaboration has been performed by e-mail, phone calls, and by a visit of Ruediger Ehlers to Prof. Tabuada's group at UCLA.

Scalability
PIs: Kress-Gazit (Cornell) and Seshia (UC Berkeley)

To improve scalability and generality of synthesis technology, we examined the problem of synthesizing systems that can handle data. As a first work in this area, we used identifiers as a data domain. In the robot waiter context, these can, for example, identify the menu items that a customer ordered. We developed both the theory for describing such specifications and an algorithm for performing synthesis from such specifications. A publication, titled *Synthesis with identifiers*, that summarizes our results is currently under review.

Platform-based Design for Robot programming
PIs: Pappas (Penn), Sangiovanni-Vincetelli (UC Berkeley), and Tabuada (UCLA)

Platform-based design means that the design proceeds in a double direction: (A) bottom-up: this is the direction that allows the designer to build the platform library, choosing the abstractions of functionalities and performance from a given architectural space; (B) top-down: in this direction, the design composes the library elements in order to define a specific function. This is a mapping process. Usually, in the top-down approach, the designer tries to compose a specific solution in an iterative way. Once he has arranged a possible implementation, he evaluates the performance information that the platform components of the library provide at that level and, eventually, he/she proceeds with additional iterations until the implementation meets the desired performance. Always in the top-level direction, if there is the need of specific platform elements that are not available at that time, the designer is allowed to instantiate new platform blocks through the definition of placeholders, that are added to the library and implemented in the lower level of abstraction in a second time. In order to avoid large-loop iterations, it is possible to define more platforms at different layers of abstraction. This process, that bring at the definition of a platform stack, is the key of the PBD. In the restaurant waiter scenario, some steps have been taken to meet the PBD principles. The first step is to define the basic functionalities that the robot is equipped with. This requires extrapolation of some information from the physical level in order to build the library of components. These elements from the lowest level include some physical quantities that could be useful in the design process(for example, energy, precision, time). We are also working to find a "common language" for guaranteeing that all the synthesis groups can interface correctly with the motion planner.

Robot Plan Synthesis in Partially Known Environments
PIs: Kavraki (Rice), Kress-Gazit (Cornell), and Vardi (Rice)

PIs Kavraki, Kress-Gazit, and Vardi have initiated a collaboration to examine the problem of motion planning under a temporal specification for a hybrid robotic system with complex and nonlinear dynamics in a partially unknown environment. Our intention is to employ a multi-layered synergistic framework that can deal with general robot dynamics and combine it with an iterative planning strategy. The goal of this work is to deal with changes in the environment only when they are discovered and avoid repeating the work done up to that point for satisfying the temporal logic specification. We will demonstrate the efficacy of our framework on a hybrid second-order car-like robot moving in an office environment with unknown obstacles.

4.3 Networked Systems

Automated Cloud Configuration
PIs: Alur (Penn), Loo (Penn), and Parthasarathy (UIUC)

This recently started collaborative effort focuses on applying automatic synthesis solutions to the problem of cloud resource management, which has to be carried out in a safe manner that respects cloud operator deployment constraints and performance goals. PIs Alur and Loo are developing techniques for synthesizing migration strategies for virtual machines such that the performance objectives (for example, good load balancing) specified by the network operator can be met, while ensuring that the migrations of virtual machines still preserve the desired network-layer correctness properties (for example, some users should not be able to see the traffic from other users). PIs Parthasarathy and Loo have started a weekly teleconference to address the problem of safe migration of Software-defined Network (SDN) routers, where formal synthesis techniques are used to ensure that SDN routers can migrate from one configuration to another, while ensuring that intermediate states preserve correctness invariants of the network.

Route Shepherd
PI: Loo (Penn)

PI Loo has been developing the Route Shepherd toolkit, where synthesis techniques are used to automatically configure routing protocols. The motivation of the work lies in the fact that interdomain routing protocol stability depends on the absence of policy conflicts between autonomous systems, but since most policy is kept private, it is hard to ensure that conflicts are avoided. Route Shepherd begins from a partially specific policy configurations, and then apply various synthesis techniques to complete the specifications in a safe manner. In particular, over the past few months, we have focused on using Route Shepherd to solve the *routing recovery* problem. Given a network that is currently oscillating (i.e. unsafe), we devise a synthesis method to determine the shortest possible sequence of configuration changes (link weight updates) in order to make a given oscillation go away, while preserving connectivity. We focus on link weight changes because these are frequently used for traffic engineering, including by automated processes, so there is precedent for deciding what the link weights should be in order to meet a network design objective. The fewer actions need to be taken, the faster the recovery process will be.

To tackle the above problem, Route Shepherd adopts ExCAPE's synthesis methodology of completing partial programs using formal tools. Here, the initial unsafe network is formulated as a *Partial Stable Paths Problem (PSPP)*, a generalization of the well-known Stable Paths Problem (SPP) formalism; this is adapted for coping with missing (partial) information. Route Shepherd will then fill in configuration parameters that are missing or broken, i.e. synthesis a set of reconfiguration steps, in order for the configuration to reach a safe state. In order for synthesis to be carried out efficiently, we adopt the use of Max-SMT techniques. Our evaluation on actual Internet topologies and evaluation on the Emulab testbed (http://www.emulab.net/) shows that Route Shepherd can result in fast recovery times that outperform random strategies. We have developed a prototype of Route Shepherd demonstrated at SIGCOMM'12. As ongoing work, we are continuing our work with traffic engineering in mind, where the resulting policies should not only be safe, but also meet traffic engineering goals of the network operator.

Synthesis for Wireless Control Networks
PI: Pappas (Penn)

We have focused on the synthesis of structured discrete-time linear controllers that mimic the behavior of non-structured controllers. Our work has been motivated by the Wireless Control Network (WCN) architecture, which employs a fully distributed control scheme that causes the entire network itself to act as a structured linear controller. Our goal has been to derive a procedure that maps existing controllers into this structured computational substrate, which will enable direct use of the well-known controllers (e.g., the algorithms for controller tuning, such as PID tuning), along with the practical integration of the WCN with existing centralized controllers and monitoring networks. Specifically, we have derived algorithms for approximation of linear systems with potentially higher order systems that have some structural constraints. We have considered the general case of structural constraints, which usually means that it is not possible to exactly match the initial and derived controllers. Consequently, we employ techniques used for model reduction to specify an error system, which allows us to formulate the problem as synthesis of an optimal structured linear controller. The difference is that our structured controller (e.g., the WCN) has some freedom in the system dimension to compensate for the structural constraints i.e., to reduce the approximation error we can increase the sizes of states maintained by some nodes in the network. The derived synthesis procedure provides a structured controller implementation that minimizes the H_{inf} norm of the error system.

4.4 Programming for Mobile Platforms PIs: Foster (Maryland) and Solar-Lezama (MIT)

This is a new research theme, not included in the original proposal. We have decided to pursue this theme as it offers an opportunity to leverage ExCAPE synthesis solutions to improve productivity of average programmers as they are required to quickly adapt to new platforms supported by mobile devices. In order to successfully transition analysis and synthesis tools to applications on mobile devices, one needs formal models of emerging platforms such as Android. PI Foster, in collaboration with PI Solar-Lezama, is exploring how ExCAPE synthesis tools can be used to automatically extract such models as explained below.

First we developed a symbolic execution engine that runs on Dalvik bytecode, which is what runs on Android phones. A major problem we encountered is that the symbolic executor can't get very far in its execution without calling into or being called by the Android platform. To keep symbolic execution tractable, we need to create an executable model of the platform—it is intractable to directly execute the platform code for a variety of reasons. While we could create such a model by hand, this is undesirable, because it is extremely time consuming and because changes to the Android platform (which happen very often) can rapidly cause such a model to decay if it fails to keep up with changes. Instead, we are developing a way to use program synthesis to create an abstract but executable model of Android that captures the most important properties of the system, but ignores the details so that we can scale our analysis. There are two inputs to our synthesis algorithm: (1) logs of method calls and returns at the API boundary between the platform and the app; and (2) templates that capture high-level design knowledge about Android. To gather logs, we use Redexer, a tool for Dalvik bytecode transformation and instrumentation; we developed Redexer as part of a separate project. For templates, we are currently developing a new high-level specification language. To perform synthesis, we will "compile" (1) and (2) into an input to the Sketch program synthesis tool, and use that as the underlying computational engine. To date we have had success using Sketch (though without the compilation step) to synthesize a model of the Android app lifecycle, the ServiceBinding facility in Android, and the LocationManager class. In the coming year, we expect to develop the specification language and expand this approach to synthesize a much more complete model of the Android platform.

5 Tools and Evaluation

The goal of this research theme is to build open-source tools and design environments, and evaluate them for both computational performance and usability. Main work in this theme so far includes new features for the program synthesis tool Sketch and for the reactive synthesis tool LTLMoP, a new effort Rosette aimed at simplifying the task of building synthesis tools, and planning of user studies. The effort on developing a common framework of *Template-based Synthesis Modulo Theories*, discussed in Section 3, will allow us to connect different tools over the upcoming year.

Sketch
PI: Solar-Lezama (MIT)

The synthesis tool Sketch Version 1.6, developed by PI Solar-Lezamas group, was recently released. It includes a benchmark suite with a range of synthesis problems that will allow for side-to-side comparisons of synthesis procedures developed by other team members.

The Sketch synthesis system is now being used by several of the ExCape efforts. PI Foster (Maryland) and his students have been using it to synthesize harnesses for the Android platform as discussed in Section 4.4. PIs Bodik (UC Berkeley) and Seshia (UC Berkeley) have been using it to synthesize reactive controllers. In order to support the use of Sketch by other team members, we have been producing documentation for the language and are in the process of documenting the API so it can be called from within other tools.

Rosette
PI: Bodik (UC Berkeley)

The Rosette language for prototyping domain-specific synthesis tools is itself a small EDSL that inherits and exposes extensive support for meta programming from its host language, Racket. As such, it offers a versatile and lightweight mechanism for developing domain-specific languages and tools for synthesis, verification, angelic execution, and fault localization. Rosette's reasoning engine is based on the Kodkod constraint solver. At its core, the engine is a partial evaluator; it reduces away all operations on purely concrete values, yielding a symbolic representation of the rest of the program. Any assertions over these symbolic values are translated to the input language of the Kodkod solver, which, in turn, reduces them to a boolean satisfiability problem, solved with an off-the-shelf SAT solver. Kodkod translates the resulting boolean solution to a relational one, and Rosette translates the relational representation back into program values or expressions.

We have used Rosette to synthesize a range of low-level loop-free, conditional-free functions that are suitable for use as primitives in GPU kernels. We have also built a simple OpenCL emulator in Rosette, using its angelic execution facilities to explore different data layouts for tree traversal kernels. Rosette is currently being used in several other projects at Berkeley to synthesize web scraping scripts and attribute grammar specifications.

LTLMoP
PI: Kress-Gazit (Cornell)

The Linear Temporal Logic MissiOn Planner (LTLMoP) is an open source, Python-based toolbox developed at Cornell that allows users to control simulated and physical robots by writing high-level specifications in structured English and LTL. The toolbox works with the Robot Operating System (ROS) and several robotic platforms such as the iRobot Create and the Aldabaran humanoid Nao. Recently we have added support for the Rice Open Motion Planning Library (OMPL - from PI Kavraki's group) and we have integrated it with a natural language processing module. This tool will be used in the robotics challenge problem (Section 4.2).

User Studies for Improving Programming Notations and Tool Interfaces
PI: Hartmann (UC Berkeley)

We have developed a user study methodology to compare whether certain types of notations are easier to interpret and match to program behavior for programmers. We developed this methodology as part of our Proton project on a declarative framework for generating multitouch gesture recognizers. In this particular case, our user study showed that users were roughly four times faster at interpreting gestures written using our high-level, declarative notation than those written in procedural event-handling code commonly used in application programming today. The results were published in October at UIST 2012, a premier venue in

Human-Computer Interaction. The study protocol shows developers multiple examples of program behavior (e.g., a 3x3 table of short video clips visualizing program output) and a single code fragment. Participants then have to match the code fragment to one of the shown output videos. Repeating this test for multiple different types of notations allows for statistical comparison of performance, with notation type as the independent variable and time on task and error rate as the dependent variables.

We hope that we can apply this same methodology to evaluate the synthesis notations and tools being developed by other ExCAPE team members. Discussions are already ongoing to design user studies, with guidance from PI Hartmann, in the following threads: (1) for evaluating the relative merits of using structured English for high-level programming in the robotics challenge problem, (2) for understanding the students' responses to hints in automatic tutoring of programming assignments, and (3) for evaluating the effectiveness of feedback for automata theory problems for online tutoring.

6 Education and Outreach

Activities aimed at education, outreach, knowledge transfer, and industry collaborations are discussed in this section.

ExCAPE Summer School 2013
Organizers: Bodik (UC Berkeley), Lafortune (Michigan), and Zdancewic (Penn)

ExCAPE summer school will be held at UC Berkeley campus from June 12 to 15, 2013. The goal of the school is to expose graduate students and junior researchers to new ideas in program synthesis. Each of three tutorial areas will be covered in three hours of lectures, plus additional hands on sessions on tools and problem solving. Tutorials will be given by PIs Ras Bodik on *Synthesizing programs with constraint solvers*, Paulo Tabuada on *Synthesis for cyber-physical systems*, and Moshe Vardi on *Reactive Synthesis*. The tutorials will be complemented by several invited lectures on theory and applications of synthesis. The current list of speakers includes Alur (Penn), Gulwani (Microsoft Research), Lafortune (Michigan), Murray (Caltech), Seshia (UC Berkeley), Solar-Lezama (MIT), Tripakis (UC Berkeley), and Weiss (Ben Gurion U.). Participation in the summer school is free, and partial support for travel and accommodation will be provided to graduate students in need. The call for participation has been distributed to a number of mailing lists, and students have already started registering (the response has been overwhelming, and by the end of March, over 100 students had registered). See `http://excape.cis.upenn.edu/summer-school.html` for more information.

Synthesis for Automatic Tutoring
PIs: Alur (Penn), Hartmann (UC Berkeley), Seshia (UC Berkeley), and Solar-Lezama (MIT)

We have realized recently that the synthesis technology can be used effectively to develop tools for automatic grading, feedback, and tutoring to improve the quality of online education. Three efforts have started to realize this promise.

PI Solar-Lezama has been leading the development of auto-grading technology for automatically grading and providing intelligent feedback on programming assignments. The tool can now evaluate assignments in Python, which required new techniques to cope with dynamic typing and list comprehensions. They are currently testing it on students in the course 6.00 at MIT and plan it to run at scale on 6.00x MOOC. A paper titled *Automated Feedback Generation for Introductory Programming Assignments*, by Singh, Gulwani, and Solar-Lezama, will appear in the conference PLDI 2013.

PI Seshia and his PhD student, Dorsa Sadigh, have obtained promising results on the problem of *automatic exercise generation*. This is a term coined to describe the three problems of generating new problems, generating new solutions, and auto-grading. Working in the context of the undergraduate Embedded Systems course at Berkeley, they have used a template-based approach to classifying problems in a recent textbook by Lee and Seshia. The approaches to problem and solution generation are based on a combination of mutation and satisfiability solving.

PIs Alur (Penn) and Hartmann (UC Berkeley), in collaboration with Gulwani (Microsoft Research) and Viswanathan (UIUC), has started developing automated tutoring systems for automata theory course using synthesis technology. One challenge in making online education more effective is to develop automatic grading software that can provide meaningful feedback. In the first phase of this work, we have a developed a tool that provides a solution to automatic grading of the standard computation-theory problem that asks a student to construct a deterministic finite automaton (DFA) from the given description of its language. Each student's answer is compared to the correct DFA using a hybrid of three techniques devised to capture different classes of errors. First, in an attempt to catch syntactic mistakes, we compute edit distance between the two DFA descriptions. Second, we consider the entropy of the symmetric difference of the languages of the two DFAs, and compute a score that estimates the fraction of the number of strings on which the student answer is wrong. Our third technique is aimed at capturing mistakes in reading of the problem description. For this purpose, we consider a description language Mosel, which adds syntactic sugar to the classical Monadic Second Order Logic, and allows defining regular languages in a concise and natural way. We provide algorithms, along with optimizations, for transforming Mosel descriptions into DFAs and vice-versa (it should be noted that these algorithms are inspired by ExCAPE synthesis techniques). These allow us to compute the syntactic edit distance of the incorrect answer from the correct one in terms of their logical representations. We report an experimental study that evaluates hundreds of answers submitted by (real) students by comparing grades/feedback computed by our tool with human graders. Our conclusion is that the tool is able to assign partial grades in a meaningful way, and should be preferred over human graders for both scale and consistency. This work is reported in a paper titled *Automatic grading of DFA constructions* to be presented at the premier conference on artificial intelligence, IJCAI 2013. In the next phase, we are exploring how to translate the metrics computed by our tool to provide meaningful feedback to students.

Access to Online Education
PI: Parthasarathy (UIUC)

PI Parthasarathy has been actively involved in building a MOOC (massive open online course) model targeting traditional students in India, in collaboration with Microsoft Research, India. While several open online course platforms have emerged (e.g., Coursera), they have so far not been able to attract traditional undergraduate students. The online course model for India called MEC (massively empowered classrooms) being proposed is significantly different, exploiting the university-college hierarchy in India, where hundreds of colleges fall under one university, and hence have a common syllabus, time-line of teaching, and exam. The MEC model exploits this by aligning the course to the syllabus the student is learning, and moreover works with the teachers in the various colleges to reach students. A pilot course is set to begin in February for a course on algorithms in the VTU university system in and around Bangalore. The MEC model has the potential for large impact, as it would significantly improve the quality of traditional education in India for thousands of students.

Courses and Workshops

ExCAPE PIs have organized and participated in a number of workshops and courses focused on software synthesis.

- SYNT 2012 (First Workshop on Synthesis) was organized at Berkeley, in conjunction with the annual conference CAV, on July 7 and 8, 2012. PIs Bodik (title: Synthesis for systems biology), Parthasarathy (title: Synthesizing programs using bounded domains and Occams razor), and Tabuada (title: Synthesizing robust systems) gave invited talks in this workshop.

- PIs Kavraki and Kress-Gazit are organizing a workshop on Synthesis for Robotics at the annual Robotics: Science and System (RSS 2013) conference to be held in Germany.

- PI Lafortune co-organized a workshop at the 2012 American Control Conference (with Yin Wang of HP Labs and Spyros Reveliotis of Georgia Tech) on "Controlling Software Execution: An Emerging Application Area for Control Engineering."

- PI Bodik and research associate Torlak offered a new graduate course at UC Berkeley (Fall 2012) titled *Program Synthesis for Everyone*. See `http://www.cs.berkeley.edu/~bodik/cs294fa12` for course material. In this course students built 12 synthesizers for a range of problems from distributed systems to learning web scripts.

- PI Lafortune co-organized an invited session at the 2012 IEEE Conference on Decision and Control on *Modeling, Analysis, and Control of Software Systems* (with Yin Wang of HP Labs). Six papers were presented by researchers from US, China, France, Italy, and Sweden.
- PI Lafortune is organizing a special session on Software Synthesis at the 2013 American Control Conference, June 17-19, 2013. ExCAPE will be represented by PIs Lafortune, Pappas, and Seshia. The session will also include a talk by Richard Murray (Caltech) on synthesis of robotic controllers.
- PI Solar-Lezama is co-chair for the meeting SYNT 2013 (Second Workshop on Synthesis) to be held in Saint Petersburg, Russia on July 13 and 14 (co-located with the annual conference on verification CAV).

Invited Talks and Tutorials

ExCAPE investigators gave a number of invited talks and tutorials as listed below:

- *Computer Augmented Program Engineering*, Alur; the Strachey Lecture at University of Oxford (April 2012), Distinguished Departmental Colloquia at UC San Diego (May 2012), and University of Texas at Austin (November 2012).
- *Program Synthesis with Constraint Solvers*, Bodik and Torlak, Invited Tutorial, CAV 2012.
- *Compositional Temporal Synthesis*, Vardi, Keynote Talk, 9th International Conference on Quantitative Evaluation of Systems and 10th Int'l Conference on Formal Modeling and Analysis of Timed Systems, London, UK, September 2012.
- *Logic and Verification*, Vardi, 10th International Winter School on MOdelling and VErifying parallel Processes December 2012, Marseille, France.
- *Motion Planning for Hybrid Systems*, Kavraki, Foundation for Research and Technology, Hellas (FORTH), Heraklion, Crete, Greece, June 2012.
- Streaming Transducers, Alur, Plenary Lecture, IFIP Theoretical Computer Science Conference, Amsterdam, The Netherlands, September 2012.
- Lafortune will present plenary lecture at the 34th International Conference on Application and Theory of Petri Nets and Concurrency, June 24-28, 2013, Milano, Italy.
- PI Pappas gave invited talks at the 22nd International Conf. on Automated Planning and Scheduling (ICAPS, June 2012, Sao Paolo, Brazil) and at the 4th IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS, June 2012, Eindhoven, Netherlands) on his work on synthesis of integrated control and scheduling for wireless control networks.
- PI Sangiovanni-Vincentelli presented a keynote lecture titled *Cybersecurity for automobiles* at the 16th I&C Research Days at EPFL, Lausanne, Switzerland in June 2012.
- PI Sangiovanni-Vincentelli delivered the luncheon keynote lecture at ICCAD (International Conference on Computer-Aided Design) 2012 , titled *ICCAD At Thirty Years: Where We Have Been, Where We Are Going* in November 2012.
- PI Sangiovanni-Vincentelli gave a Lectio Magistralis at the ceremony for the award of an Honorary Doctorate at KTH titled *50 Years of EDA: From Transistors to Smart Planets*.
- *Synthesizing Programs over Bounded Data Domains*, P. Madhusudan; Workshop on Verification of Infinite-State Systems, colocated with FSTTCS 2012, 18th December, 2012, Hyderabad, India.

Outreach to Industry and Government Agencies

ExCAPE PIs are collaborating with industrial researchers, and have started discussions with other Government organizations to find new avenues for applying the synthesis technology. Representative efforts are listed below:

- PI Bodik has started a collaboration with Intel researchers (M. Talupur) on using the ROSETTE system for synthesis of cache coherence protocols.
- PIs Bodik and Solar-Lezama are collaborating with DOE on adding synthesis to DOE compilers for Exascale machines.
- PI Vardi gave the presentation *Specifying System-Level Protocols* at Intel Strategic CAD Lab in Oregon in July 2012.
- PI Lafortune is collaborating with researchers in HP Labs to develop synthesis techniques for avoiding concurrency bugs.
- PI Sangiovanni-Vincentelli is a member of the Technology Advisory Council of United Technologies and has been active in defining the design methodology to be followed for complex avionics systems using platform-based design and software synthesis and formal analysis from models.
- PIs Alur, Hartmann, and Solar-Lezama are collaborating with Microsoft Researchers (S. Gulwani) on using synthesis technology for intelligent tutoring systems.
- PIs Alur, Pappas, and Tabuada are involved in DARPA's HACMS program for synthesis of platform-aware attack-resilient control systems.
- PIs Alur, Pappas, and Tabuada gave presentations at the annual PI meeting of the NSF CPS program to highlight challenges and opportunities for synthesis research for cyber-physical systems.
- PIs Alur and Bodik participated in the NSF Workshop on *Future Directions in Formal Methods* in San Diego in December 2012.

7 Management

The ExCAPE Executive Committee consists of Alur (the lead PI), Bodik, Lafortune, Sangiovanni-Vincetelli, and Vardi. This committee has been responsible for ensuring timely progress on research goals.

Liz Ng was recruited to be the Manager for ExCAPE, and has been providing wonderful administrative support for our activities.

We offered the position of ExCAPE Research Coordinator to Barbara Jobstmann, which she unfortunately declined. The position is now offered to Domagoj Babic. In case he declines our offer, the current recruiting season offers a convenient time to recruit a suitable person.

ExCAPE proposed a novel plan for post-doctoral researchers: each recruit must have at least two mentors from two different institutions, and should foster integrative research activities. Ruediger Ehlers who finished his PhD at University of Saarland in reactive synthesis was recruited as the first ExCAPE post-doctoral researcher under this plan. Ruediger spent the Fall semester at UC Berkeley (mentor: Seshia) and is spending the Spring semester at Cornell (mentor: Kress-Gazit). Ruediger is leading the effort to coordinate and integrate activities in the robotics challenge problem (as discussed in Section 4.2). This *rotating postdoc* experiment has worked out very well leading to unexpected and fruitful collaborations.

We plan to recruit post-doctoral researchers for other challenge problems over the next few months. Advertisements for these positions have been posted on several mailing lists, and a number of candidates have already applied. We are specifically looking for researchers who can contribute to the themes of (1) synthesis for networked systems, (2) synthesis for programming of mobile platforms, (3) integrated synthesis engine and the synthesis competition, and (4) programmer productivity studies and human-computer interaction aspects.

To ensure that the research directions and the challenge problems lead us towards progress that is meaningful for problems in system design faced by industry, we had set up an industrial advisory board consisting of highly distinguished scientists that represent the range of industries who can potentially benefit from ExCAPE research. The kick-off meeting in June 2012 was attended by John Field (Google), Limor Fix (Intel), Patrice Godefroid (Microsoft), Aarti Gupta (NEC), Himanshu Khurana (Honeywell), Andreas Kuehlmann (Coverity), Pieter Mosterman (Mathworks), Mark Wegman (IBM), and Pamela Zave (AT&T), and we were fortunate to receive valuable feedback from them.

8 Collaboration

We have used frequent teleconferences as a means of fostering collaboration. Teleconferences are organized by groups of PIs interested in a common thread. For example, PIs Alur, Bodik, Martin, Seshia, and Solar-Lezama, together with their students, have started a regular telecon to formalize the computational problem common to many ExCAPE projects (see the discussion on *Template-based Synthesis Modulo Theories* in Section 3); and PIs Lafortune, Tripakis, and Vardi have started a new collaboration to find synergies between reactive synthesis and supervisory control.

The events that bring all the PIs together include the Annual PI meetings and the monthly ExCAPE webinars as discussed below.

Annual PI Meetings

The ExCAPE project got off to a great start with the Kick-off meeting held at Penn on June 4 and 5, 2012. The meeting was attended by 17 (out of 18) PIs, 31 students, and 10 researchers from industrial organizations. The lively sessions were organized along the themes of the proposal as summarized below (see `https://excape.cis.upenn.edu/Kickoffmtgdetail.html` for details and presentations):

- June 4, 9-10: Overview and Introduction of the participants
- June 4, 10.30-12.15: Synthesis Approaches and Computational Engines
- June 4, 1.15-3: Challenge Problems: Four parallel sessions on (a) Multicore protocols, (b) Robotics, (c) Networked Systems, and (d) Concurrent Programming
- June 4, 3.30-4.30: Tools and Evaluation
- June 4, 4.30-5.30: Education and Outreach
- June 4, 6-7: Student poster session
- June 5, 8.30-10.30: Synthesis Methodology
- June 5, 11-12.30: Industry Panel and Feedback

The next ExCAPE PI meeting is scheduled for June 10 and 11, 2013, to be held on UC Berkeley campus.

ExCAPE Webinar

ExCAPE organizes a monthly seminar series over the web. For each lecture, about 50 participants login from all over the country. The seminar series provides an excellent opportunity for ExCAPE investigators and students to learn about different aspects of synthesis on a regular basis. The seminars so far have been:

- *Sketch tutorial*, Armando Solar-Lezama (MIT), September 10, 2012.
- *Compositional temporal synthesis*, Moshe Vardi (Rice), October 1, 2012.
- *Integrating induction, deduction, and structure for synthesis*, Sanjit Seshia (UC Berkeley), November 5, 2012.

- *Synthesizing robust systems*, Paulo Tabuada (UCLA), December 3, 2012.
- *Synthesis for education*, Sumit Gulwani (Microsoft Research), January 14, 2013.
- *Synthesis and robotics*, Hadas Kress-Gazit (Cornell), Feb 4, 2013.
- *Program synthesis using smoothed numerical search*, Swarat Chaudhuri (Rice), March 4, 2013.
- *Synthesis for concurrency*, Martin Vechev (ETH Zurich), April 1, 2013.