Volume 1: Technical and Management Proposal

BAA Number: DARPA-HR001117S0045
Technical Area: TA1, TA2, and TA3
Proposal Title: Assured Autonomy for Learning Enabled Vehicles (Safeguard)
Lead Institution: University of Southern California
Type of organization: "OTHER EDUCATIONAL"

Technical Point of Contact
Dr. Craig A. Knoblock
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
Tel: 310-448-8786
E-mail: knoblock@isi.edu

Administrative Point of Contact
Sapphire Masterson
USC Dept. of Contracts & Grants
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
Tel: (310) 448-9161
E-mail: sapphirm@usc.edu

Award instrument requested: Procurement Contract, Cost-Reimbursement, No Fee

Total amount of the proposed effort: $ ...
Phase I: $ ...
Phase II: $ ...
Phase III: $ ...

Place(s) of performance: USC, Marina del Rey, CA; Los Angeles, CA; Tempe, AZ; Palo Alto, CA
Period(s) of performance: 04/02/2018 - 03/31/2022

Other team members:

Kestrel Technology
(small business)
POC: Matthew Barry
3260 Hillview Avenue
Palo Alto, CA 94304
phone: (832)205-4876
mrbarry@kestreltechnology.com

Arizona State University
(Other Educational)
POC: Chitta Baral
Department of Computer Science and Engr.
Ira A. Fulton School of Engineering
Arizona State University
Brickyard Suite 572, 699 S. Mill Avenue
Tempe, AZ 85281-8809, U.S.A.
email: chitta@asu.edu

Proposal validity period: 180 days

Data Universal Numbering System (DUNS) number: 072933393
Taxpayer identification number: 95-1642394
Commercial and Government Entity (CAGE) code: 1B729 Marina del Rey, CA
Proposer's reference number (if any): 4409-0

1 Table of Contents

Contents

2 Executive Summary

As we rapidly move into a world where machine learning plays a central role in realizing autonomous systems, it is becoming increasingly important to develop techniques that assure that these systems will operate safely and perform as expected. Current approaches are limited to providing assurance for systems with limited or no learning capabilities. In this context, DARPA's Assured Autonomy BAA seeks to *develop rigorous design and analysis technologies for continual assurance of learning-enabled autonomous systems*. USC in collaboration with Kestrel Technology and ASU is pleased to submit a comprehensive TA1, TA2, and TA3 proposal entitled *"Assured Autonomy for Learning Enabled Vehicles (Safeguard)."* We plan to provide an end-to-end solution to support autonomous systems with learning-enabled components, ranging from design technologies for assurance, to assurance monitoring and control techniques, to representation and online evaluation of assurance cases. We have assembled a strong team of experts that cover the range of technologies that are required to create such an end-to-end system. If successful, the project will provide the technologies for building the next-generation of learning-enabled autonomous systems. The entire project will take four years and cost $??, with an initial version completed at the end of Phase I and successive versions with additional capabilities and improved scalability at the end of Phase II and Phase III.

In the remainder of this section, we first introduce an unmanned surface vehicle scenario that will be used throughout the proposal to describe the approach. Next, we describe our approach to design, monitoring, and dynamic assurance. Finally, we introduce the team involved in the project.

Motivating Scenario. Consider an autonomous unmanned surface vehicle (USV) guarding a valuable asset in the ocean when an unknown vehicle approaches the security perimeter, under challenging weather conditions. In this scenario, the USV is required to approach the intruding vehicle, issue a warning signal, and escort it to a safe distance from the controlled area. However, as the USV has no a priori knowledge of its external environment behaviors (e.g., water depth, waves, wind, current, visibility), pre-computing a feasible trajectory, let alone optimal, becomes a non-trivial problem. For trajectory planning, the USV must continuously perform the following tasks:

- Sense the current state of the surrounding environment (e.g., water depth, waves, wind, current, visibility) and estimate its own maneuverability constraints (e.g., braking distance, available acceleration, maximum velocity, turning radius, turning rate, safety distance) based on the state of the environment;
- Sense the static obstacles in the sensor range and generate a traversability map;
- Sense the moving obstacles and classify them;
- Predict future trajectories of moving obstacles;
- Determine if any of the COLREGS [30] rules will be in effect with respect to one or more of the nearby vessels and identify the vessels with the right of way.

The above information will be used by the trajectory planner to compute an initial trajectory, which will be continuously refined as the USV gathers additional information. The USV will use learning enabled components to take decisions as it encounters new situations, such as

- Classifiers to identify moving obstacles based on physical appearance and motion signatures,

- Algorithms to estimate the sensor capabilities in adverse weather conditions,
- Algorithms to accurately estimate uncertainty in the environment,
- Classifiers to generate traversability maps,
- Prediction of external vessel behaviors based on motion histories,
- Reinforcement learning to ensure COLREGS compliance of maneuvers,
- Algorithms to learning pursuit behaviors.

Learning enabled components will interact with each other in complex ways, where a misclassification error in one component may eventually compromise the entire mission. While exhaustive validation of learning-enabled cyber-physical systems (LE-CPSs) is a prohibitive task [59], their complexity, heterogeneity, and highly dynamic nature make it challenging to even leverage existing model-based development techniques to effectively assess system correctness at design time or enforce it at runtime.

Design for Assurance. Safeguard uses a platform-based design approach [86] to organize the design process for a LE-CPS and to build assurance cases. Composite models are developed at several levels of abstraction, from top-level system requirements and safety constraints down to the implementation level. Intermediate levels add detail to the levels above. The different levels are connected by refinement mappings that allow properties established at one level to be preserved at the next level (see Figures 2 and 3).

Contracts are used to formally specify components and composite models in terms of (1) Assumptions – the assumed behaviors of the environment and the behaviors of other components, and (2) Guarantees – the behavior properties that a model guarantees if it operates in a context that satisfies its assumptions. A calculus of contracts allows horizontal composition of contracts to generate contracts for composite models. Vertical contracts are used to specify the mapping or refinement relation between models at different levels of abstraction. The system design process starts with a high-level contract that expresses overall system assumptions and requirements. Subsequent levels express models with increasing detail until the lowest level expresses the system in terms of hardware components and their software controllers.

The assurance case for a CPS arises from the horizontal and vertical structure of the design in several ways. The components used within a particular level are either (1) synthesized using correct-by-construction design tools together with proofs, (2) derived statically or dynamically using safety-aware machine-learning techniques, (3) written manually and verified by analysis tools, or (4) written manually and validated by extensive testing. The assurance case for the whole reflects its compositional structure. We anticipate that well-specified contracts together with the calculus of contracts will eliminate well-known problems with unexpected emergent behaviors in CPS systems.

The assurance case for the lowest-layer design arises from both the intra-level assurance and from properties and their proofs that are preserved under the refinement mapping from the top-level requirements. The refinement mappings between model layers will be constructed using a variety of techniques. A contract at an abstract level can be mapped to a component or refined contract by (1) retrieval of pre-verified components from a platform library, (2) synthesis using correct-by-construction design and optimization tools, or (3) manual coding to satisfy a contract. The mapping of a composite model will be composed from the mappings of its constituent components

or contracts. When a composite model cannot be mapped compositionally to the next level, it will be generated using correct-by-construction design and optimization tools.

Assurance Monitoring and Control. We provide an integrated framework for safety-aware learning, assurance monitoring and control, detecting distribution shifts. Three major components offer an efficient TA2 architecture as well as interfaces with TA1 and TA3, that is, (a) safety-aware learning and planning, (b) assurance monitors for guarding architectural and safety constraints; and (c) distribution shift detection.

We will develop a new learning-enabled online decision-making framework that allows opportunistically composing a sequence of actions (maneuvers) to reduce uncertainty in the system capability model without suspending the progress toward the mission goals or compromising safety. Each candidate action is evaluated based on three criteria: (1) the risk of violating a safety constraint using the current uncertainties in the parameter estimates; (2) its relevance to the mission goals; (3) its expected information gain, i.e., reduction in uncertainty, with respect to the parameter estimates. These evaluations are combined to produce a cumulative mission utility value for each action that drives our learning-enabled decision-making framework. The problem of generating and evaluating sequences of actions can be posed in several way. For example, it can be solved using a branch-and-bound search method like Anytime A*, or formulated with the finite-horizon Markov Decision Process (MDP) framework. We will develop new scalable search strategies to solve this problem efficiently, by potentially evaluating a recent method developed at USC, called FastMap, that can significantly improve the execution time.

We will develop monitors for architectural and safety constraints. To achieve scalability and decrease the overhead, we propose the application of a technique that we currently use in DARPA's RSPACE program, which leverages a physical model of the vehicles dynamics and its interactions with the environment to efficiently determine the readout frequency. We propose two extensions of this basic idea. First, we will use the theory of Variable Elimination to prioritize which variables to monitor, e.g., controllable, versus uncontrollable, adversarially controlled, or unobservable variables. Second, we invoke the dynamic assessment of assurance cases only when needed. This decreases the number of times dynamic assessment of assurance cases is initiated as well as the communication bandwidth between the TA2 and TA3 components.

Finally, we will identify a distribution shift by combining statistical and machine learning techniques to differentiate between environmental and sensor changes. We will exploit a categorization of the shifts based on their cause and duration as well as extend our earlier work on detecting and mitigating sensor failures for all types of monitored variables.

Dynamic Assurance: The Safeguard *design for assurance* activity takes a systems-theoretic stance toward safety. Consequently, it presumes that safety is an emergent property of the system, and that hazards can present themselves through unintended interactions and performance violations in addition to causal events such as component failures. Our design approach includes consideration of intent as well as hazard analysis and mitigation. The artifacts from these activities populate contracts and assumptions for the dynamic assurance case. We thus build safety into the product by working at a systems-level viewpoint, using lexicon and design patterns familiar to both hardware and software engineers; safety is an emergent property of the system, not an afterthought. As system behavior evolves during runtime owing to learning, threats, degradation, or some other factor, the dynamic assurance case identifies whether the safety constraints continue to be satisfied. If not, it provides notifications or issues recovery instructions directly from a lookup table.

Our implementation of the dynamic assurance case employs a declarative knowledge base

inference engine and a domain-specific language tailored to our approach. We have used them successfully for assurance case tool sets and arguments, and will extend them to reason about uncertainty and learning. Our approach to achieve scalability is to specialize solvers toward modularity and to take advantage of domain knowledge. Specifically, we will develop answer set programming techniques for context-dependent learning for reasoning about the learning-enabled components as well as learning assurance rules. We will develop new formalisms for uncertainty to include causality, using weights for computing probabilities, and probabilistic non-monotonicity. To achieve scaling objectives we will implement specializations using modularity, weighted CSPs, and message passing.

Team. We have assembled a team that is exceptionally well-qualified to build the proposed Safeguard system. The team will be led by Dr. Craig Knoblock, the Principal Investigator for the effort, who currently leads the Intelligent Systems Division at the Information Sciences Institute. He has led many large DARPA and IARPA projects over the years and has a strong track record in conducting leading edge research and then transitioning the technology to commercial use. He will be supported by Dr. Michael Orosz as the Project Manager, who also has experience in managing large research projects and on autonomous systems. The TA1 team will be led by Dr. Pierluigi Nuzzo, who is an expert in embedded system design methodologies and the application of formal methods to cyber-physical systems. The TA1 team also includes Dr. Doug Smith, who has spent many years working on scalable correct-by-construction techniques and Dr. Henny Sipma, who has significant experience in applying program verification methods to real-world problems. The TA1 team also includes Ryan Goodfellow, who has done a large amount of work on simulation-based testing. The TA2 team will be led by Dr. Knoblock who has worked on topics related to both monitoring and detecting distribution changes. He will be supported by Dr. Satyandra Gupta, who is an expert on autonomous surface vehicles as well as on safety-aware learning. He will also be supported by Drs. Anoop Kumar and Satish Thittamaranahalli, who have also previously worked on efficient methods for execution monitoring. The TA3 team will be lead by Dr. Matthew Barry, who has experience in creating the technologies for assurance cases. He will be supported by Dr. Chitta Baral, who is an expert on ASP solvers and by Dr. Thittamaranahalli who is an expert on SAT solvers, both of which will be applied to provide scalable assurance case reasoning. Finally, Dr. Petros Ioannou, who is an expert on control systems for autonomous vehicles will provide an autonomous vehicle platform, which will form the focus of our work until the TA4 teams provide additional vehicle platforms for development.

3 Innovative Claims and Deliverables

In this project we will develop and build an end-to-end system for assured autonomy. This section describes the key innovations by technical area and then the overall deliverables of the project.

Design for Assurance

- We address the LE-CPS design challenges via a holistic approach that can contextually generate design artifacts and assurance cases. We develop a compositional, contract-based modeling framework, methods, and tools to support the design process from system-level requirement capture, formalization, and analysis, to the generation, testing, and continual monitoring of software and hardware artifacts in feedback loop with a physical process.
- We develop compositional abstractions and interfaces (vertical contracts) that can bridge heterogeneous formalisms and heterogeneous decomposition architectures to make system analysis and synthesis tractable, consistently combine different verification and synthesis methods at design time, and provide seamless support for dynamic assurance at run time.
- We develop a unifying framework and efficient algorithms to reason about the combination of discrete and continuous dynamics and constraints in the presence of uncertainties in LE-CPS using a satisfiability modulo convex approach [103] for contract-based system verification and scalable trajectory planning.
- We provide an environment for high-fidelity CPS testing, in which production-ready software, e.g., safety-critical learning and control, may be deployed and tested with time dilation facilities, so that it synchronizes with a physical simulation that is not necessarily running in real time, while still having the perception of real time.
- We propose an approach for unanticipated behavior space identification and test coverage maximization which leverages results from the theory of differential algebraic equation (DAE) [18, 55, 17, 71] to prune the behavior search space and identify smaller regions of interest for efficient simulation-based testing.

Assurance Monitoring and Control

- Our safety-aware learning framework enables the system to opportunistically select and execute actions to assist the learning-enabled component in reducing model uncertainty without compromising safety or deviating from the mission goals. The value of uncertainty reduction is explicitly incorporated in the optimization process for selecting the best action.
- For safety-aware learning, we propose the idea of preprocessing the search space of the problem domain before queries and observations come in. With such a linear-time preprocessing phase, the performance of search and optimization algorithms can be significantly boosted. For example, in regular A* search, the intensional or extensional search space can be preprocessed in near-linear time to yield an embedding of each state as a point in Euclidean space [29]. Then, when the query comes in, A* search can make use of these Euclidean distances as heuristic distances between two states to yield order-of-magnitude speedups.

- As massive amounts of sensor information flow in, it is imperative for us to efficiently process this information for monitoring architectural and safety constraints. Building on our past work on similar tasks, we propose novel technologies for efficiently monitoring constraints. These algorithms can yield an exponential reduction in the amount of sensor data that needs to be processed. Doing this also reduces the message complexities between the various modules.
- We will develop techniques to identify distributional shifts and determine the underlying cause (e.g., change in environment, sensor failure, etc.), as well as strategies for handling the various distributional shifts. Notably, we propose to build on our past work and use compact representations to exploit historical data to identify distributional shifts.

Dynamic Assurance

- We demonstrate the integration of dynamic assurance for safety-critical learning-enabled dynamic systems in which evolutionary behaviors are expected and tolerated as a property of the functionality. The impact will be consequential contributions safety-critical dynamic systems in which evolutionary behaviors are expected and tolerated as portion of the functionality.
- We implement dynamic assurance by combining features of system safety, formal methods, logic programming, uncertain reasoning, and domain-specific languages. We populate assurance case arguments at several levels of modeling and implementation abstraction, using the analysis results to produce design-time evidence supporting assurance claims.
- We develop and demonstrate ASP formalisms crucial to applications in dynamic assurance. We demonstrate the suitability of the technology especially for assurance case arguments owing to the improved legibility, consistency and completeness checks, handling of uncertain and default reasoning, and scalability. We provide a novel approach to handling uncertainty that provides the ability to do causal and counter-factual reasoning as well as probabilistic non-monotonicity. Overcoming limitations of traditional inductive logic techniques, we develop a novel iterative and incremental approach based on context dependent learning.

Deliverables During the course of this project, we will build and deliver a fully-operational system that covers all three of the technical areas. The detailed capabilities of this system are described in the individual technical sections. The resulting system will be available as open source under a permissive license, which will allow other organizations to use the work, extend it in new directions, and even commercialize the software. Kestrel Technology has significant experience in this space and has built and applied these types of technologies to a variety of real world tasks. Kestrel is ideally suited to pursue commercial uses of this technology and the permissive license will facilitate exploring these opportunities since there will be no need to negotiate intellectual property rights.

4 Technical Plan

4.1 Design for Assurance

4.1.1 Current Approaches

Consider an autonomous unmanned surface vehicle (USV) guarding a valuable asset in the ocean when an unknown vehicle approaches the security perimeter, under challenging weather conditions. In this scenario, the USV is required to approach the intruding vehicle, issue a warning signal, and escort it to a safe distance from the controlled area. However, as the USV has no a priori knowledge of its external environment behaviors (e.g., water depth, waves, wind, current, visibility), pre-computing a feasible trajectory, let alone optimal, becomes a non-trivial problem. The USV must continuously sense a large set of parameters and update its estimations about the external world to compute a trajectory that is safe and satisfies the maneuverability constraints and the COLREGS rules. It must continuously generate traversability maps by sensing fixed and moving obstacles in the sensor range, by classifying moving obstacles, and by possibly predicting their trajectories.

Over the last decade, machine learning (ML) algorithms have achieved impressive results in solving large-scale perception problems for decision-making under uncertainty in cyber-physical systems (CPSs) (see, e.g., [23, 58, 80]) such as the USV in our scenario. While a successful mission will heavily depend on ML components, its safety-critical nature raises concerns on the overall system safety and trustworthiness. In fact, recent efforts toward the verification of ML programs [110] show how even well-trained neural networks can be sensitive to small perturbations, e.g., small modifications can lead the network to misclassify the altered input with large confidence. In systems with complex interactions, a misclassification error in one component may eventually compromise the entire mission. While exhaustive validation of learning-enabled cyber-physical systems (LE-CPSs) is a prohibitive task [59], their complexity, heterogeneity, and highly dynamic nature make it challenging to even leverage existing model-based development techniques to effectively assess system correctness and compliance with regulations at design time. LE-CPSs require significant levels of adaptiveness and resilience to unpredictable scenarios. For these systems, assumptions made at design time may break at run time, and correctness claims must often be based on evidence that can only be provided during system operation.

In this program, we address the LE-CPS design challenges via a holistic approach that can contextually generate design artifacts and assurance cases. Rather than focusing on ML components in isolation, we seek to systematically specify and analyze the behaviors of ML components within the context of a CPS. We treat safety as an emergent property of the system, wherein unintended interactions and performance constraint violations are sometimes more consequential than failed components. We develop a modeling framework, methods, and tools to support the design process from system-level requirement capture, formalization, and analysis, to the generation, testing, and continual monitoring of software and hardware artifacts in feedback loop with a physical process.

We will pursue a contract-based approach [86, 98] leveraging modularity and compositional reasoning to enhance the *scalability* of design and verification tasks and aiming to produce artifacts that are *correct-by-construction*. A contract can be intuitively seen as a pair (A, G) of properties, where A specifies the assumptions that a component makes on the external environment and G the guarantees it provides under these assumptions. In contract-based design, formal

support for assume-guarantee reasoning is directly built into the modeling framework since the early design stages. Contracts enable modular and hierarchical verification of global properties of a system, whose satisfaction can be proven based on the satisfaction of local properties of the components [28]. Contracts support stepwise refinement, where hierarchical specifications can be used to reason about component decompositions, even if the component implementations are not yet available [28, 87, 54]. Contracts facilitate component reuse, as any components satisfying a contract directly inherit its guarantees [54]. Contract constructs, carrying claims as well as evidence, context and assumptions, thus become critical reasoning elements in our dynamic assurance case.

Overall, contracts have shown to be effective for specifying and reasoning about systems, especially when component models belong to the same level of abstraction (e.g., algorithm, software, architecture) or adopt the same formalism. However, there is no universal modeling formalism that can capture every aspect of complex, heterogeneous LE-CPSs, and guarantee, at the same time, tractable analysis. Designers usually "decompose" a complex system into semantically different models, by adopting the most convenient formalisms to represent different design viewpoints (e.g., system functionality, timing, safety, and reliability) at different abstraction levels. They then leverage the most suitable tools to analyze and synthesize these models separately. For example, the control law for a vehicle is typically derived and initially evaluated using control-oriented models, e.g., discrete-time linear models, in which details of the implementation and the physical dynamics are simplified or neglected. These details are usually modeled and evaluated using other formalisms and tools, e.g., by simulation of continuous-time differential algebraic equations (DAE). Each model represents some aspects of system design and occludes others, by making simplifying assumptions that are often undocumented or informally captured at best. A set of challenges then remain for model-based design flows when models are to be formulated and manipulated along the design flow, across different abstraction levels, and between design time and operation time. ML components further exacerbate these difficulties due to the lack of practical and reliable abstractions that can encompass the heterogeneity of ML algorithms and architectures (e.g., data-driven classifiers for perception and situational awareness, reinforcement learning for developing control policies, and algorithms for learning system dynamics) and the different nature of their performance and safety guarantees.

Our vision for LE-CPS design is that compositional abstractions should bridge heterogeneous formalisms and heterogeneous decomposition architectures, to make system analysis and synthesis tractable, to consistently combine different verification and synthesis methods at design time, and to provide seamless support for dynamic assurance at run time. This requires new abstraction mechanisms to quantitatively capture the confidence in the satisfaction of requirements under uncertain or unknown conditions, and resilience properties of systems at both the abstract (algorithm) and the concrete (physical, hardware, and software) levels, to enable trade-off evaluation between resilience, performance, and cost. This vision informs the research efforts detailed below.

4.1.2 System Architecture and Design Approach

The control architecture for an autonomous vehicle, such as the USV in our scenario, is hierarchical. For a given set of mission goals and tasks, determined by the *supervisory controller*, in charge of the higher level decision-making, a *trajectory planner* generates optimal trajectories that avoid obstacles and satisfy the environment constraints. The *vehicle* motion is then governed by its dynamics and the lower-level feedback *controller* driving the actuators to follow the planned trajectory. A simplified diagram is shown in Figure 1. Classical feedback control can be designed

to accommodate limited modeling and sensor inaccuracies and unpredicted environment conditions. However, when inaccuracies exceed tolerable bounds, or unexpected changes occur in the surrounding environment, the control loop needs to be updated, adapted, and reconfigured in real time as a result of learning. Learning and adaptation mechanisms may be pervasive (in the controller, reference model, and perception components); while each LE component may include its own assurance monitors, e.g., detecting and rejecting potentially erroneous sensor data or misclassifications, the overall feedback loop must be guaranteed stable and robust [57, 56, 60]. As safety and correctness are insured across the entire control hierarchy, the complex interactions between component-level safety and system-level safety may ultimately lead to intricate, multi-layer monitoring architectures with significant testing and validation overhead.

Our approach to address the above challenges is via the formulation and solution of a design space exploration problem, where we aim to rigorously quantify and possibly minimize the overhead associated with dynamic assurance under safety and correctness constraints. We follow the multi-step approach shown in Figure 2, inspired by platform-based design (PBD) [65], which has been a successful paradigm for electronic design automation, to *explore* the heterogeneous design space and to *construct proofs* of correctness. At each abstraction layer, a representation of the design is built out of a library (collection) of components according to composition rules. The bottom-up phase of design flow consists in building the component library. In the top-down phase, the high-level requirements are formalized and a *design refinement* step called *mapping* is performed, where the requirements are mapped into the implementation library components. Mapping is the mechanism that allows moving from a level of abstraction to a lower one using the available components within the library. After each mapping step, the current representation of the design serves as a specification for the next mapping step, until the physical implementation is reached. We will use contracts to provide formal guarantees about the correctness of each refinement step [86].

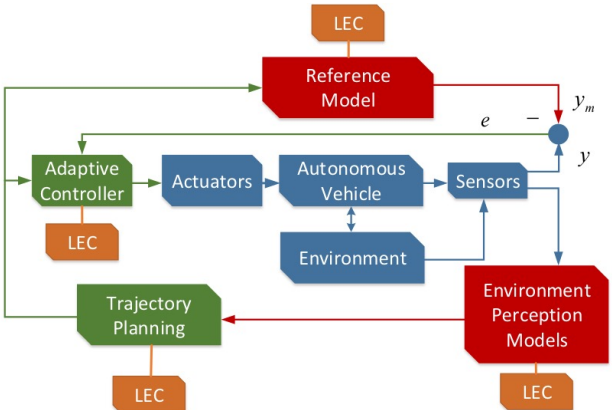

Figure 1: Simplified control architecture for a LE autonomous system.

Contracts for LE-CPS Specification, Analysis, and Design. All the components and their aggregations will be specified by *contracts*, each expressing a set of properties that a component or system satisfies. The contracts will be used to construct mathematical evidence that a design is correct and consistent with its physical implementation under certain assumptions on the external environment. By relying on contracts, we can provide formal assurance, for instance, on whether a set of contracts is realizable, or facets of these are inherently conflicting, and thus no implementation is feasible (contract compatibility or consistency checking). We will develop algorithms and tools for capturing requirements of LE-CPSs, for translating them into contracts, for analyzing and validating them using contract operations and relations, and for synthesizing design and verification artifacts from contracts. Recent work from the investigators includes an end-to-end framework for requirement capture, formalization, and validation, which combine a practical front-end formal specification language based on specification patterns together with a rigorous synthesis and verification back-end based on assume-guarantee contracts [85]. Our framework features a modular

and extensible software infrastructure with interfaces that can support different mathematical formalisms as well as pre-existing verification languages and tools. We will significantly extend this framework by developing domain-specific languages for the specification of LE components, systems, and their dynamics as well as supporting automatic translation of requirement patterns to mathematical languages. Our initial results show that this approach can substantially facilitate the orchestration of formal, exhaustive analyses of temporal properties of CPSs that would otherwise be lengthy, tedious, or error-prone to even formulate, let alone check.

Vertical Contracts for Hierarchical Design. We will develop a methodology and scalable algorithms, supported by tools, for hierarchical architecture exploration, mapping between heterogeneous layers, and generation of correctness proofs by combination of formal verification, synthesis, and simulation-based testing techniques. The investigators have demonstrated an approach to hierarchical design on an aircraft power system distribution network [87, 86], which combines mathematical optimization tools to reason about *architecture constraints*, *linear temporal logic* (LTL) [89] to reason about safety and real-time requirements on a discrete system abstraction, and signal temporal logic (STL) [75] to monitor complex dynamical properties related to the physical plant and the hardware and software implementations of the control algorithm by simulation, including optimization-based search procedures to approximate unsafe regions of the state space. Such a layered approach is a natural framework to explore trade-offs between fidelity of abstractions and scalability of the verification flow. The selection of abstractions and their manipulation was mostly an *ad hoc*, manual process, demonstrated on scenario with a dozen requirements on a hybrid model with about 60 binary and 10 continuous states without learning components. We contend that orders of magnitude improvements in scalability can be achieved by a systematic, automated, and compositional approach to property-preserving mappings between models.

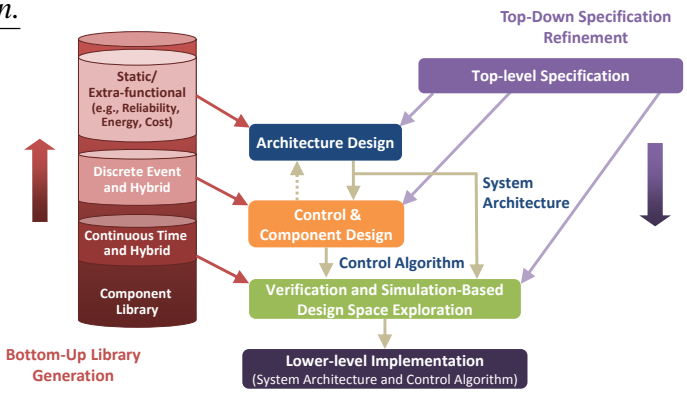

Figure 2: Structure of the proposed methodology.

We will develop a theory of *vertical contracts* as a foundation to the rigorous and systematic integration of complex systems, to fully support multi-layer design with heterogeneous models, and the generation of correctness proofs by combining effective algorithms and tools at different abstraction layers [86]. Intuitively, if vertical contracts are satisfied, a mapping mechanism can be used to produce design refinements that are correct. A formalization of vertical contracts and its systematic implementation to perform model transformations are, however, in their infancy. In model-based verification, abstractions between heterogeneous models were established in the past for specific pairs of formalisms, such as hybrid abstractions of nonlinear systems [52, 34], linear hybrid automata abstractions of linear hybrid systems [39], discrete abstractions of hybrid systems [8, 7, 27]. Our work will generalize these notions toward a formulation that is compositional and systematically applies to virtually any pair of formalisms.

A key criterion on mappings is that they preserve safety properties. In this program, we will use *abstract interpretation* as a means for verifying and enforcing safety properties in a behavioral model. Abstract interpretation works by creating an abstraction of the behavioral model in which

it is more efficient (1) to infer state invariants and behavioral properties, and (2) to prove properties on the given model [31, 88]. In this way, we build systematic mechanisms that can encompass any logical breakdown of complex system verification and synthesis problems into arbitrary conjunctive and disjunctive combinations of smaller sub-problems. The resulting verification and synthesis artifacts support the system safety and correctness claim and are returned to the TA3 investigators for the construction of the assurance case. As shown in Figure 3, lower-level (e.g., physical-level) contracts will mostly be inferred using simulation-based testing to provide conditional evidence for the higher layers. By encapsulating knowledge from feasible implementations of design elements and architectural solutions, a library-based approach spanning different levels of abstraction and granularity will also contribute to reduce the size of the state space and accelerate verification. Previous work by the proponents demonstrated up to two orders of magnitude improvement in execution time using a contract library to verify the functionality of embedded controllers subject to architectural constraints [54]. Similarly, [38] reports about the selection of an optimal aircraft management system design out of a library of over 1.5M configurations in under two hours, by exploiting simulation-based mapping and a monotonicity property of system requirements with respect to the component library.

Contracts for Probabilistic Reasoning. Learning components are usually required to achieve high accuracy, integrate data from several sensors, and run efficiently on embedded processors. In some situations a fast training phase yielding a low-latency, but slightly inaccurate output might be more valuable than a slower but more accurate output. However, such variations in the learning systems must not compromise on key safety and performance objectives. When using machine learning and statistical sensor fusion algorithms to infer information from the external world, providing support for reasoning about probabilistic behaviors and for the development of robust design techniques that avoid over-design becomes crucial. We have recently proposed a contract framework relying on stochastic signal temporal logic (StSTL) to specify and reason about stochastic system behaviors, which is expressive enough to represent hybrid system behaviors and amenable to sound and efficient formulations of verification and control synthesis problems [74]. We will build on this work to develop theories and tools for abstraction and analysis of heterogeneous learning systems that can be used to inform architectural decisions and trade performance with computation time.

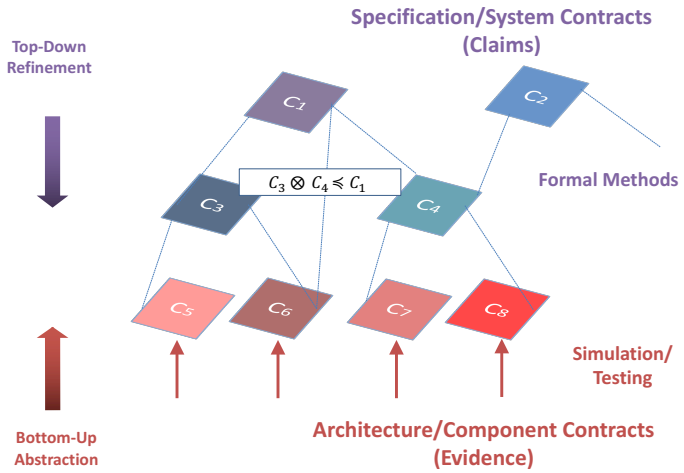

Figure 3: Example of contract-based assurance case.

Scalable Verification and Synthesis via Satisfiability Modulo Convex (SMC) Programming. The robust control of continuous dynamical systems often involves the reduction of the control problem to a convex optimization problem. On the other hand, the design and verification of discrete systems involves the use of Boolean satisfiability (SAT) [76] solving and related technologies such as satisfiability modulo theories (SMT) [13]. Recently, the investigators have devised a

new framework that combines both SAT solving and convex optimization, termed as Satisfiability Modulo Convex optimization (SMC) [103], that holds great potential for developing efficient computational tools to jointly reason about discrete and continuous dynamics. Preliminary results show that SMC can achieve about 4 orders of magnitude better performance than competing tools on satisfiability problems with a few thousands continuous and Boolean variables. We will build on this work to develop an SMC toolkit with significantly greater expressiveness, capable of supporting constraint solving problems that arise in the verification and synthesis of LE components and systems in the context of our contract-based approach. Extensions will include support for optimization as well as robust offline and online trajectory planning in the presence of uncertainties.

Property Enforcement and Formal Verification by Abstract Interpretation. A system S satisfies a specification P if all behaviors of S are included in P. Proving satisfaction requires the computation of the set of reachable states of S, whose termination may be undecidable, or not even guaranteed, in practice. Abstract interpretation [31], a mathematical theory of approximation provides a powerful solution to these problem, by performing the computation in an abstract domain in which the detection of convergence is decidable, and guaranteeing that the results can be translated back to the concrete domain. The theory of abstract interpretation is well understood and many abstract domains of various levels of expressiveness have been proposed in the literature over the past decades [61, 32, 100, 101, 99]. Its practical application, however, has proven difficult. Kestrel Technology is one of only a few organizations that have an abstract interpretation engine, called CodeHawk, that scales to real-world systems. In particular, CodeHawk has been applied to sound memory safety analysis of C programs of up to 400,000 lines of code and the analysis of various other (state) properties of Java programs of more than a million lines of code.

For behavioral requirements on the CPS, our abstract interpretation technology provides a foundation for the verification of contract refinement relationships described above as well as the generation of efficient, localized monitoring code. We propose to develop (1) new front ends for our abstract interpretation engine for the modeling languages created under this project, that allow proving properties over models and refinements between different levels of abstractions; (2) new abstract domains that can express nonlinear behaviors; (3) a new methodology for proving temporal properties over models and programs by creating and discharging proof obligations that ensure system transitions stay within the behavior allowed by an automaton; (4) a new methodology for enforcing temporal safety properties over models and programs by refining contracts on components to eliminate unsafe behaviors; (5) techniques to provide bounds on performance constraints such as maximum CPU time and memory usage; and (6) transformations to generate runtime monitors based on local residual properties that cannot be proved or enforced.

Conditions to monitor can come from critical assumptions on the CPS design, modeled as automata, and well-known algorithms can be used for synthesizing efficient monitors, e.g., [51, 94]. Moreover, in previous work, we explored how to use abstract interpretation to symbolically simulate a safety property, expressed as an automaton, over a behavioral model. This technique can be used to verify that the safety property holds, and, if a local contract on a component is too weak to support the property, to refine the contract to eliminate unsafe behaviors. Residual conditions from the safety property that cannot be discharged by verification or refinement are transformed into runtime monitors and inserted in the code.

Simulation-Based Testing and System Testing. During the first phase of the program, we will consider the scenario of an autonomous vehicle moving in an unknown urban environment to prototype our verification methodology and tools. We will then leverage the state-of-the-art micro-

scopic traffic simulator PTV VISSIM to test the real-time performance and safety of our solutions. VISSIM allows simulating traffic patterns based on the dynamics and driver actions of each vehicle as well as their interaction with each other and the road. An API allows the user to replace the default driver model with a customized one as well as implement and simulate the dynamics of learning components in the loop. Further, it allows generating unpredictable scenarios and collect data to train ML components. As we target industrial-strength platforms, we will also offer a CPS testbed infrastructure that 1) provides a level of fidelity for simulation-based testing that can be relied upon for critical applications and 2) leverages physical model analysis tools to facilitate the identification of problematic behavior spaces.

Monolithic simulation architectures [3, 5, 2, 4], where both the cyber and physical parts of a system are modeled using the same language and executed by a single overarching runtime, provide limited flexibility for industrial-strength designs, since cyber components must be implemented or linked with the language and runtime of the simulation framework. To be able to execute machine learning libraries that could be used in a production setting, in a possibly distributed system[47, 16], we propose, instead, a modular architecture (Figure 4), where a physical simulation can interact with cyber components over a network via virtual sensors and actuators. We will build on our previous work including a system, called Cypress [47], that follows this architecture and significantly extend it to provide support for the simulation of Modelica models, widely used for modeling DAE-based physical systems. Full Modelica support will be achieved by integrating parts of the Cypress engine into the OpenModelica engine. To provide timing guarantees, we will pursue a time dilation approach [47], where all cyber components run inside virtual machines whose clocks are synchronized to the simulation engine through a protocol that is part of the simulation architecture. Static time dilation mechanisms have been shown to be effective in aligning physical simulation time with the execution time of cyber components [48], when a worst case bound on the dilation factor can be computed a priori. We will start with these techniques, and then pursue more recent dynamic time dilation techniques [73].

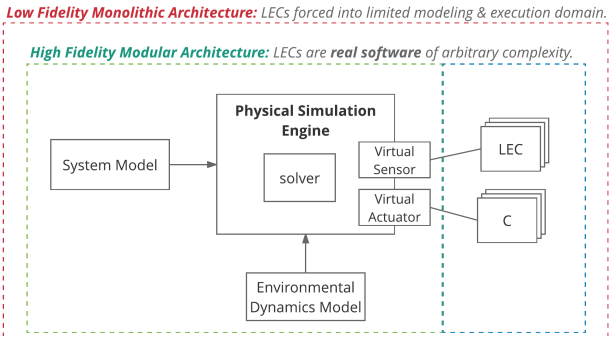

Figure 4: Example of contract-based assurance case.

Many physical systems may be modeled as constrained dynamical systems, represented by DAE, that can be simulated using commercial and open-source tools such as [3, 5, 4, 2, 1]. Drawing from recent results in the DAE-theory literature [18, 55, 17, 71] we will develop a notion of controllability to develop and inform strategies for *unanticipated behavior elicitation and test coverage maximization*. These mechanisms may be viewed as a means to help prune the simulation search space. As the recent work is primarily focused on linear models, we will investigate extensions of these notions to nonlinear models used for LE control systems.

Table 1: Project Plan for TA1

Phase	Plan
Phase I	Develop initial support for the design framework using at least three abstraction levels, each with its own modeling language. Develop a library of components (including LECs) and their contracts for each level. Develop a simulation testbed. Integrate algorithms for component synthesis and refinement. Extend the abstract interpreter to reason about temporal properties and develop abstract domains for reasoning about LECs. Synthesize monitors for assumptions and unverified safety requirements.
Phase II	Extend support for the design framework and the libraries of components (including LECs) and their contracts for each level. Extend the simulation testbed. Extend the library of algorithms for component synthesis and refinement. Extend abstract interpreter to enforce safety properties by contract refinement. Synthesize monitors for unverified safety requirements and reduce monitoring overhead to 40%.
Phase III	Extend support for the design framework and the libraries of components (including LECs) and their contracts for each level. Complete the library of algorithms for component synthesis and refinement. Complete the simulation testbed. Demonstrate scalable verification of models at all levels of the LE-CPS design. Synthesize predictive monitors for unverified safety requirements and reduce monitoring overhead to 20%.

Table 2: How TA1 Interfaces with TA2 and TA3

TA1	Interfaces and Interaction
TA2	TA1 will provide the specification of assurance monitors and guards for conditional evidence, such as architectural and temporal constraints to be monitored, software implementation of synthesized monitors and their interface, specification of the benchmarks used in (simulation-based) testing. TA2 will provide support to TA1 for modeling machine learning architectures.
TA3	TA1 will provide properties and (conditional) evidences in the form of contracts at various levels of abstraction as well as their refinement relationships in support of safety and correctness claims. Evidence may include architectural constraints, safety and performance constraints on control and estimation components, and coverage estimates based on testing with or without simulation.

Table 3: Technical Challenges and Mitigation Approach for TA1

Technical Challenge	Mitigation Approach
Complexity and heterogeneity of LE-CPS design	Use a general contract-based formalism that supports capture of requirements, horizontal composition of models layers, and refinement maps between layers to support incremental addition of design detail. The formalism is general enough to allow the interoperation of different languages, logics, and tools.
Complexity of constructing the assurance case for a CPS design	Compose the overall assurance case from the assurance cases for individual components and from the horizontal and vertical design structures. Scalability arises from compositionality and from generating assurances at the appropriate level of abstraction. Provide static proofs that the design satisfies required safety constraints together with runtime monitoring of assumptions and unverified properties.
Constructing a safe CPS design using learning-enabled components (LECs)	Use a structured platform-based approach together with specialized non-linear abstract-interpretation domains to reason about LECs and verify their safety requirements.

4.2 Assurance Monitoring and Control

4.2.1 Current Approaches and State-of-the-Art

Autonomous vehicles need on-line sequential decision making capabilities to successfully complete tasks to meet mission goals. The decision making process needs to produce a sequence of actions that maximizes the expected value of mission utility. The autonomous system uses a system capability model and the environment state estimate to evaluate the potential actions it can take. The system capability model can be represented as a set of parameters; and a search method is required to identify near-optimal action sequences with respect to mission utility.

Consider a scenario where a USV needs to closely follow another vessel. The USV will need to use a hierarchical decision making approach to select the control action [102, 21, 92, 112, 108, 111]. At the lowest level it will generate a trajectory and use a feedback controller to track the planned trajectory in presence of disturbances. In order to avoid collisions and comply with COLREGS during the execution of planned trajectory, a monitoring system is necessary. In general, operation time assurance monitoring and control is an essential component of LE-CPS that has to strike the optimal tradeoff between safe execution of actions and achieving mission goals without being extremely conservative.

Unfortunately, complete safety cannot be assured during design time. Similarly, simulation cannot cover all possible cases, and unit and/or end-to-end testing cannot provide guarantees on LECs. Related research also reveals that very few USVs are equipped with on-board monitoring and collision avoidance systems [82]. Implementing monitoring systems to meet safety regulations, such as COLREGs imposes further challenges since the rules are often written in plain English for human operators. While a human operator can interpret and employ common sense to abide by these rules, imparting the same capability to an autonomous USV is challenging.

Capabilities of autonomous system might change due to environmental conditions (e,g., current, waves, wind, etc.). This might affect the trajectory tracking accuracy. The system needs to update its own capability model in a new environment by using safety-aware learning to operate effectively. The on-board perception system might not perform well and might make errors in classification of obstacles. The monitoring system will need to make sure that the classification system is operating over the expected input and should be able to detect cases where the classification system might not work due to lack of training data.

Here, we propose a comprehensive monitoring and control approach that does the following: (a) uses safety-aware learning methods and allows LE-CPS to safely explore new actions/maneuvers without risking the success of missions; (b) monitors architectural and safety constraints; and (c) tracks shifts in distributions. Our monitoring components use the formalism of TA1 and facilitates assurance reasoning by invoking TA3 in a feedback loop.

4.2.2 Approach

Scalable Safety-Aware Learning In most real-world scenarios, there are uncertainties in parameter estimates that describe the system capability model and the obstacles in the environment. Pre-operation testing can be used to reduce uncertainties for a small number of environment conditions. However, when the autonomous vehicle encounters a new operational environment for which test data does not exist, uncertainties in parameter estimates increase significantly. For example, consider an autonomous USV that operates in an environment with significant currents,

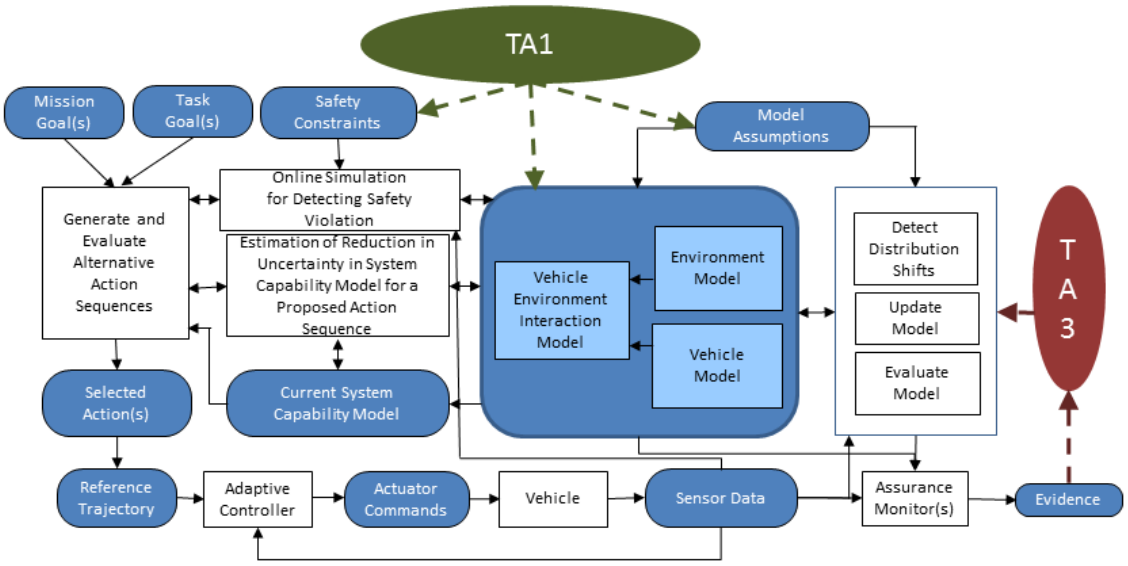

Figure 5: Shows the interactions between various components of TA2.

winds, waves, and fog. Any previously used parameter estimates for stopping distance, turning radius, and/or trajectory tracking accuracy will now be far from accurate with uncertainties on them increasing significantly. The system now has the following three options:

- It can use conservative parameter values—e.g., the largest possible value of stopping distance—for ensuring safety. This conservative strategy—in the same example—is successful in moving the vehicle very slowly in order to avoid collisions with dynamic obstacles. As new data becomes available, the system tries to improve the underlying dynamics model. However, the system converges to the dynamics model very slowly and hence its performance is poor with respect to the mission goal.
- It can suspend the mission temporarily and execute actions/maneuvers that enable it to learn the system capability model under new environmental conditions. But doing so makes the system temporarily unavailable to perform mission-related tasks.
- It can initially use conservative parameter values to ensure safety. It can then opportunistically use actions that not only make progress towards the mission goals but also enable the learning components to reduce uncertainty in the parameter estimates. This strategy carefully selects safe actions/maneuvers that not only reduce uncertainty in the system capability model but also make progress towards the mission goals.

Recent work in safe learning area computes states that can lead to unsafe situations based on the uncertainty in the dynamics model and only allow control actions that ensure system safety [45, 46, 95]. This enables the system to use learning based controllers. Methods have also been developed to ensure that unsafe regions of the state space are not explored during the learning process [90, 78, 77]. Gaussian processes based models are often used to learn system dynamics model [6, 19, 20]. Recent work in this area shows that Gaussian Processes in a Dirichlet Process mixture model can handle large changes in system dynamics [79].

Most of the previous approaches in this area can be viewed as variants of the first two approaches described above. We are interested in pursuing the third approach that can lead to much

faster reduction in uncertainty in state estimation without sacrificing safety. Our work will build upon previous work on learning system dynamics and avoiding unsafe states.

We propose to develop a new on-line learning-enabled decision making framework that enables the system to compose the right sequence of actions/maneuvers to reduce uncertainty in the system capability model opportunistically without suspending the progress towards the mission goals or compromising safety. Each candidate action/maneuver is evaluated based on three criteria. First, it is evaluated on the risk of violating a safety constraint by performing worst case analysis. Second, it is evaluated on its relevance to the mission goals. Third, it is evaluated on its expected information gain, i.e., reduction in uncertainty, using a surrogate version of the learning component. These three evaluations are combined to produce a cumulative mission utility value for each action/maneuver that drives our learning-enabled decision making.

We plan to use learning approaches based on Gaussian Process regression to estimate uncertainty in the system capability model. We will select appropriate approaches for estimating uncertainties in classification system based on the selected learning system. The underlying LECs are designed to accept only data that satisfy the monitor and guard conditions. By design, our framework also discards actions/maneuvers that violate safety constraints. The system will utilize a learning framework that will ensure that the known model structure is utilized to guide the learning process by enforcing the model structure constraints. The learning process will also ensure that the data violating modeling assumption does not corrupt the model, but if appropriate is used to update uncertainty estimates.

The safety-aware learning module faces the combinatorial problem of generating and evaluating sequences of actions that have to heed to safety constraints, mission goals, and information gain. Because uncertainties associated with the parameters are time-varying and trajectory-dependent, existing approaches are unlikely to be scalable. We therefore propose to develop new scalable search strategies to solve this problem efficiently. The problem of generating and evaluating sequences of actions can be posed in several frameworks, including a branch-and-bound search framework like Anytime A*, or an Markov decision processes (MDP) framework like a finite-horizon MDP. We plan to evaluate FastMap, a recently developed method by our group at USC, that can significantly improve the efficiency of the algorithms in the search framework as well as in the MDP framework [29].

While in many application domains, heuristic search begins after the specification of the start and goal states, FastMap preprocesses the extensional or intensional graphs over which search is carried out. By efficiently embedding nodes in Euclidean space, FastMap preprocesses a graph in near-linear time and produces admissible and consistent heuristics. For example, A* with FastMap heuristics produces optimal solutions 30-40 times faster than many other commonly used heuristics. If the available computational time requires us to use anytime search algorithms to accept suboptimal solution from mission utility point of view, the use of improved heuristics will lead to significantly better quality solutions. Using FastMap the Euclidean embedding can be precomputed in near-linear time for any update to the parameters of the vehicle-environment interaction model.

In finite-horizon MDPs, the same principle can be used to improve the performance of Value Iteration. FastMap can be used to preprocess and transform the state space of an MDP into an artificially created Euclidean space in near-linear time. Given a new mission goal, or any set of goal states, the Euclidean distances to their corresponding points is used as an initialization for each state. Value Iteration using FastMap is expected to converge much faster since its initialization and

Bellman updates are based on better information.

Scalable Assurance Monitoring An important component of TA2 is the ability to monitor architectural and safety constraints for assurance. Consider the assurance monitoring of an architectural constraint on the coefficient of friction μ. Suppose μ is modeled to be independent of the velocity v under the assumption that v is always less than a threshold velocity v_T. While this physical model is fairly accurate, it holds only under the assumption that $v \leq v_T$, therefore the constraint has to be continually monitored. Similarly, consider a safety constraint of the USV having to maintain a safe distance of at least δ from an obstacle. Monitoring architectural and safety constraints accounts for a lot of computational processing since each constraint has to be checked over and over again as a massive amount of sensor information continually flows in.

To increase the scalability of the monitoring algorithms, we propose novel techniques that we have successfully used in DARPA's RSPACE program. Our systematic use of a physical model of the autonomous vehicle also enables us to monitor architectural and safety constraints much more efficiently. In the same example, suppose we know that the USV has a maximum acceleration of $a_{\max}$. At any time t, if the velocity is $v_t < v_T$, we can conclude that it would require at least $\frac{v_T - v_t}{a_{\max}}$ time units to violate the architectural constraint. Therefore, no sensor information needs to be processed until time $t + \frac{v_T - v_t}{a_{\max}}$ for checking this constraint. At time $t + \frac{v_T - v_t}{a_{\max}}$, if the USV indeed has a velocity v_T, i.e., if it is on the verge of violating the architectural constraint, an appropriate control action is taken to avert the situation. If not, the USV has a velocity $v < v_T$; and the same reasoning can be applied recursively to compute the next point of time when the sensor information would really matter. This recursive reasoning procedure reduces the amount of sensor information that has to be processed in monitoring constraints. We have used similar constraint monitoring techniques in DARPA's RSPACE program to extract exponential increases in computational efficiency.

We envision two viable extensions of our current work to the objectives of TA2. First, we can extend it to uncertain environments where variables may or may not be controlled adversarially. Our approach in this context is very principled and is based on the idea of Variable Elimination (VE) in Artificial Intelligence (AI). VE is successfully used is many areas of AI, including constraint satisfaction, probabilistic reasoning, and hierarchical planning. Using VE, an architectural or safety constraint that involves a mix of observable/unobservable, controllable/uncontrollable, and discrete/continuous variables can be reduced to a substrate constraint that characterizes a dominant strategy of the controllable variables over the unobservable and uncontrollable variables. In the case of USVs, for example, the theory of VE provides a principled solution to how architectural and safety constraints can be efficiently monitored in either congenial circumstances, adversarial conditions, or for COLREGs.

The second important extension of our work is in the dynamic assessment of assurance case (DAAC) for TA3. Once again, the idea is to invoke the DAAC only when needed and not at all times when conditional evidence is available. The same kind of reasoning explained above can be used to detect the time intervals during which DAAC is not required. Doing so is expected to improve the scalability by orders of magnitude not only by decreasing number of DAAC initiations but also the communication between the TA2 and TA3 components.

Detecting Distributional Shifts When the LE-CPS is running in operational mode, its performance can change significantly if the distribution of the live data differs from that of the training data. These shifts may be caused by physical changes in the environment, adversaries trying to actively defeat the system, or faulty sensors. In general, there are four broad categories of reasons for distributional shifts. First, they can come from sensor failures. Second, they can come from modeling incompleteness such as, inaccurate models for wake dynamics in the USV domain. Third, they can come from a change of conditions between training data and live data, e.g., when the USV starts operating under dense fog, for which no training data was available. Fourth, they can come from physical changes to the vehicle, like deformation in one of the propeller blades, that necessitate changes to the vehicle-environment interaction model.

Our distribution shift detector continually monitors the distributions, detects any shifts, identifies the category and cause, invokes dynamic assurance, and either updates the model or raises an alert. We propose to build on our DARPA BRASS effort, where we developed an approach to detect sensor changes and failures and distinguish them from distributional shifts caused by other factors. We do this by learning the relationships among sensors from historical data and then exploiting their relationships to detect failures and deviations. Compared to many existing methods based on change-point/outliers detection [15, 49, 24, 9], our approach is different because it leverages relationships between multiple sensors. Although [36, 37, 35] also leverage such relationships, our approach is capable of examining different combinations of sensors and extracting only a small set of linear or nonlinear relationships between sensors.

In the second case, if no actions have been taken to avert the violation of a modeling assumption or an architectural constraint, we temporarily resort to a safe mode of operation. In the third case, we begin training on data gathered from the new environment and accumulate enough such data to train a new model without any structural changes to it. Here, we rely on the power of modern LECs to quickly enable this transition. In the fourth case, we make structural changes to the model using LECs and their concomitant stochastic contracts provided by TA1.

In operational mode, it is impractical to store all sensor data and use historical information for detecting distributional shifts. We therefore propose to develop a four-fold approach to detect distributional shifts efficiently. One, we test the basic general bounds on the tail probability of an input variable, defined by Markov, Chebyshev, and Chernoff inequalities [41]. Two, we extend the algorithms that track statistics and detect changes over a sliding window, such as ADWIN [22], by incorporating knowledge from the physical models to reduce computational costs. Three, we adopt commonly-used data reduction techniques, such as sampling, synopses and histograms, wavelets, and discrete Fourier transforms, to detect distributional shifts using compact representations of data [25, 114]. Four, we explore the potential applicability of modern ML classifiers [91] and ensemble-based methods [113] to identify these distributional shifts.

4.3 Dynamic Assurance

4.3.1 Current Approaches and Issues

A recent survey of safety engineering compliance practices relates to our dynamic assurance case challenge ([83]). The survey found that practitioners frequently relied upon *process-based evidence*, such as verification and validation plans and on safety management plans. When the practitioners provided *product-based* evidence they provided the requirements specification and test

Table 4: Project Plan for TA2

Phase	Plan
Phase I	Implement system with 2 LECs and code-level optimizations to reduce the monitoring overhead to 50%. Explore and implement framework for safety-aware learning. Adapt models and systems defined by TA1. Extend sensor failure detection in BRASS effort to detect distributional shifts. Develop APIs for integration with TA1 and TA3.
Phase II	Implement FastMap and heuristic Bellman updates to boost safety-aware Reinforcement Learning. Incorporate physical models of vehicle-environment interactions to allow scaling up to 4 LECs and reduce monitoring overhead to 30%. Extend sliding window and sampling-based techniques to detect distributional shifts.
Phase III	Focus on proposed novel algorithmic techniques to allow scaling up to 6 LECs and reduce monitoring overhead to 10%. Add features necessary to support physical trials.

Table 5: How TA2 Interfaces with TA1 and TA3

TA2	Interfaces and Interaction
TA1	• *Delivery interface* to accept fully implemented designs including all executables, libraries, and models along with an interface to review the version of deployed components and monitors. • *Monitoring interface* to specify the variables to monitor along with specifications on how to monitor, such as the monitoring frequency and functions.. • *Constraints interface* to provide distributions of monitoring variables, constraints, and state machines to implement monitoring capabilities. • *Safety templates* to specify actions, switch modes, and apply remedial actions schemes while assuring safety of the mission.
TA3	• TA2 will provide evidence, conditional evidence, and relevant sensor data. • An API to specify when to collect evidence, access historical evidence and sensor data. • An API for reporting Assurance Measures.

results. The study authors mention their intrigue that evidence types concerning risks and hazards are not among the most frequently reported. Their highest challenge is determining the confidence in evidence supporting a claim about system safety. Formal verification results were used by fewer than 30% of the study respondents. Completeness of the evidence is checked *manually* and through a manual predefined process, according to more than half of the respondents.

Regarding the structure of an assurance case, it has been popular for many years to refer to legal reasoning and graphical notations ([64]). The tree-structured *goal structuring notation* (GSN) is popular, and its cousin dubbed the *claim-argument-evidence* is an alternate representation. For larger arguments, the tree structures quickly become undesirable, fine for computer internal representation but clumsy for human-computer interfaces or printed documents. Moreover, there is no specification for the automated analysis of these trees, leaving the user to impose writing requirements to facilitate analysis ([97]). Somewhere in-between graphical notations and free-form text is semi-structured text ([53]), a DSL for writing case arguments while lending support for fancy editors and automated analysis.

Interesting assurance case structures that lend themselves to analysis have been developed. The Resolute language consists of both a logic and a computational sublanguage ([40]). The logic of

Table 6: Technical Challenges and Mitigation Approach for TA2

Technical Challenge	**Mitigation Approach**
Interpreting the designs and monitors produced by TA1 could be challenging.	Develop APIs to specify designs and monitors early in the program.
Action selection problem is computationally challenging in the presence of multiple learning enabled components and the approach produces sub-optimal decisions for applications requiring fast decision making	Use multi-resolution adaptive action primitives to speed up the search. Explore the use of parallel computing to speed up the search process by using multi-core architecture.
Detecting safety constraint violation using online simulation is too slow.	Use conservative approximation to develop simplified simulation that can run fast.
Models produce large uncertainties and uncertainty reduction process is very slow.	Develop specialized sequence of actions to rapidly reduce uncertainty and use them as meta-actions during search.

Resolute is an intuitionistic logic similar to pure Prolog, but augmented with explicit quantification. The Evidential Tool Bus (ETB) ([96]) is similar in syntax and semantics to Resolute. It provides a Datalog-style logic and is designed to combine evidence from a variety of sources. However, the focus of the ETB is on distribution and on provenance. The DSL L takes a semi-structured text approach ([14]). It provides an English-like language having a reasonable syntax and semantics for authors to write cases in an argument structure, but also provides analytic support for the document. The L tool set provides an answer set solver to reason about the case argument and return results to the author.

The safety practitioner survey revealed that simple textual templates are the most frequently found technique for evidence structuring. Study respondents reported rarely using argumentation-based graphical notation such as goal-structuring notation (GSN) or process models such as the OMG's software and system process engineering metamodel (SPEM). Regarding argumentation-based graphical notations, the study authors note "[t]he results suggest that a lot of research effort has been spent on a technique that has seen little industrial adoption thus far". We speculate, based on our own experience building tool sets for such graphical models, that the problem of quickly becoming unwieldy when modeling real-world products undermines the appeal of these techniques.

The notation problem can be addressed by domain-specific languages that enable analysis support. We introduced *answer set programming* (ASP) technology to provide this support [14], illustrating analysis of arguments, but the implementation lacks strong formalisms for handling learning, reasoning about uncertainty, and performance scalability. Although there are some existing algorithms for learning assurance rules in ASP [93, 11, 72, 10, 62, 63], they do not work well with large data sets. The best performing among them is the system XHAIL. For handling uncertainty popular formalisms such as Markov Logic Networks or its recently proposed ASP variant LP^{MLN} do not provide for non-monotonic specification of randomness. An ASP-specific variant P-log does have the required construct, but lacks in its ability to specify arbitrary weights.

The LE components introduce unique issues for safety assurance [70]. For example, neural networks lend themselves only to black box analysis but are wanted under verification and validation regimes that require white box analysis. Hazard analysis cannot be performed in the usual manner owing to the way the network algorithms work. Implicit, process-based safety assurance is insufficient for neural networks owing to a lack of concern for the functional behaviors of the soft-

ware. Explicit, product-based arguments such as proposed herein are more appropriate to consider the functional behaviors.

4.3.2 Approach

Domain Specific Language We refer to assurance cases having the conventional structure of a legal argument justifying trustworthiness for the system, in a *claim-argument-evidence motif*. We populate the assurance case differently from most in that we do not use the conventional tree structures. Our declarative approach employs a DSL as semi-structured text that lends itself to automated analysis. We use the design and verification content to populate the assurance case argument for safety property claims and evidence at build-time, coupled with run-time monitors produced alongside the evidence to be checked during operation. Both the build-time evidence and the production evidence in the learning-enabled environment integrate with the dynamic assurance case to maintain the argument for the trustworthiness of the system.

We will enhance an existing assurance case DSL called L. The DSL works in conjunction with an answer set program (ASP) solver to check assurance case arguments. We choose ASP technology here over other solvers such as SAT or SMT because, after grounding or translation, it provides better access modeling and specialization opportunities for domain-specific scalability. When the dynamic argument solution shows that trustworthiness can no longer be established, the the answer set solution will show the conflict by way of admitting multiple models. We bring the language, the editors, and solvers forward as our starting point for our dynamic assurance case (DAC) implementation.

Leveraging our systems-theoretic approach to design and implementation, we reason about the essential elements of *intent* and *state* in the DAC. Our DAC focuses on reasoning about the states of the system because these states are essential for understanding and managing the constraints imposed for LE-CPS safety. Moreover, through the DAC we adorn the states with associated design contracts, safety constraints, assumptions, intentions, and interactions among the states. For human interaction and debugging, when we provide system design intent in terms of state we can use this information from the DAC to support diagnostic explanations. The current L language resembles a blend of English and Prolog, and we provide editor support to emphasize ease of use.

Assurance Solver Implementations We will improve one or more reasoning engines for our DAC DSL to improve their suitability for dynamic assurance. There exists several excellent inference engines for basic ASP. These include Clingo [43] and Potassco [42], smodels [109], and DLV for basic ASP and some systems for languages such as P-log and LP^{MLN} that augment ASP with probabilities and weights. We anticipate building specialized solvers, rather than general-purpose solvers, to support our goals for the program.

Supporting Learning

During Phase I, we will modify L to accommodate learning of assurance rules. To have a common formal representation of various modules and components of the learning enabled autonomous systems, we aim to develop learning enabled components that can be characterized using the common representation.

The general sub-area of learning logic rules from data is called inductive logic programming (ILP). In statistical machine learning, it is common to learn a function from a series of $\langle x, y \rangle$ pairs where x denotes the input and y denotes the desired output. On the other hand, ILP deals with

learning a logic program H given some background knowledge and a dataset containing positive and negative examples. More formally, Given a set of positive examples E^+, negative examples E^- and some background knowledge B, an ILP algorithm [81] finds a hypothesis H such that, $B \cup H \models E^+$, $B \cup H \not\models E^-$. The possible hypothesis space is often restricted with a language bias that is specified by a series of mode declarations M.

Because of the mismatch between statistical machine learning and ILP definitions, one needs to convert all the $\langle x, y \rangle$ to create a global E^+ and E^- and extra care needs to be taken so that different $\langle x, y \rangle$ pairs do not interfere with each other. This approach fails to scale up. That is because, when we have a large number of $\langle x, y \rangle$ pairs, the ILP solvers will then have to deal with large ASP programs, We propose to develop a novel iterative and incremental approach where instead of making the above conversion, the learning of rules is done by iteratively going over each example $\langle x, y \rangle$ pair. Our approach is based on *context dependent learning*. By changing the inductive logic programming paradigm to context dependent learning, we will be avoiding dealing with large examples that is made of combining the set of examples. This allows us to pursue an iterative and incremental approach making our approach scalable. We expect an order of magnitude or more improvement.

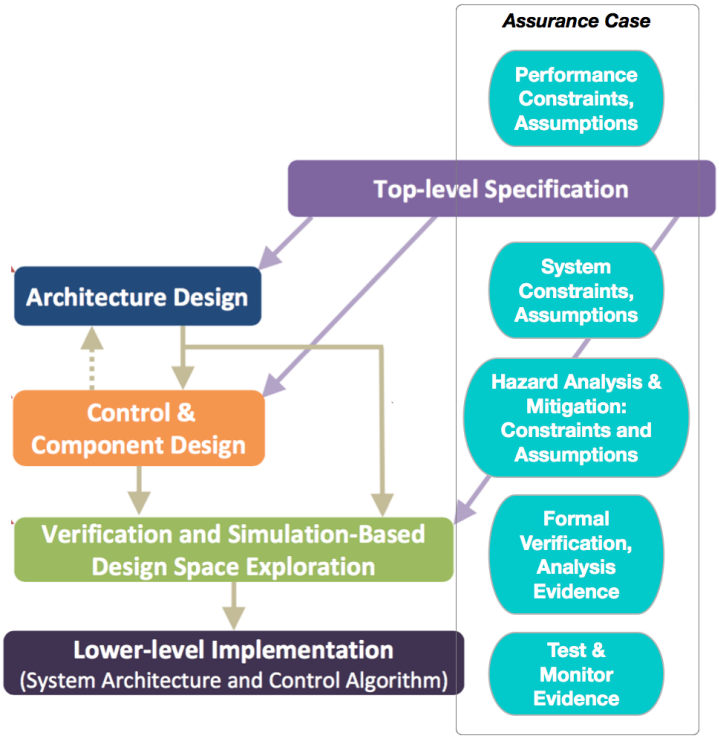

Figure 6: The system design and runtime activities provide different kinds of content, at different levels of abstraction or refinement, to the dynamic assurance case.

Another learning aspect we will address is allowing previously undefined predicates in H. Currently, in all ILP systems and algorithms the mode definitions of rules in H do not allow previously undefined predicates. The universal quantification plays an important role in the specification of temporal operators "*always*" and "*until*".

Supporting Uncertainty

During Phase II, we will add support for uncertainty. We propose to develop a new formalism that incorporates reasoning features crucial to assurance with respect to autonomous systems, such as causality, counter-factual reasoning, use of weights for computing probabilities, and probabilistic non-monotonicity – the ability to have new possible models as a result of new information. Probabilistic non-monotonicity is useful for non-naïve conditioning, as naïve conditioning may fail with new sensors that ask different kind of questions and as a result need a new set of possible worlds [50, 12].

Supporting Scaling

During Phase III, we will focus on ASP solver speed improvements. There exist excellent

inference engines for basic ASP, but none of them has emphasized speed. We will develop a more efficient system by modularizing the ASP programs and making domain-specific restrictions on the modules.

Fortunately, we have many innovative ideas for improving the performance of such solvers by several orders of magnitude. Our ideas are derived from the state-of-the-art techniques developed by our teammate for solving Weighted Constraint Satisfaction Problems (WCSPs). The WCSP serves as a substrate combinatorial problem for ASP augmented with probabilities and weights. In addition, the WCSP is representationally powerful enough to capture many other combinatorial problems in probabilistic reasoning. We have developed the state-of-the-art methods for solving WCSPs. We can apply these techniques to solve ASP augmented with probabilities and weights. Our intuitions include:

- The Constraint Composite Graph (CCG) [67, 68, 69] is a graph associated with a WCSP. It can be constructed in polynomial time. Solving a WCSP is equivalent to solving the Minimum Weighted Vertex Cover (MWVC) problem on its CCG.

- The *half-integral* property of the MWVC problem can be exploited towards a kernelization procedure for the WCSP [26]. We can stage a polynomial-time maxflow algorithm that preprocesses a WCSP and fixes the optimal values of a large number of its variables. The remaining variables constitute the *core* of the combinatorial problem.

- Message Passing (MP) is a well known technique used for large scale combinatorial problems [66]. It avoids exponential time complexity and constitutes an anytime algorithm. It is known to work very well on several inference problems in Information Theory [116]. We have shown that MP applied on the CCG of a WCSP produces solutions of much higher quality compared to MP applied directly on the WCSP [115].

ASP in general is an NP-complete problem. The stratified subset of ASP is quadratic. By having a modular language that separates the specification to a small non-stratified part and having a specialized solver for stratified ASP, we expect order of magnitude improvement in performance. A recent work [33] with similar ideas has shown good initial results. In addition, grounding can be done faster using a parallelized system in a multi-core environment.

Recent ASP competition results [44] note the apparent performance benefit of domain modeling and specialization. Because we develop a paired DSL and solver implementation, such techniques are applicable to us for scalability enhancements. The competition revealed that on decision problems like ours, the fastest general-purpose solvers produced solutions at a rate of 0.2 problems per second on the reference Linux platforms. We expect our specialized solvers to well exceed this performance level on our target application.

Integration and Demonstration We envision providing an API for content producers and consumers to use for interacting with the DAC. Our design and verification tools and run-time monitors and operators will assert assurance argument structures and features by way of the API. Any DAC control parameters will be done with a separate API. We envision using a representational state transfer (REST) stateless specification and synthesizing software adapters for this API.

As an example of high-level safety requirements, we would commence with explicit control system safety claims in our design activity. A representative set is given in [84] regarding com-

Table 7: Project Plan for TA3

Phase	Plan
Phase I	We will build the correspondence mechanisms for the safety-verification-monitor checks across the assurance case, focusing on contracts and constraints at various levels of abstraction in the design and implementation. We update an existing DSL and ASP solver to focus on this LE-CPS application, and build an API for exchanging constructs and data with the DAC. We show how the reasoning unfolds for successful and unsuccessful events.
Phase II	We will elaborate on our Phase I activities to include compositional arguments, uncertainty representations, learning-enabled ASP solvers, adjusting DSL, API, and solvers as required for successful demonstration. We anticipate having more formal verification and monitor data available at different levels of granularity, both with impact to the DSL. Initial efforts for ASP solver speed scale-up will be undertaken.
Phase III	We will emphasize scale-up with a primary concern for speed. We will implement modular solvers tailored to our domain, runtime environment, and operating constraints. We anticipate having more formal verification data available from TA1 at different levels of granularity, as well as more monitor types available from TA2, both with impact to the DSL.

Table 8: How TA3 Interfaces with TA1 and TA2

TA3	Interfaces and Interaction
TA1	• Receive assurance-related content in the form of contracts, assumptions, and safety constraints. Additional information such as bounds for any learning conducted within the assurance case itself. Much of this content becomes claims for the assurance case argument. • Receive content in the form of property specifications, open and closed proof obligations, assumptions, interactions, and constraints that led to those results. • Specify and implement an API for the assurance platform which accepts formal verification, simulation, and system testing claims along with assurance case.
TA2	• To support moving evidence into the DAC, TA3 will specify and implement an API for the assurance platform that can receive evidence, conditional evidence, and supporting data. • Evaluate DACs and provide assurance measure when conditional evidence is received. • To support visualization and logging, TA3 will provide an API and services for run-time monitoring of the DAC. • To support operational integrity, the design may require fault tolerance for DAC services. TA3 will provide features for checkpoints, redundancy, roll-backs, or other integrity features.

pleteness of requirements for safety-critical software. These address conditions such as reachability of safe and unsafe states, operator overrides, error handling, and so on. We will assert these as specific claims under a top-level safety claim for the assurance argument, then propagate the claims through our hierarchical design and hazard analysis procedure. Verification tools produce monitors to cover the behaviors of the system that could not otherwise be proven correct – for which more information is needed. We determine and justify the need for the run-time monitor and its conditional evidence as part of the design. Our DAC reasoning engine may be designed in as an integral part of the safety framework, thus having safety requirements of its own in the integrated system. We will provide all integration support via software installation and API services, and support demonstration planning and conduct as requested.

Table 9: Technical Challenges and Mitigation Approach for TA3

Challenge	**Mitigation Approach**
Hazard analysis data volume	The ASP technology we use for DAC can easily accommodate large volumes of data. The ASP reasoning technique enables filtering the results into easily contrasted outcomes so that the practitioner can identify conflicts.
DAC design and implementation validity	We consider validation from four perspectives: *construct validity* (on the theory of DAC and its implementation), *conclusion validity* (claims to evidence), *internal validity* (causal or logical relationships within an argument), and *external validity* (generalization of results).
Identifying sources of verification errors	Potential sources of verification errors include: modeling errors; testing harnesses and stubbed behaviors; property specification errors; incomplete test results; interpretation (assumption) errors; soundness vs. completeness inference errors in formal methods; artifact misinterpretation or corruption; and tool configuration managements and operation errors. To mitigate these risks we impose assurance arguments on our tool chains.
ASP solver speed improvement by specialization to application	Some expectation of availability and reliability of the DAC application will be imposed by the runtime environment. The challenges are accelerating the ASP solvers and handling time synchronization across multiple DAC instances. We explore multiple forms of modular and potentially translated implementations. We incorporate the DAC into the system design, potentially including hard real-time and safety downmode features.

5 Management Plan

The Principal Investigator for this effort is Dr. Craig Knoblock who is responsible for all aspects of the effort, will coordinate the parallel team efforts, and will ensure high levels of performance from individual team members. The Co-P/I, Dr. Michael Orosz, will provide project management and will assist all performers in the execution of the project. The project team is divided into three working groups (Figure 7) corresponding to Technical Areas 1-3, however, members of each team contribute across all project activities. Table 11 defines the major contributions of each project team member to the project tasks.

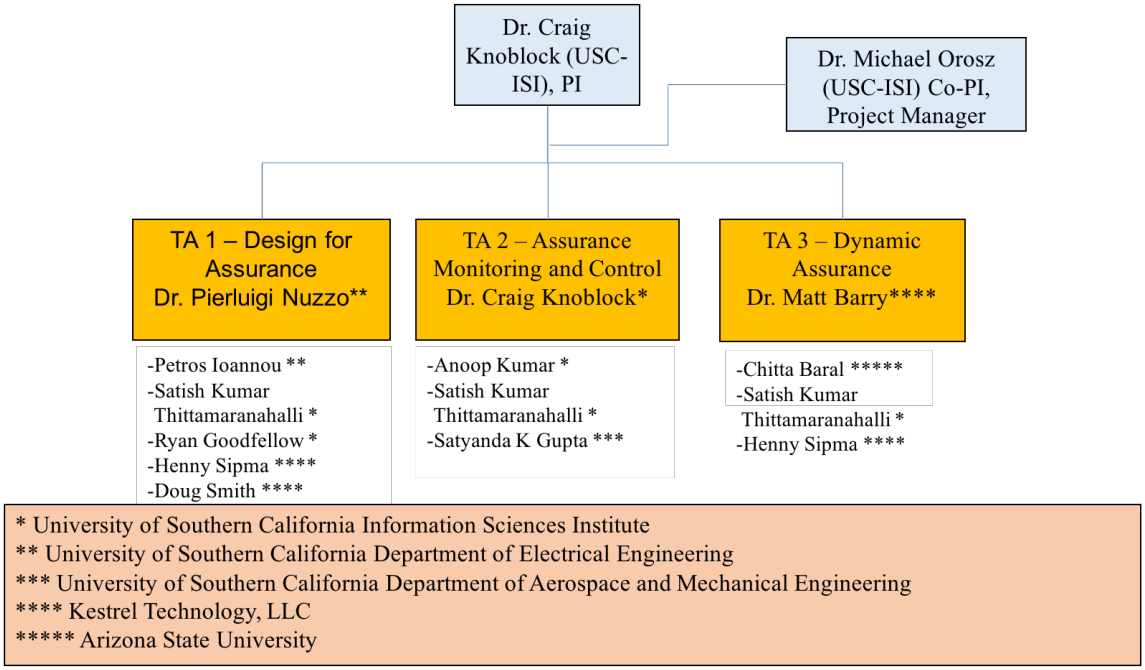

Figure 7: Organization Chart

Coordination: To maximize collaboration and reduce risk to project failure from lack of communication and technical exchange, we plan to employ a wide variety of working styles and communication/coordination so that all can contribute. At the core of our project will be regularly scheduled meetings bridging the diversely distributed team (Table 10). These meetings will address project status, identify challenges, implement risk mitigation strategies and participate in technology exchanges and system integration efforts (when appropriate)

Table 10: Project Meetings and Events

Meeting	**Frequency**
Conference calls among investigators (discuss project status, address concerns and project risks)	Weekly
Technical exchange and coordination meetings using Bluejeans or another videoconference technology	At least twice a month and more frequently as needed
Face-to-Face meetings (prior to P/I and demonstration meetings)	Every 3 to 6 months and more frequently (especially at the beginning of the project) as needed

Table 11: Key Project Team Member Responsibilities

Key Member	**Responsibilities**	**Tasks**
Dr. Craig Knoblock	Principal Investigator responsible for project, leads TA 2 – Assurance Monitoring and Control. Will lead the overall project and lead the TA2 team. Served as the PI on many DARPA projects and has sucessfully led many large teams. Effort on project: 25%	1.1.6, 1.2.2 1.2.3, 1.2.4, 1.3.4, 1.4.1, 2.1.6, 2.2.2 2.2.3, 2.2.4, 2.3.4, 2.4.1, 3.1.6, 3.2.2, 3.2.3, 3.2.4, 3.3.4, 3.4.1
Dr. Michael Orosz	Co-Principal Investigator responsible managing the day-to-day operations of the project, assist technical teams as needed, coordinate with TA4 teams. Has led many large complex multi-disciplined/multi-organizational projects in academic and industry environments. Effort on project: 50%	1.1.6, 2.1.6, 3.1.6, 1.4.1, 2.4.1, 3.4.1
Dr. Pierluigi Nuzzo	Co-Principal Investigator. Leads the TA 1 - Design for Assurance team and conducts research on the formal methods for the design of the TA1 system. Research experience on methodologies and tools for the design of cyber-physical systems; contracts, interfaces, and compositional methods for embedded system design; the application of automated formal methods and optimization theory to problems in embedded and cyber-physical systems. Effort on project: 2 months/year (16.6%)	1.1.1, 2.1.1, 3.1.1
Dr. Matthew Barry	Key personnel. Leads the TA 3 – Dynamic Assurance. He will conduct the research on the dynamic assurance case language editors and parsers, the run-time system, and system integrations. Effort on project: 66%	1.3.2, 2.3.2, 3.3.2
Dr. Chitta Baral	Key personnel responsible for learning assurance rules, supporting assurance rules with uncertainty and improving solver speed. Expertise on ASP solvers, which will be used to reason about the assurance cases. Effort on project: 20%	1.3.1, 2.3.1, 3.3.1
Dr. Doug Smith	Key personnel will support formal methods aspects of TA1, and lead the effort on abstract refinement. Expertise in field of automated correct-by-construction program generation. Effort on project: 40%	1.1.5, 2.1.5, 3.1.5
Dr. Henny Sipma	Key personnel who will support the program verification tasks under TA1. Will lead the effort on program verification. Effort on project: 45%	1.1.5, 2.1.5, 3.1.5, 1.3.2, 2.3.2, 3.3.2
Dr. Petros Ioannou	Key personnel responsible providing and extending the assurance test bed, which will be available at the start of the project for autonomous vehicles. Effort on project: 1 month/year (8.3%)	1.1.2, 2.1.2 (optional), 3.1.2 (optional)
Dr. Satyandra Kumar Gupta	Key Personnel providing autonomous command and control expertise to the TA-2 team. Will lead the research on safety aware learning on TA2. Past research on physics-aware decision making to facilitate automation. Effort on project: 1 month/year (8.3%)	1.2.1, 2.2.1, 3.2.1
Dr. Anoop Kumar	Key personnel providing support to the TA 2 project team. Will lead the research on monitoring & control and detecting distribution shifts. Effort on project: 50%	1.2.1, 1.2.2, 1.2.3, 1.2.4, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 3.2.1, 3.2.2, 3.2.3, 3.2.4
Dr. Satish Thittamaranahalli	Key personnel developing scalable algorithms for TA1, TA2, and TA3 project teams. Has extensive experience on scalable algorithm design, machine learning, and constraint reasoning. Effort on project: 50%	1.2.1, 1.2.2, 1.2.3, 1.2.4, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 3.2.1, 3.2.2, 3.2.3, 3.2.4, 1.1.4, 2.1.4, 3.1.4
Dr. Ryan Goodfellow	Key personnel providing support to the TA-1 project. Will lead the research on simulation-based testing. Has extensive experience on simulation-based testing. Effort on project: 30%	1.1.3, 2.1.3, 3.1.3

6 Personnel, Qualifications and Commitment

Dr. Craig Knoblock, the PI on this effort, is a Research Professor of both Computer Science and Spatial Sciences at the University of Southern California (USC) and Director of the Intelligent Systems Division at the USC Information Sciences Institute. He received his Ph.D. from Carnegie Mellon University in computer science. In previous projects he has worked on developing scalable approaches to execution monitoring, accurate detection of sensor failures, and automatic modeling and reconstruction of sensors. He has published more than 300 journal articles, book chapters, and conference papers on these topics. Dr. Knoblock is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI), a Distinguished Scientist of the Association of Computing Machinery (ACM), a Senior Member of IEEE, past President and Trustee of the International Joint Conference on Artificial Intelligence.

Dr. Michael Orosz, a Co-PI on this effort, is a Research Associate Professor of Civil and Environmental Engineering at the University of Southern California (USC) and Research Director of the Decision Systems Group at the USC Information Sciences Institute. Dr. Orosz has over 30 years' experience in commercial and government software development, basic and applied research, project management, academic research and has developed and deployed several commercially successful products. His research interests are in machine learning and decision analytics as applied to intelligence analysis and autonomous command and control such as smart building controls. Dr. Orosz has extensive experience in managing large complex multi-disciplined/multi-teamed research projects. He received his Ph.D. in computer science from the University of California, Los Angeles.

Dr. Pierluigi Nuzzo, a Co-PI on this project, is an Assistant Professor in the Department of Electrical Engineering at the University of Southern California. He received the Ph.D. in Electrical Engineering and Computer Sciences from the University of California at Berkeley. His research interests include: methodologies and tools for cyber-physical system and mixed-signal system design; contracts, interfaces and compositional methods for embedded system design; the application of formal methods and optimization theory to problems in embedded and cyber-physical systems and electronic design automation. Prof. Nuzzo received a Marie Curie Fellowship from the European Union in 2006, the University of California at Berkeley EECS departmental fellowship in 2008, the IBM Ph.D. Fellowship in 2012 and 2014, and the David J. Sakrison Memorial Prize in 2016 for his doctoral research.

Dr. Satyandra K. Gupta is Smith International Professor in the Department of Aerospace and Mechanical Engineering at the University of Southern California. He served as a program director for the National Robotics Initiative at the National Science Foundation from September 2012 to September 2014. Dr. Gupta's interest is in the area of physics-aware decision making to facilitate automation. He has published more than 300 technical articles. He is a fellow of the American Society of Mechanical Engineers (ASME) and editor of ASME Journal of Computing and Information Science in Engineering. Dr. Gupta has received the Young Investigator Award from the Office of Naval Research in 2000, CAREER Award from the National Science Foundation in 2001, Presidential Early Career Award for Scientists and Engineers (PECASE) in 2001, Invention of the Year Award at the University of Maryland in 2007, Kos Ishii-Toshiba Award from ASME in 2011, and Excellence in Research Award from ASME in 2013.

Ryan Goodfellow is a computer scientist at ISI working in combined cyber physical simulation and emulation platform development. His formal background is in simulation algorithms and

modeling techniques using differential-algebraic equations (DAE). He has applied this knowledge in the CPS space by integrating DAE modeling languages and simulation engines with network testbeds to create comprehensive scientific experimentation platforms for cyber-physical systems. These experimentation platforms have been used in the power grid research space.

Dr. Petros Ioannou is a Professor in the Department of Electrical Engineering, Director of the Center for Advanced Transportation Technologies and Associate Director for Research for the DOT supported University Transportation Center at USC. He received his MS and PhD from the University of Illinois at Urbana Champaign in Mechanical and Electrical Engineering, respectively. His research interests are in robust adaptive control, vehicle dynamics and control, human factors and safety, automated vehicles, nonlinear systems and Intelligent transportation Systems. He received the 2016 IEEE Transportation Technologies field award and the 2016 IEEE Control system society Transition to Practice Award. He is a Fellow of IEEE, IFAC and IET and author/coauthor of 8 books and over 400 papers.

Dr. Matthew Barry will serve as lead for the TA3 tasks. Dr. Barry currently is CEO at Kestrel Technology LLC, and previously spent 20 years in NASA space mission operations at the Jet Propulsion Lab and Johnson Space Center. At NASA Headquarters he led the introduction of dependability case requirements and plans for flight computing systems in upcoming manned space exploration missions, as well as the development of Agency-level software-related safety-critical control system requirements. He recently served as a Principal Investigator on DHS/Cyber S&T STAMP (Static Tool Analysis Modernization Program), DARPA CSFV (Crowd Sourced Formal Verification), three NASA Aeronautics R&D projects, and the AFRL-sponsored Static Analysis of Numerical Algorithms project. Dr. Barry earned BSME, MS, and PhD degrees in mechanical engineering, and an MBA degree, from Rice University.

Dr. Henny Sipma will support the program verification tasks under TA1. Dr. Sipma currently is the CTO at Kestrel Technology LLC. She has spent the past 10 years with Kestrel Technology as a static analysis expert; previously developed and taught static analysis techniques as senior research associate at Stanford University for eight years; and developed industrial process controls as an senior systems analyst at Shell. She has been Principal Investigator or company lead on several recent R&D projects for Federal agencies, including two projects under the IARPA STONESOUP (Securely Taking On New Executable Software of Uncertain Provenance) program; the DHS Cyber S&T Gold Standard project; and the DARPA-sponsored STAC (Space-Time Analysis for Cybersecurity) and MUSE (Mining and Understanding Software Enclaves) programs. Dr. Sipma earned MS and PhD degrees in computer science from Stanford University.

Dr. Douglas R. Smith will support formal methods aspects of TA1, including the enforcement of safety properties and the generation of monitors. He is President of Kestrel Technology LLC and Principal Scientist at Kestrel Institute. He is a Fellow of the American Association of Artificial Intelligence (AAAI) and an ASE Fellow (Automated Software Engineering). From 1986 to 2000, he taught an advanced graduate course on correct-by-construction software development at Stanford. Other recent projects focused on automated policy enforcement [104, 105], synthesis of secure network protocol codes, and the synthesis of high-performance constraint-solvers[107, 106]. Dr. Smith has over 30 years experience in the field of automated correct-by-construction program generation and has published over 100 papers. He has one patent. He received the Ph.D. in Computer Science from Duke University

Dr. Chitta Baral is a Professor in the Department of Computer Science and Engineering at Arizona State University. He will support the TA3 efforts on Learning assurance rules, supporting

assurance rules with uncertainty and improving solver speed. Dr. Baral has expertise in various aspects of autonomy and Artificial Intelligence. He wrote the first book on answer set programming (published by Cambridge University Press) the formal language behind our assurance rules. Some of his other works relevant to this proposal are: goal specification for autonomous systems, automatic construction of control rules for autonomous systems that satisfy given goals, combining machine learning with reasoning in various contexts, including image understanding.

Dr. Satish Kumar Thittamaranahalli (T. K. Satish Kumar) leads the Collaboratory for Algorithmic Techniques and Artificial Intelligence (CATAI) at USC's Information Sciences Institute. He has published over 60 papers on numerous topics in Artificial Intelligence spanning such diverse areas as Constraint Reasoning, Planning and Scheduling, Probabilistic Reasoning, Robotics, Combinatorial Optimization, Approximation and Randomization, Heuristic Search, Model-Based Reasoning, Knowledge Representation and Spatio-Temporal Reasoning. He and is a winner of the 2016 Best Robotics Paper Award and the 2005 Best Student Paper Award from the International Conference on Automated Planning and Scheduling. Dr. Kumar received his PhD in Computer Science from Stanford University.

Dr. Anoop Kumar is a senior computer scientist at USC ISI and has broad expertise in machine learning, statistical modeling, and software engineering. Dr. Kumar is the technical lead on the DARPA RSPACE program and has played a vital role in developing a system that fuses air operations data from multiple sources, maintains world state, and issues warnings. Previously, he led the research and development of the BBN's election forecasting system for the IARPA OSI program. Dr. Kumar helped design and develop BBN's commercially available, hosted speech and medical transcription services offering.

Project	Status	Hours		
		P1	P2	P3
Craig Knoblock				
Safeguard	Pro	770	641	641
ELICIT	Cur	308	256	120
WTNIC	Cur	11	0	0
EFFECT	Cur	641	107	0
LinkedMaps	Cur	203	25	0
PRINCESS	Cur	608	96	0
SCHARP	Cur	481	54	0
MINT	Pen	650	534	285
Michael Orosz				
Safeguard	Pro	1560	1300	1300
SMC/SY	Cur	1803	0	0
Matthew Barry				
Safeguard	Pro	2078	1690	1554
Starlite	Cur	1840	1692	0
Anoop Kumar				
Safeguard	Pro	1560	1300	1300

Project	Status	Hours		
		P1	P2	P3
Pierluigi Nuzzo				
Safeguard	Pro	520	433	433
Mirage	Cur	433	0	0
Satyandra Gupta				
Safeguard	Pro	260	217	217
Human	Cur	22	0	0
Vehicles	Cur	36	0	0
Robot	Cur	116	0	0
Assembly	Cur	33	0	0
Solar	Cur	4	0	0
Petros Ioannou				
Safeguard	Pro	260	217	217
CPS	Cur	130	0	0
Ryan Goodfellow				
Safeguard	Pro	936	780	780
STEAM	Cur	416	0	0

Project	Status	Hours		
		P1	P2	P3
Chitta Baral				
Safeguard	Pro	659	485	485
PostdocBP	Cur	176	0	0
Languages	Pen	528	264	264
CAREER	Pen	88	44	44
CHS	Pen	510	255	0
Doug Smith				
Safeguard	Pro	1222	984	840
RSPACE	Cur	342	0	0
PLANX	Cur	154	0	0
HACCS	Pen	923	769	769
Henny Sipma				
Safeguard	Pro	1372	962	840
STAC	Cur	797	0	0
Satish Thittamaranahalli				
Safeguard	Pro	1560	1300	1300
MapF	Cur	103	103	0

Table 12: Individual commitments of key personnel

7 Capabilities

USC has strengths in number of areas that are closely related to the proposed work:

- Dr. Nuzzo has done extensive research on contracts and compositional methods for heterogeneous system design and design space exploration, with application to aircraft electric power systems and environmental control systems. His work has helped transition rigorous system design foundations, innovative design methodologies, and new systems engineering paradigms to industry (IBM, United Technologies).
- Dr. Satyandra K. Gupta has worked on autonomous surface vehicles, autonomous ground vehicles for operation on rugged terrains, and autonomous flapping wing aerial vehicles. His group has developed a hierarchal decision making approach for realizing autonomous systems. His group has also developed new methods for learning reactive behaviors in adversarial environments and COLREGS compliant trajectory planning.
- Dr. Knoblock has developed methods that learn the relationships between sensors to both identify failures and changes in sensor and reconstruct those sensors, providing estimates of the accuracy of the reconstructed sensors.
- Ryan Goodfellow has extensive experience in simulation based testing through high-fidelity CPS testbed environment development and operation, using the Deter network testbed as the core which has supported several large scale government projects from a variety of agencies and thousands of users.
- Dr. Ioannou worked on several DOT funded projects on automated vehicles and intelligent highway systems where he demonstrated his vehicle control designs for safety and performance on actual automated vehicles in test trucks and I-15 highway.
- Drs. Knoblock, Kumar, and Thittamaranahalli have developed highly scalable approaches for monitoring message traffic to identify potential problems and issue warnings and alerts.
- Dr. Thittamaranahalli has developed state-of-the-art methods for efficiently solving large-scale search and optimization problems.

Kestrel Technology's strength is in program analysis, specifically static analysis of both source and binary targets. The company performs applied R&D and product development for a variety of static analysis applications pivoting primarily on the abstract interpretation technique. The company recently initiated development of program analysis applications using logical equivalence techniques. As a provider of verification evidence in the form of mathematical proofs, the company also has expertise in the design and development of assurance case arguments for high-integrity systems using such evidence.

Chitta Baral at Arizona State University has developed various software to learn assurance rules and various ASP solvers, which he has made available as open-source.

Most of the software carried forward for implementation or derivation is open source. The single exception is Kestrel Technology's *KT Advance* static analysis tool (TA1), in particular the abstract interpretation engine therein, which is company proprietary and is US EAR export-controlled. We will continue to provide the Federal government a restricted use license for that particular item.

There are no specialized facilities, data, or GFE required for this effort.

8 Statement of Work

We propose work for TA 1 – TA 3 for all three phases. All tasks span the four years of the program. For each task we provide an objective, the high-level approach (focusing on the responsibilities of each contributing organization), and the specific approach and milestones planned for each task for each phase. On all tasks, we will deliver design documents, software implementations, demonstrations, and publications. With the exception of several tasks accomplished by Kesler Technology, LLC, all tasks that accomplished at a university (USC/ISI, USC, and ASU) are believed to be fundamental research.

Tasks 1.1.1, 2.1.1, 3.1.1 -Design for Assurance System Models and Formal Verification (USC)
Objective: Develop contract-based formalisms and mapping tools to represent and reason about LE-CPSs at multiple levels of abstraction and generate assurance cases. Undertake scalable formal verification and synthesis via Satisfiability Modulo Convex Programming.
Approach: Develop modeling formalisms to represent components and contracts for LE-CPSs, including physical plant (e.g., autonomous vehicle, sensors, actuators, environment, controllers, and learning components. Formalisms will encompass different control and learning architectures (e.g., neural networks, statistical methods, graphical models, ensemble methods, decision trees) and support mapping between abstractions. Develop a formal domain-specific language to capture and formalize requirements on LE components, systems, and their dynamics as contracts. Develop a unifying framework and efficient algorithms to reason about the combination of discrete and continuous dynamics and constraints in the presence of uncertainties in LE cyber-physical systems
Phase 1 (1.1.1): Milestone 1: Develop initial design followed by development and testing of individual components. Milestone 2: Library of components and contracts for the autonomous vehicle application driver. Milestone 4: Library of components and contracts for the platforms provided by TA4 performers. Extension of the methodology and to support up to 20 continuous dimensions and 2 learning components for the 2 application drivers from TA4. Milestone 6: -Prototype toolkit (software package) for capturing requirements, for translating them into contracts, for analyzing and validating them using contract operations and relations. Prototype toolkit for capturing probabilistic requirements and behaviors of LE components, systems, and their dynamics, for translating them into stochastic assume-guarantee contracts, for analyzing and validating them using contract operations and relations, and for synthesizing design and verification artifacts from contracts. Extension of the SMC framework and toolkit to support reactive and robust task and trajectory planning in the presence of uncertainties.
Phase 2 (2.1.1) Milestone 7: Refinement of design. Milestone 9: extension of methodology, design, toolkits and libraries to support 40 continuous dimensions, 4 LECs, 30% monitoring overhead. Extension of the SMC framework and toolkit from Phase 1 to support verification and synthesis on system with 40 dimensions and 4 LECs. Milestone 10: Demonstration of the SMC framework and toolkit. Contribution to Phase II report and dissemination of the results in conferences and journals.
Phase 3 (3.1.1) Milestone 11: Update design based on Phase II demo. Milestones 12-13: extend methodology, design, toolkits and libraries to support 100 dimensions, 6 LECs and 10% monitoring overhead. Milestone 14: Undertake Phase III demonstration on both platforms and submit final project report.
Tasks 1.1.2, 2.1.2, 3.1.2: Design for Assurance Testbed (USC)
Objective: Develop a simulation test bed for data generation and LE algorithm testing, redesign and/or refinement. Simulator used as the test bed until the TA4 platforms are available. Test bed will be used for internal research/prototype after TA4 platform availability.
Approach: Leverage previous work on microscopic traffic simulations in urban and rural environments using the commercial software VISSIM and Vortex Studio and built in extensions for automated driving. Develop testbed for autonomous vehicles in road/off-road environments to allow LEs to collect data, learn and make control decisions on line and in real time by simulating scenarios. The testbed together with analytical tools used to refine and redesign LEs and control algorithms by taking into account effects revealed by the simulation and not accounted for in the design stage. In the event the TA4 platforms are not available, the test bed will be extended further by integrating all the LE components, controllers and sensors for demonstration purposes and evaluation of the proposed methodology.
Phase 1 (1.1.2): Milestones 1-2: Extension of existing simulator test beds. Milestones 3-5: Testing of individual components under normal and unpredicatble situations and demonstrating the results in VISSIM under several different driving scenarios.
Phase 2 (2.1.2) – Optional: Milestones 7-8: Extension of existing simulator test beds to support the TA1-TA3 teams. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead.
Phase 3 (3.1.2) – Optional: Milestones 11-12: Extension of existing simulator test beds to support the TA1-TA3 teams. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead.
Tasks 1.1.3, 2.1.3, 3.1.3: Design for Assurance Simulation Based Testing (USC/ISI)
Objective: Develop external Discrete Control Mechanisms for OpenModelica. Develop/package virtual-machine based static time dilation systems. Undertake network testbed integration and develop physical system behavioral analysis tooling.
Approach: Leverage previous external discrete control mechanisms for DAEs, implement similar facilities for OpenModelica to allow LEs to observe and control a physical system over a network. Contributions pushed back upstream to OpenModelica project. Implement DieCast for modern libvirt. Develop tooling to deploy integrated CPS models on the Deter network testbed. Apply modern DAE control theory in the form Modelica analysis packages usable by non DAE experts.
Phase 1 (1.1.3): Milestones 1-2: Initial CPS simulation concept and components. Milestones 3-5: Testing of individual components under normal and unpredictable situations and demonstrating the results capable of meeting 20 dimensions, 2 LECs and 50% or under monitoring overhead conditions. Milestone 6: Demonstrate technology in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.
Phase 2 (2.1.3): Milestones 7-8: Apply lessons learned from Phase I and extend existing simulations to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Contribute to Phase II final report and disseminate software and publications.

Phase 3 (3.1.3): Milestones 11-12: Apply lessons learned from Phase II and extend existing simulations to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Contribute to Phase III final report and disseminate software and publications.

Tasks 1.1.4, 2.1.4, 3.1.4: Scalable Algorithms for Formal Verification (USC/ISI)

Objective: Develop innovative algorithms for scalable formal verification.

Approach: Use state-of-the-art techniques for solving combinatorial problems with discrete/continuous variables and hybrid constraints.

Phase 1 (Task 1.1.4): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Integrate framework that is capable of supporting 20 dimensions, 2 LECs and 0.1x trials to assurance. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.

Phase 2 (Task 2.1.4): Milestones 7-8: Apply lessons learned from Phase I and extend existing design to support 30 dimensions, 3 LECs and 0.05x trials to assurance. Milestones 9-10: Demonstrate technology capable of supporting 40 dimensions, 4 LECs and 0.01x trials to assurance. Participate in Phase II demonstration, contribute to Phase II final report and disseminate software and publications.

Phase 3 (Task 3.1.4): Milestones 11-12: Apply lessons learned from Phase II and extend design/approach to support 70 dimensions, 5 LECs and 0.005x trials to assurance. Milestones 13-14: Demonstrate technology capable of supporting 100 dimensions, 6 LECs and 0.001x trials to assurance. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.

Tasks 1.1.5, 2.1.5, 3.1.5: Design for Assurance Program Verification (Kestrel Technology, LLC)

Objective: Develop and integrate program analysis and monitor synthesis functionality with TA1 functions and services and integrate combined TA1 functions with TA4 platform.

Approach: Integrate existing analysis tools into development environment. Design and implement abstract domains and properties for one or more modeling layers. Design and implement analyzer front-end for modeling layers. Implement test framework for verification tools. Implement content providers and/or consumers for DAC via DAC API. Leverage existing algorithms and tools to generate monitors for assumptions and unproven safety constraints. Integrate program analysis and monitor synthesis functionality with TA1 functions and services, integrate combined TA1 functions with TA4 platform. Prepare software and data installation kits and operating instructions;install software and confirm configuration.

Phase 1 (1.1.5) : Milestones 1-2: Initial framework design and unit tools, TA1-TA3 interfaces defined. Milestones 3-5: Testing of individual components/tools capable of meeting 20 dimensions, 2 LECs and 50% or under monitoring overhead conditions. Milestone 6: Demonstrate technology in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.

Phase 2 (2.1.5): Milestones 7-8: Apply lessons learned from Phase I and extend existing design to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Contribute to Phase II final report and disseminate software and publications.

Phase 3 (3.1.5): Milestones 11-12: Apply lessons learned from Phase II and extend existing simulations to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Contribute to Phase III final report and disseminate software and publications.

Tasks 1.1.6, 2.1.6, 3.1.6: System integration, deployment, and testing (USC/ISI)

Objective: Develop and implement integration, testing and deployment plan supporting TA1 for all three phases.

Approach: Develop an internal TA1 integration and testing plan (unit tests, etc.) and, in close collaboration with TA2 and TA3 performers on project, develop an overall TA1-TA3 integration and testing plan. Working with TA4 performers, extend and execute plan for TA4 platform (when available).

Phase 1 (1.1.6): Milestones 1-2: Develop initial integration and testing plan and implement on unit testing. Milestones 3-5: Oversee integration and testing of TA1-TA3 components for system capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Complete integration of technology into TA4 testbeds, contribute to Phase I final report and disseminate software and publications.

Phase 2 (2.1.6): Milestones 7-8: Apply lessons learned from Phase I and extend existing integration and testing plan to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Complete integration of technology into TA4 platforms. Contribute to Phase II final report and disseminate software and publications.

Phase 3 (3.1.6): Milestones 11-12: Apply lessons learned from Phase II and extend existing integration and testing plan to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.

Tasks 1.2.1, 2.2.1, 3.2.1: Safety Aware Learning (USC)

Objective: Enable the system to learn efficiently without violating safety constraints.

Approach: Integrate LECs with search methods to select the optimal actions/maneuvers to maximize mission utility.

Phase 1 (Task 1.2.1): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Integrate two LECs with search methods and integrate into framework that is capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.

Phase 2 (Task 2.2.1): Milestones 7-8: Apply lessons learned from Phase I and extend existing design to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Participate in Phase II demonstration. Contribute to Phase II final report and disseminate software and publications.

Phase 3 (Task 3.2.1): Milestones 11-12: Apply lessons learned from Phase II and extend design/approach to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.

Tasks 1.2.2, 2.2.2, 3.2.2: Assurance Monitor and Guards (USC)

Objective: Build scalable algorithms for assurance monitoring of architectural and safety constraints

Approach: Use physical models to reduce processing of sensor information for assurance monitoring. Use Variable Elimination to handle uncontrollable, Adversarially controlled, or unobservable variables

Phase 1 (Task 1.2.2): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Develop monitors for two LECs and integrate into framework that is capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Develop APIs for integration with TA1 and TA3. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.
Phase 2 (Task 2.2.2): Milestones 7-8: Apply lessons learned from Phase I, incorporate physical models of vehicle-environment interactions and extend existing design to support 30 dimensions, 3 LECs and incorporate physical models to bring down monitoring overhead to 40% or less. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Participate in Phase II demonstration. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (Task 3.2.2): Milestones 11-12: Apply lessons learned from Phase II and identify core constraints to monitor and correlation between variables to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.2.3, 2.2.3, 3.2.3: System integration, deployment, and testing: (USC/ISI)
Objective: Develop and implement integration, testing and deployment plan supporting TA2 for all three phases.
Approach: Develop an internal TA2 integration and testing plan (unit tests, etc.) and, in close collaboration with TA1 and TA3 performers on project, develop an overall TA1-TA3 integration and testing plan. Working with TA4 performers, extend and execute plan for TA4 platform (when available).
Phase 1 (1.2.3): Milestones 1-2: Develop initial integration and testing plan and implement on unit testing. Milestones 3-5: Oversee integration and testing of TA1-TA3 components for system capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Complete integration of technology into TA4 testbeds, contribute to Phase II final report and disseminate software and publications.
Phase 2 (2.2.3): Milestones 7-8: Apply lessons learned from Phase II and extend existing integration and testing plan to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Complete integration of technology into TA4 platforms. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (3.2.3): Milestones 11-12: Apply lessons learned from Phase II and extend existing integration and testing plan to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.2.4, 2.2.4, 3.2.4: Detecting Distributional Shifts (USC)
Objective: Develop a comprehensive framework to detect distribution shifts in LECs
Approach: Extend our prior work on sensor failure detection to distribution shifts. Implement an approach that looks at single variable, sliding window, and distributions and employs classifiers and ensemble methods.
Phase 1 (Task 1.2.4): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Develop framework that is capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Extend sensor failure detection in BRASS effort to detect distributional shifts. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.
Phase 2 (Task 2.2.1): Milestones 7-8: Apply lessons learned from Phase I and implement sliding window and sampling based methods to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Participate in Phase II demonstration. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (Task 3.2.1): Milestones 11-12: Apply lessons learned from Phase II and implement data reduction and machine learning techniques to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.3.1, 2.3.1, 3.3.1 - Checking Assurance Case Arguments for Dynamic Assurance – (ASU)
Objective: Enhance assurance case DSL to accommodate learning of assurance rules. Enhance Dynamic Assurance Case (DAC) implementation to support uncertainty. Enable ASP solver speed improvements
Approach: We will develop algorithms and an implemented module that can learn assurance rules from a set of input-output pairs. We will illustrate the scalability of our method as compared to existing Inductive Logic Programming methods. We will develop a variant of L that incorporates various uncertainty and automated reasoning related features such as causality, counterfactual reasoning, use of weights for computing probabilities and probabilistic non-monotonicity. We will develop a highly efficient ASP reasoning system (that forms the heart of our assurance case DSL) by modularizing the ASP programs and making domain specific restrictions (such as stratification on a big part of the program) on the modules
Phase 1 (Task 1.3.1): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Integrate two LECs with search methods and integrate into framework that is capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.
Phase 2 (Task 2.3.1): Milestones 7-8: Apply lessons learned from Phase I and extend existing design to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Participate in Phase II demonstration. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (Task 3.3.1): Milestones 11-12: Apply lessons learned from Phase II and extend design/approach to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.3.2, 2.3.2, 3.3.2 - Program Verification and Run-Time Monitoring for Dynamic Assurance (Kestrel Technology, LLC)
Objective: Develop the DAC language, the API for DAC interaction between TA1/TA2/TA3 and implement the technology in the three phases
Approach: Develop initial DAC language and APIs and extend based on testing against internal and TA4 provided scenarios.

Phase 1 (Task 1.3.2): Milestone 6: An initial DSL grammar specification; a DAC API Specification, a program client/server protocol and content specification for use interacting with the DAC; initial learning-enabled solver; and integrated DAC API-solver software for the demonstration platform
Phase 2 (Task 2.3.2): Milestone 7: Updated design/plans based on Phase I lessons learned. Milestone 10: deliver a program client/server protocol and content specification for use interacting with the DAC; initial uncertainty-enabled solver; and integrated DAC API-solver software for the demonstration platform.
Phase 3 (Task 3.3.2): Milestones 11: Apply lessons learned from Phase II and extend design/plan. Milestone 14: Deliver a program client/server protocol and content specification for use interacting with the DAC; final and modularity-enabled solver; and integrated DAC API-solver software for the demonstration platform.
Tasks 1.3.3, 2.3.3, 3.3.3: Scalable Algorithms for Checking Assurance Arguments (USC/ISI)
Objective: Develop innovative algorithms for efficient dynamic assessment of assurance cases.
Approach: Use state-of-the-art techniques for solving Weighted CSPs to solve ASPs with weights and probabilities.
Phase 1 (Task 1.3.3): Milestones 1-2: Develop initial design plan and initial concepts. Milestones 3-5: Integrate framework that is capable of supporting 20 dimensions, 2 LECs and 10 conditional evidences. Milestone 6: Participate in Phase I demonstration, contribute to Phase I final report and disseminate software and publications.
Phase 2 (Task 2.3.3): Milestones 7-8: Apply lessons learned from Phase I and extend existing design to support 30 dimensions, 3 LECs and 50 conditional evidences. Milestones 9-10: Demonstrate technology capable of supporting 40 dimensions, 4 LECs and 100 conditional evidences. Participate in Phase II demonstration, contribute to Phase II final report and disseminate software and publications.
Phase 3 (Task 3.3.3): Milestones 11-12: Apply lessons learned from Phase II and extend design/approach to support 70 dimensions, 5 LECs and 500 conditional evidences. Milestones 13-14: Demonstrate technology capable of supporting 100 dimensions, 6 LECs and 1000 conditional evidences. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.3.4, 2.3.4, 3.3.4 - System integration, deployment, and testing: (USC/ISI)
Objective: Develop and implement integration, testing and deployment plan supporting TA3 for all three phases.
Approach: Develop an internal TA3 integration and testing plan (unit tests, etc.) and, in close collaboration with TA1 and TA2 performers on project, develop an overall TA1-TA3 integration and testing plan. Working with TA4 performers, extend and execute plan for TA4 platform (when available).
Phase 1 (1.2.3): Milestones 1-2: Develop initial integration and testing plan and implement on unit testing. Milestones 3-5: Oversee integration and testing of TA1-TA3 components for system capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Complete integration of technology into TA4 testbeds, contribute to Phase II final report and disseminate software and publications.
Phase 2 (2.2.3): Milestones 7-8: Apply lessons learned from Phase II and extend existing integration and testing plan to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Complete integration of technology into TA4 platforms. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (3.2.3): Milestones 11-12: Apply lessons learned from Phase II and extend existing integration and testing plan to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.
Tasks 1.4.1, 2.4.1, 3.4.1 – Project Management: (USC/ISI)
Objective: Provide overall project management for Phase 1. Assist in system design, integration and testing. Interface with TA4 performers to ensure collaboration
Approach: Establish weekly status meetings among team members, collaboration platform (e.g., Dropbox), provide technical assistance to integration efforts, resolve programmatic issues, develop monthly, quarterly and final reports. Schedule and participate in technical exchange meetings, assist in developing component interfaces, establish test procedures, prototype testing. Meet with TA4 performers to discuss test scenarios, platform integration and performance issues
Phase 1 (1.2.3): Milestones 1-2: Establish meeting schedules and collaboration platforms. Assist teams in developing design and undertaking unit testing. Milestones 3-5: Assist integration and testing of TA1-TA3 components for system capable of supporting 20 dimensions, 2 LECs and 50% or less monitoring overhead. Milestone 6: Assist integration of technology into TA4 testbeds, contribute to Phase II final report (C) and disseminate software and publications.
Phase 2 (2.2.3): Milestones 7-8: Apply lessons learned from Phase II and extend existing integration and testing plan to support 30 dimensions, 3 LECs and 40% monitoring overhead. Milestones 9-10: Support demonstration of technology capable of supporting 40 dimensions, 4 LECs and 30% monitoring overhead. Complete integration of technology into TA4 platforms. Contribute to Phase II final report and disseminate software and publications.
Phase 3 (3.2.3): Milestones 11-12: Apply lessons learned from Phase II and extend existing integration and testing plan to support 70 dimensions, 5 LECs and 20% monitoring overhead. Milestones 13-14: Support demonstration of technology capable of supporting 100 dimensions, 6 LECs and 10% monitoring overhead. Complete integration of technology into TA4 platform. Contribute to Phase III final report and disseminate software and publications.

9 Schedule and Milestones

The schedule is shown in Figure 8 and the milestones are listed in Table 14.

Table 14: The project has the following fourteen (14) milestones

Mile-stones	Month	Description	Deliverables
1	2	Initial Design completed. Design includes finalized research plans, identification of internal TA milestones, initial interfaces between the three TAs, planned interface with the TA4 platforms.	
2	3	Individual components developed and tested. TA1, TA2 and TA3 Interface design completed	Quarterly Report
3	6	Initial working system for Design Time (i.e., TA1 – TA3 interaction). Continued development of TA2. Supports includes one LEC. First P/I meeting. Review TA4 scenarios.	Quarterly Report, slide presentation
4	12	Working system for both Design Time and Operation Time (i.e., TA1, TA2 and TA3 interactions), supports 10 dimensions and one LEC. Second P/I meeting. Initial discussions with TA4 teams on interfaces	Quarterly Report, slide presentation
5	17	Working system that supports 20 dimensions and 2 LECs with no more that 50% monitoring overhead, 10 conditional evidence monitors and 0.1x reduced trails to assurance. Start integration effort into both TA4 platforms	Working system (software) available for integration into TA4 platforms. Monthly perfomance and financial reports
6	18	Phase I demonstration on both TA4 platforms	Phase I report, quarterly reports
7	19	Design review based on Phase I demo (lessons learned).	
8	25.5	Prototype system capable of supporting 30 dimensions, 3 LECs, with no more than 40% monitoring overhead, 50 conditional evidence and 0.05x reduced trails to assurance. Third P/I meeting	Quarterly report
9	32	Working system that supports up to 40 dimensions, 4 LECs, with no more than 30% monitoring overhead, 100 conditional evidence monitors and 0.01x reduced trails to assurance. Begin Integration into both TA4 platforms	Working system (software) available for integration into TA4 platforms. Monthly perfomance and financial reports
10	33	Phase II demonstration on both TA4 platforms	Phase II report, quarterly reports
11	34	Design review based on Phase II demo (lessons learned)	
12	40.5	Refined system to support 70 dimensions, 5 LECs, 500 conditional evidences and 20% monitoring overhead – Forth P/I meeting	Quarterly report
13	47	Working system that supports 100 dimensions, 6 LECs, 1000 conditional evidences, .001x reduction in assurance trials and 10% monitoring overhead	Working system (software) available for integration into TA4 platforms. Monthly perfomance and financial reports
14	48	Phase III demonstration on both TA4 platforms. Phase III report, final project reporet.	Phase III report, quarterly reports, Final project report

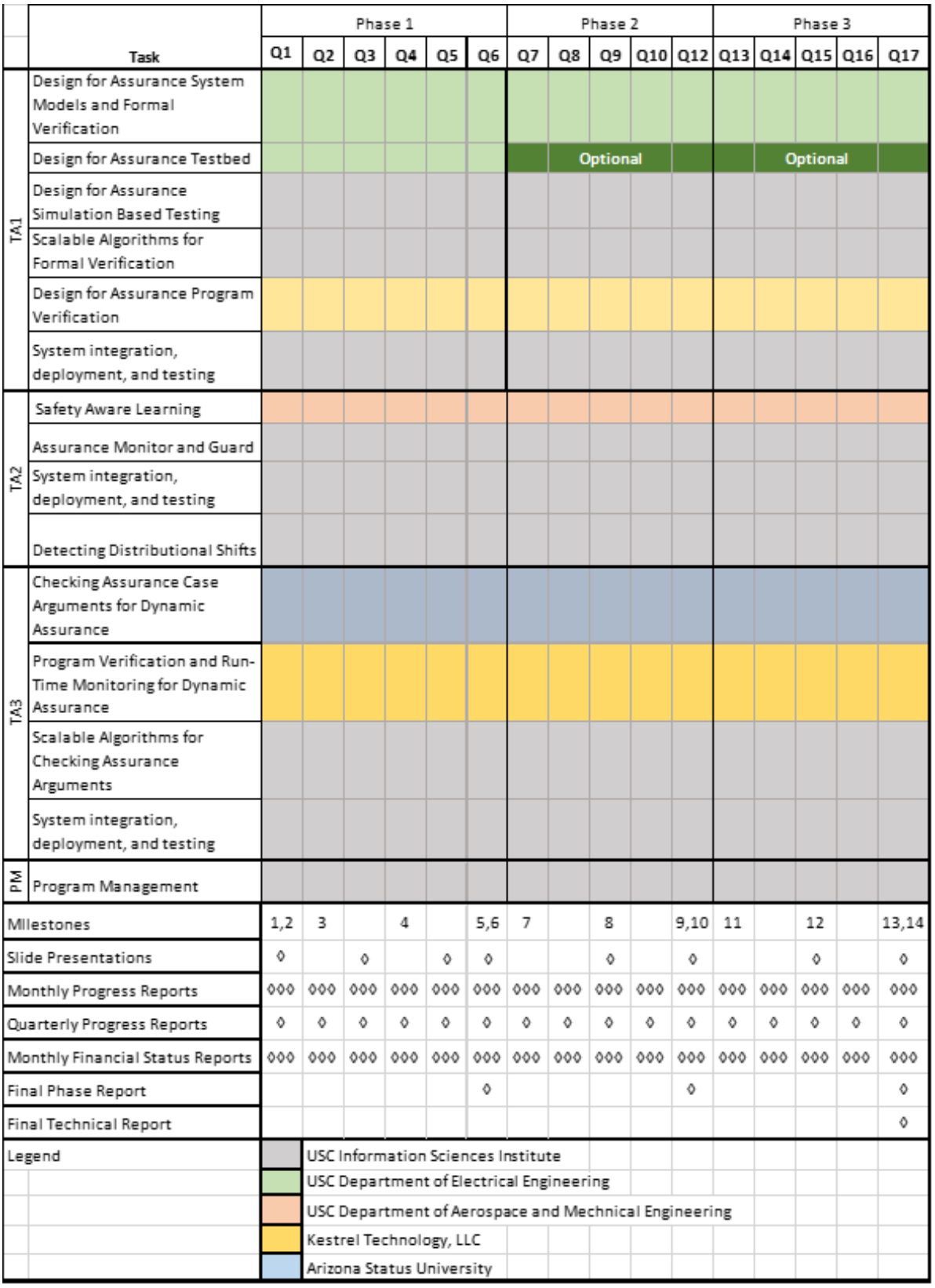

Figure 8: Project schedule along with a summary of milestones. The legend maps task color to organization primary responsible for the task.

Appendix A: Team Members and Other Information

Team Member Identification:

Name	Role	Organization	Non-US Org?	Non-US Ind?	FFRDC or Gov
Craig A. Knoblock	Prime	USC	N	N	N
Michael Orosz	Prime	USC	N	N	N
Satish Thittamaranahalli	Prime	USC	N	Y	N
Ryan Goodfellow	Prime	USC	N	N	N
Anoop Kumar	Prime	USC	N	N	N
Satyandra Gupta	Prime	USC	N	N	N
Pierluigi Nuzzo	Prime	USC	N	Y	N
Petros Ioannou	Prime	USC	N	N	N
Chitta Baral	Subcontractor	ASU	N	N	N
Matt Barry	Subcontractor	Kestrel Technology	N	N	N
Douglas Smith	Subcontractor	Kestrel Technology	N	N	N
Henny Sipma	Subcontractor	Kestrel Technology	N	Y	N

Government or FFRDC Team Member Proof of Eligibility to Propose: NONE

Government or FFRDC Team Member Statement of Unique Capability: NONE

Organizational Conflict of Interest Affirmations and Disclosure: NONE

Intellectual Property (IP):

COMMERCIAL ITEMS			
Technical Data, Computer Software To be Furnished With Restrictions	**Basis for Assertion**	**Asserted Rights Category**	**Name of Person Asserting Restrictions**
KT Advance	Developed with mixed funding.	Restricted	David Kulich, Contracts Manager, Kestrel technology, LLC.

Human Subjects Research (HSR): NONE

Animal Use: NONE

Representations Regarding Unpaid Delinquent Tax Liability or a Felony Conviction under Any Federal Law:

1. The proposer is not a corporation that has any unpaid Federal tax liability that has been assessed, for which all judicial and administrative remedies have been exhausted or have lapsed, and that is not being paid in a timely manner pursuant to an agreement with the authority responsible for collecting the tax liability,

2. The proposer is not a corporation that was convicted of a felony criminal violation under a Federal law within the preceding 24 months.

Cost Accounting Standards (CAS) Notices and Certification: NONE

Appendix B: Bibliography and Technical Papers

References

[1] Dymola - Multi-Engineering Modeling and Simulation, 2017.

[2] MapleSim - High Performance Physical Modeling and Simulation - Technical Computing Software, 2017.

[3] OpenModelica, 2017.

[4] Simscape - Physical Systems Simulation - MATLAB & Simulink, 2017.

[5] Wolfram SystemModeler: Modeling, Simulation & Analysis, 2017.

[6] AKAMETALU, A. K., FISAC, J. F., GILLULA, J. H., KAYNAMA, S., ZEILINGER, M. N., AND TOMLIN, C. J. Reachability-based safe learning with gaussian processes. In *53rd IEEE Conference on Decision and Control* (Dec 2014), pp. 1424–1431.

[7] ALUR, R., DANG, T., AND IVANČIĆ, F. Counterexample-guided predicate abstraction of hybrid systems. *Theoretical Computer Science 354*, 2 (2006), 250–271.

[8] ALUR, R., HENZINGER, T., LAFFERRIERE, G., AND PAPPAS, G. Discrete abstractions of hybrid systems. *Proc. IEEE 88*, 7 (July 2000), 971–984.

[9] AMINIKHANGHAHI, S., AND COOK, D. J. A survey of methods for time series change point detection. *Knowledge and Information Systems* (2016), 1–29.

[10] ATHAKRAVI, D., ALRAJEH, D., BRODA, K., RUSSO, A., AND SATOH, K. Inductive learning using constraint-driven bias. In *Inductive Logic Programming*. Springer, Cham, 2015, pp. 16–32.

[11] ATHAKRAVI, D., CORAPI, D., BRODA, K., AND RUSSO, A. Learning through hypothesis refinement using answer set programming. In *International Conference on Inductive Logic Programming* (2013), Springer, pp. 31–46.

[12] BARAL, C., AND HUNSAKER, M. Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning. In *IJCAI* (2007), pp. 243–249.

[13] BARRETT, C., SEBASTIANI, R., SESHIA, S. A., AND TINELLI, C. Satisfiability modulo theories. In *Handbook of Satisfiability*, A. Biere, H. van Maaren, and T. Walsh, Eds., vol. 4. IOS Press, 2009, ch. 8.

[14] BARRY, M. R., RUSHTON, J. N., AND BUSH, E. W. Final report, certware absa project, argument-based safety assurance program. Tech. rep., NASA Langley Research Center, 2015.

[15] BASSEVILLE, M., NIKIFOROV, I. V., ET AL. *Detection of abrupt changes: theory and application*, vol. 104. Prentice Hall Englewood Cliffs, 1993.

[16] BENZEL, T. The science of cyber security experimentation. In *Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC '11* (New York, New York, USA, 12 2011), ACM Press, p. 137.

[17] BERGER, T., ILCHMANN, A., THOMAS HOTZ, J.-P., AND MEHRMANN, V. On Differential-Algebraic Control Systems.

[18] BERGER, T., AND REIS, T. Controllability of Linear Differential-Algebraic Systems—A Survey. In *Surveys in Differential-Algebraic Equations I*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 1–61.

[19] BERKENKAMP, F., AND SCHOELLIG, A. P. Safe and robust learning control with gaussian processes. In *2015 European Control Conference (ECC)* (July 2015), pp. 2496–2501.

[20] BERKENKAMP, F., SCHOELLIG, A. P., AND KRAUSE, A. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 491–496.

[21] BERTASKA, I. R., SHAH, B., VON ELLENRIEDER, K., ŠVEC, P., KLINGER, W., SINISTERRA, A. J., DHANAK, M., AND GUPTA, S. K. Experimental evaluation of automatically-generated behaviors for usv operations. *Ocean Engineering 106*, Supplement C (2015), 496 – 514.

[22] BIFET, A., AND GAVALDA, R. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (2007), SIAM, pp. 443–448.

[23] BLUM, A. L., AND LANGLEY, P. Selection of relevant features and examples in machine learning. *Artificial intelligence 97*, 1 (1997), 245–271.

[24] BRODSKY, E., AND DARKHOVSKY, B. S. *Nonparametric methods in change point problems*, vol. 243. Springer Science & Business Media, 2013.

[25] CHAN, K.-P., AND FU, A. W.-C. Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on* (1999), IEEE, pp. 126–133.

[26] CHLEBÍK, M., AND CHLEBÍKOVÁ, J. Crown reductions for the minimum weighted vertex cover problem. *Discrete Applied Mathematics 156*, 3 (2008), 292–312.

[27] CHUTINAN, A., AND KROGH, B. Verification of infinite-state dynamic systems using approximate quotient transition systems. *IEEE Trans. Automatic Control 46*, 9 (Sep 2001), 1401–1410.

[28] CIMATTI, A., AND TONETTA, S. Contracts-refinement proof system for component-based embedded systems. *Science of Computer Programming 97, Part 3* (2015), 333 – 348.

[29] COHEN, L., URAS, T., JAHANGIRI, S., ARUNASALAM, A., KOENIG, S., AND KUMAR, T. K. S. The fastmap algorithm for shortest path computations. *Proceedings of the Fifteenth International Symposium on Artificial Intelligence and Mathematics (ISAIM-2018; in review); arXiv preprint arXiv:1706.02792* (2018).

[30] COMMANDANT, U. International regulations for prevention of collisions at sea, 1972 (72 colregs). *US Department of Transportation, US Coast Guard, COMMANDANT INSTRUCTION M 16672* (1999).

[31] COUSOT, P., AND COUSOT, R. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL* (1977), ACM, pp. 238–252.

[32] COUSOT, P., AND HALBWACHS, N. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978* (1978), pp. 84–96.

[33] CUTERI, B., AND RICCA, F. A compiler for stratified logic programs: preliminary report. In *Proceedings of the First International Workshop on Practical Aspects of Answer Set Programming* (Espoo, Finland, 2017).

[34] DANG, T., MALER, O., AND TESTYLIER, R. Accurate hybridization of nonlinear systems. In *Proc. Hybrid Systems: Computation and Control* (New York, NY, USA, 2010), HSCC '10, ACM, pp. 11–20.

[35] DERESZYNSKI, E. W., AND DIETTERICH, T. G. Spatiotemporal models for data-anomaly detection in dynamic environmental monitoring campaigns. *ACM Transactions on Sensor Networks (TOSN) 8*, 1 (2011), 3.

[36] DERESZYNSKI, E. W., AND DIETTERICH, T. G. Probabilistic models for anomaly detection in remote sensor data streams. *arXiv preprint arXiv:1206.5250* (2012).

[37] DIETTERICH, T. G., DERESZYNSKI, E. W., HUTCHINSON, R. A., AND SHELDON, D. R. Machine learning for computational sustainability. In *IGCC* (2012), p. 1.

[38] FINN, J., NUZZO, P., AND SANGIOVANNI-VINCENTELLI, A. A mixed discrete-continuous optimization scheme for cyber-physical system architecture exploration. In *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design* (Nov. 2015).

[39] FREHSE, G. PHAVer: algorithmic verification of hybrid systems past HyTech. *Int. J. Software Tools for Technology Transfer 10* (2008), 263–279.

[40] GACEK, A., BACKES, J., COFER, D. D., SLIND, K., AND WHALEN, M. Resolute: An assurance case language for architecture models. *CoRR abs/1409.4629* (2014).

[41] GAMA, J. *Knowledge discovery from data streams*. CRC Press, 2010.

[42] GEBSER, M., KAMINSKI, R., KAUFMANN, B., AND SCHAUB, T. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[43] GEBSER, M., KAMINSKI, R., KAUFMANN, B., AND SCHAUB, T. Clingo = asp + control: Preliminary report. *CoRR abs/1405.3694* (2014).

[44] GEBSER, M., MARATEA, M., AND RICCA, F. The sixth answer set programming competition. *J. Artif. Intell. Res. 60* (2017), 41–95.

[45] GILLULA, J. H., AND TOMLIN, C. J. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *2012 IEEE International Conference on Robotics and Automation* (May 2012), pp. 2723–2730.

[46] GILLULAY, J. H., AND TOMLIN, C. J. Guaranteed safe online learning of a bounded system. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Sept 2011), pp. 2979–2984.

[47] GOODFELLOW, R., AND KLINE, E. Cypress: A Testbed for Research in Networked Cyber-Physical Systems. In *10th International Conference on Testbeds and Research Infrastructures for the Developemnt of Networks and Communicites (TRIDENTCOM)* (Vancouver, Canada, 2015).

[48] GUPTA, D., VISHWANATH, K. V., MCNETT, M., VAHDAT, A., YOCUM, K., SNOEREN, A., AND VOELKER, G. M. DieCast. *ACM Transactions on Computer Systems 29*, 2 (5 2011), 1–48.

[49] GUSTAFSSON, F., AND GUSTAFSSON, F. *Adaptive filtering and change detection*, vol. 1. Citeseer, 2000.

[50] HALPERN, J. Y. *Reasoning about uncertainty*. MIT press, 2003.

[51] HAVELUND, K., AND ROSU, G. Efficient monitoring of safety properties. *Int. J. Softw. Tools Technol. Transf. 6*, 2 (Aug. 2004), 158–173.

[52] HENZINGER, T., HO, P.-H., AND WONG-TOI, H. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans. Automatic Control 43*, 4 (Apr. 1998), 540–554.

[53] HOLLOWAY, C. M. Safety case notations: Alternatives for the non-graphically inclined? In *3rd IET International Conference on System Safety* (11 2008), pp. 1 – 6.

[54] IANNOPOLLO, A., NUZZO, P., TRIPAKIS, S., AND SANGIOVANNI-VINCENTELLI, A. L. Library-based scalable refinement checking for contract-based design. In *Proc. Design, Automation and Test in Europe* (Mar. 2014).

[55] ILCHMANN, A., AND MEHRMANN, V. A Behavioral Approach to Time-Varying Linear Systems. Part 1: General Theory. *SIAM Journal on Control and Optimization 44*, 5 (1 2005), 1725–1747.

[56] IOANNOU, P., AND XU, Z. Throttle and brake control systems for automatic vehicle following. *Journal of Intelligent Transportation Systems 1*, 4 (1994), 345–377.

[57] IOANNOU, P. A., AND SUN, J. *Robust adaptive control*, vol. 1. PTR Prentice-Hall Upper Saddle River, NJ, 1996.

[58] JIA, Y., SHELHAMER, E., DONAHUE, J., KARAYEV, S., LONG, J., GIRSHICK, R., GUADARRAMA, S., AND DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (2014), ACM, pp. 675–678.

[59] KALRA, N., AND PADDOCK, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice 94* (2016), 182–193.

[60] KANARIS, A., KOSMATOPOULOS, E. B., AND LOANNOU, P. A. Strategies and spacing requirements for lane changing and merging in automated highway systems. *IEEE transactions on vehicular technology 50*, 6 (2001), 1568–1581.

[61] KARR, M. Affine relationships among variables of a program. *Acta Inf. 6* (1976), 133–151.

[62] KATZOURIS, N., ARTIKIS, A., AND PALIOURAS, G. Incremental learning of event definitions with inductive logic programming. *Machine Learning 100*, 2-3 (2015), 555–585.

[63] KAZMI, M., SCHÜLLER, P., AND SAYGIN, Y. Improving scalability of inductive logic programming via pruning and best-effort optimisation. *Expert Systems with Applications* (2017).

[64] KELLY, T. A systematic approach to safety case management. In *SAE Technical Paper* (03 2004), SAE International.

[65] KEUTZER, K., MALIK, S., NEWTON, R., RABAEY, J., AND SANGIOVANNI VINCENTELLI, A. System Level Design: Orthogonalization of Concerns and Platform-Based Design. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. 19*, 12 (2000), 1523–1543.

[66] KOLLER, D., AND FRIEDMAN, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[67] KUMAR, T. K. S. A framework for hybrid tractability results in Boolean weighted constraint satisfaction problems. In *the International Conference on Principles and Practice of Constraint Programming* (2008), pp. 282–297.

[68] KUMAR, T. K. S. Lifting techniques for weighted constraint satisfaction problems. In *the International Symposium on Artificial Intelligence and Mathematics* (2008).

[69] KUMAR, T. K. S. Kernelization, generation of bounds, and the scope of incremental computation for weighted constraint satisfaction problems. In *the International Symposium on Artificial Intelligence and Mathematics* (2016).

[70] KURD, Z., KELLY, T., AND AUSTIN, J. Safety criteria and safety lifecycle for artificial neural networks.

[71] LAMOUR, R., MÄRZ, R., AND TISCHENDORF, C. *Differential-Algebraic Equations: A Projector Based Analysis*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[72] LAW, M., RUSSO, A., AND BRODA, K. Inductive learning of answer set programs. In *European Workshop on Logics in Artificial Intelligence* (2014), Springer, Cham, pp. 311–325.

[73] LEE, H. W., THUENTE, D., AND SICHITIU, M. L. Integrated simulation and emulation using adaptive time dilation. In *Proceedings of the 2nd ACM SIGSIM/PADS conference on Principles of advanced discrete simulation - SIGSIM-PADS '14* (New York, New York, USA, 5 2014), ACM Press, pp. 167–178.

[74] LI, J., NUZZO, P., SANGIOVANNI-VINCENTELLI, A., XI, Y., AND LI, D. Stochastic assume-guarantee contracts for cyber-physical system design under probabilistic requirements. *arXiv preprint arXiv:1705.09316* (2017).

[75] MALER, O., AND NICKOVIC, D. Monitoring temporal properties of continuous signals. In *Formal Modeling and Analysis of Timed Systems* (2004), pp. 152–166.

[76] MALIK, S., AND ZHANG, L. Boolean satisfiability from theoretical hardness to practical success. *Communications of the ACM 52*, 8 (2009), 76–82.

[77] MANNUCCI, T., VAN KAMPEN, E. J., DE VISSER, C., AND CHU, Q. Safe exploration algorithms for reinforcement learning controllers. *IEEE Transactions on Neural Networks and Learning Systems PP*, 99 (2017), 1–13.

[78] MARTÍNEZ, D., ALENYÀ, G., AND TORRAS, C. Safe robot execution in model-based reinforcement learning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sept 2015), pp. 6422–6427.

[79] MCKINNON, C. D., AND SCHOELLIG, A. P. Learning multimodal models for robot dynamics online with a mixture of gaussian process experts. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (May 2017), pp. 322–328.

[80] MICHALSKI, R. S., CARBONELL, J. G., AND MITCHELL, T. M. *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[81] MUGGLETON, S. Inductive logic programming. *New generation computing 8*, 4 (1991), 295–318.

[82] NAEEM, W., AND IRWIN, G. W. An automatic collision avoidance strategy for unmanned surface vehicles. In *Life System Modeling and Intelligent Computing*. Springer, 2010, pp. 184–191.

[83] NAIR, S., DE LA VARA, J. L., SABETZADEH, M., AND FALESSI, D. Management of evidence for compliance with safety standards: a survey on the state of practice. Tech. Rep. 2012-27, 2013.

[84] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION. *NASA Procedural Requirement 7150.2B, Software Engineering Requirements*, 2014.

[85] NUZZO, P., LORA, M., FELDMAN, Y., AND SANGIOVANNI-VINCENTELLI, A. CHASE: Contract-Based Requirement Engineering for Cyber-Physical System Design. Submitted.

[86] NUZZO, P., SANGIOVANNI-VINCENTELLI, A., BRESOLIN, D., GERETTI, L., AND VILLA, T. A platform-based design methodology with contracts and related tools for the design of cyber-physical systems. *Proc. IEEE 103*, 11 (Nov. 2015).

[87] NUZZO, P., XU, H., OZAY, N., FINN, J., SANGIOVANNI-VINCENTELLI, A., MURRAY, R., DONZÉ, A., AND SESHIA, S. A contract-based methodology for aircraft electric power system design. *IEEE Access 2* (2014), 1–25.

[88] PASSERONE, R., BURCH, J. R., AND SANGIOVANNI-VINCENTELLI, A. L. Refinement preserving approximations for the design and verification of heterogeneous systems. *Formal Methods in System Design 31*, 1 (Aug 2007), 1–33.

[89] PNUELI, A. The temporal logic of programs. In *Annual Symp. on Foundations of Computer Science* (Nov. 1977), pp. 46–57.

[90] POLO, F. J. G., AND REBOLLO, F. F. Safe reinforcement learning in high-risk tasks through policy improvement. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (April 2011), pp. 76–83.

[91] QUIONERO-CANDELA, J., SUGIYAMA, M., SCHWAIGHOFER, A., AND LAWRENCE, N. D. *Dataset shift in machine learning*. The MIT Press, 2009.

[92] RABOIN, E., ŠVEC, P., NAU, D. S., AND GUPTA, S. K. Model-predictive asset guarding by team of autonomous surface vehicles in environment with civilian boats. *Autonomous Robots 38*, 3 (Mar 2015), 261–282.

[93] RAY, O. Nonmonotonic abductive inductive learning. *Journal of Applied Logic 7*, 3 (2009), 329–340.

[94] ROSU, G., CHEN, F., AND BALL, T. Runtime verification. Springer-Verlag, 2008, ch. Synthesizing Monitors for Safety Properties: This Time with Calls and Returns, pp. 51–68.

[95] ROY, N., NEWMAN, P., AND SRINIVASA, S. *Reducing Conservativeness in Safety Guarantees by Learning Disturbances Online: Iterated Guaranteed Safe Online Learning*. MIT Press, 2013, pp. 504–.

[96] RUSHBY, J. An evidential tool bus. In *In Proceedings of ICFEM 2005* (2005).

[97] RUSHBY, J. Formalism in safety cases. In *Making Systems Safer: Proceedings of the Eighteenth Safety-Critical Systems Symposium* (2010), C. Dale and T. Anderson, Eds., pp. 3 – 17.

[98] SANGIOVANNI-VINCENTELLI, A., DAMM, W., AND PASSERONE, R. Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems. *European Journal of Control 18-3*, 3 (2012), 217–238.

[99] SANKARANARAYANAN, S., COLÓN, M., SIPMA, H. B., AND MANNA, Z. Efficient strongly relational polyhedral analysis. In *Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Charleston, SC, USA, January 8-10, 2006, Proceedings* (2006), pp. 111–125.

[100] SANKARANARAYANAN, S., SIPMA, H., AND MANNA, Z. Non-linear loop invariant generation using gröbner bases. In *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004* (2004), pp. 318–329.

[101] SANKARANARAYANAN, S., SIPMA, H. B., AND MANNA, Z. Scalable analysis of linear systems using mathematical programming. In *Verification, Model Checking, and Abstract Interpretation, 6th International Conference, VMCAI 2005, Paris, France, January 17-19, 2005, Proceedings* (2005), pp. 25–41.

[102] SHAH, B. C., ŠVEC, P., BERTASKA, I. R., SINISTERRA, A. J., KLINGER, W., VON ELLENRIEDER, K., DHANAK, M., AND GUPTA, S. K. Resolution-adaptive risk-aware trajectory planning for surface vehicles operating in congested civilian traffic. *Autonomous Robots 40*, 7 (Oct 2016), 1139–1163.

[103] SHOUKRY, Y., NUZZO, P., SANGIOVANNI-VINCENTELLI, A. L., SESHIA, S. A., PAPPAS, G. J., AND TABUADA, P. SMC: Satisfiability modulo convex optimization. In *Proc. Hybrid Systems: Computation and Control* (2017), pp. 19–28.

[104] SMITH, D. R. Transformation automata and policy enforcement. In *Proceedings of the Sixth Workshop on Foundations of Aspect-Oriented Languages (FOAL 2007)* (2007), ACM Press/Digital Library.

[105] SMITH, D. R. Aspects as invariants. In *Automatic Program Development: a Tribute to Robert Paige* (2008), O. Danvy, F. Henglein, H. Mairson, and A. Pettorosi, Eds., Springer-Verlag.

[106] SMITH, D. R., AND WESTFOLD, S. Toward the Synthesis of Constraint Solvers. Tech. rep., Kestrel Institute, 2013. http://www.kestrel.edu/home/people/smith/pub/CW-report.pdf.

[107] SMITH, D. R., AND WESTFOLD, S. J. Synthesis of Propositional Satisfiability Solvers. Tech. rep., Kestrel Institute, 3260 Hillview Ave., Palo Alto, CA 94304, April 2008. http://www.kestrel.edu/home/people/smith/pub/SAT-Tech-Report.pdf.

[108] ŠVEC, P., THAKUR, A., RABOIN, E., SHAH, B. C., AND GUPTA, S. K. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots 36* (2014), 383–405.

[109] SYRJÄNEN, T., AND NIEMELÄ, I. *The Smodels System*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 434–438.

[110] SZEGEDY, C., ZAREMBA, W., SUTSKEVER, I., BRUNA, J., ERHAN, D., GOODFELLOW, I., AND FERGUS, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[111] THAKUR, A., ŠVEC, P., AND GUPTA, S. Gpu based generation of state transition models using simulations for unmanned surface vehicle trajectory planning. *Robotics and Autonomous Systems* (2012).

[112] ŠVEC, P., AND GUPTA, S. K. Automated synthesis of action selection policies for unmanned vehicles operating in adverse environments. *Autonomous Robots 32*, 2 (2012), 149–164.

[113] WANG, H., FAN, W., YU, P. S., AND HAN, J. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2003), KDD '03, ACM, pp. 226–235.

[114] WITTEN, I. H., FRANK, E., HALL, M. A., AND PAL, C. J. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[115] XU, H., KUMAR, T. K. S., AND KOENIG, S. The Nemhauser-Trotter reduction and lifted message passing for the weighted CSP. In *Proceedings of the 14th International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming (CPAIOR)* (2017), pp. 387–402.

[116] YEDIDIA, J. S., FREEMAN, W. T., AND WEISS, Y. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium 8* (2003), 239–269.

Technical Papers We have included below 3 technical papers and reports by team members that relevant to the proposed effort.