Research Statement: Engineering Trustworthy Autonomous Systems

Ivan Ruchkin
https://www.seas.upenn.edu/~iruchkin

Our society is undergoing a profound change: connected computing systems are being deployed ubiquitously in our physical surroundings. A rapidly developing class of such systems, termed *Cyber-Physical Systems* (CPS, sometimes referred to as Robotic Systems), is illustrated in Figure 1. Enabled by advances in hardware, sensing, and artificial intelligence, these systems promise enormous economic returns with autonomous operation across a multitude of industries. The widespread use of these systems is critically dependent on their *safety* and *trustworthiness,* as well as their *cost-effective engineering* on a large scale. This engineering needs to overcome *three complexity factors:*

- **Physicality:** continual close interaction with the physical world
- **Autonomy:** purposeful behavior without direct human supervision
- **Heterogeneity:** mixture of component types and engineering techniques

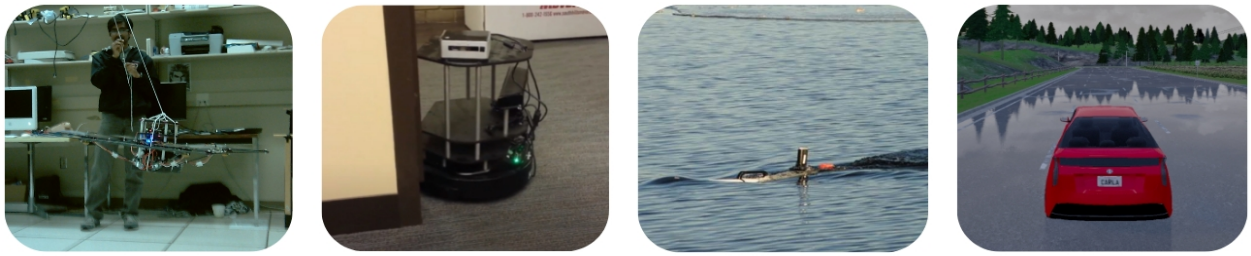

Figure 1. Example systems targeted in my research: (a) quadrotor, (b) service robot, (c) underwater vehicle, and (d) autonomous car.

Today's major challenge is that it is *difficult* and *costly* to build **effective**, **safe**, and **trustworthy CPS**. They execute in unstructured and unpredictable *physical environments*, beyond what software-controlled systems have navigated in the past. To operate *autonomously* in these environments, CPS rely on data-driven learning components, particularly Deep Neural Networks (DNNs). Unfortunately, these components introduce unintuitive and non-robust behavior and, hence, threaten the safety and trustworthiness of the systems. Modeling and analysis – the traditional basis of trustworthiness – are impeded by the high-dimensional black-box nature of learning components. These circumstances call for a wide variety of *heterogeneous* models and techniques [5, 6, 15], often making these systems too complex for engineers to work with effectively and, thus, leading to higher costs and slower technological progress.

My research agenda is to develop **integrated high-assurance methods** for engineering modern CPS.

I pursue this agenda by building upon techniques and tools in **Formal Methods** (formal modeling, logic, and design/run-time verification), **Software/Systems Engineering** (design representations, domain-specific languages, and testing/simulation), and **Artificial Intelligence** (probabilistic and statistical reasoning and learning) – and combining them to provide **design automation** and **strong guarantees** for **large-scale high-complexity** systems.

My research contributions thus far can be organized in two thrusts:

1. **Integration of Modeling Methods for CPS:** formal specifications and semantic foundations for engineering CPS using multiple heterogeneous models and analyses.
2. **Evaluation of Trustworthiness for Learning-Enabled CPS:** quantification of design-time and run-time trustworthiness in CPS with learned components using partial and uncertain domain knowledge.

Impact. My work on integrating modeling methods has been widely recognized, receiving the ACM SIGSOFT Distinguished Paper Award, IEEE World Forum on the Internet of Things Best Paper Award, ACM Student Research Competition Gold Medal, and the SIGSOFT/SIGBED Frank Anger Memorial Award for crossover between embedded systems and software engineering. My research has been disseminated at top conferences and journals across several areas, including EMSOFT, FM, TAC, and TCAD, and applied to many domains: quadrotors [9, 19, 20], service robots [10, 11, 14, 18], autonomous cars [13, 21], cloud infrastructure [9, 12], battery design [20], underwater vehicles [1, 7, 8], and clinical care [3]. Furthermore, my research outcomes have been deployed on real platforms

funded by the Defense Advanced Research Projects Agency (DARPA): my model integration techniques connected Carnegie Mellon's models for adaptation in service robots [10] in the Building Resource-Adaptive Software Systems (BRASS) program, whereas my confidence composition framework served as an overarching methodology for the University of Pennsylvania's demonstrations on unmanned underwater vehicles in the Assured Autonomy program.

Thrust 1: Integration of Modeling Methods for CPS

Modeling methods form a rigorous foundation of engineering computing systems: how we think about and design systems is strongly influenced by our models. CPS require a variety of modeling methods so that each model is used in a suitable scenario: a differential equation for reachability analysis, a probabilistic automaton for model checking, and a signal-flow model for falsification. However, such heterogeneity brings about a fundamental challenge: how to **discover** and **prevent inconsistencies** between the models?

To address this challenge, my PhD research developed a framework for **integrated use** of diverse **modeling methods:**
(1A) Create **unifying abstractions** of system models and **verify** that they are **logically consistent**.
(1B) Automatically **analyze** and **improve** these representations while **preventing errors**.

(1A) Heterogeneous CPS models vary widely in their syntax and underlying formalism. It is often difficult to express what it means for two models to be consistent, let alone check it automatically. My key idea was to distill the essential information from models into two abstractions: the static, unchanging elements of models are represented with **views** [18] (i.e, annotated object models inspired by software architecture), whereas the dynamic, changing elements are represented with **behavior traces** [21]. These two abstractions can be connected with logical consistency properties specified in the **Integration Property Language** (IPL) [11], which combines first-order quantification over views and modal constraints on behaviors. IPL accommodates a variety of formalisms through its extensible sublanguages and model checkers. IPL formulas can be verified with a provably sound algorithm that combines Satisfiability Modulo Theories and behavioral model checking. This verification discovered multiple real-world **consistency errors**, some of them safety-critical, in the models of a quadrotor (Figure 1a) and a service robot (Figure 1b).

(1B) When originating in different technical domains, multi-model algorithms may produce incorrect results and introduce errors when applied to the same system without coordination. To execute correctly, such algorithms typically require human supervision: some apply only to specialized systems, others may encounter out-of-date or overwritten inputs, and yet others may change the system to violate the implicit assumptions of some other algorithms. To automate the execution of multi-model algorithm without errors, I developed a lightweight specification called **analysis contracts** [13, 16, 19, 20] for the inputs, outputs, assumptions, and guarantees of each model-based algorithm. The inputs and outputs are specified in terms of abstract model elements from the views described in (1A). The assumptions/guarantees are multi-model integration properties specified in IPL and checked before/after the algorithm execution. Based on the contracts, I ensure error-free execution by constructing a graph of algorithms, selecting an sequence of algorithms that satisfies the dependencies, and executing each algorithm while checking its assumptions and guarantees. This approach has successfully combined algorithms from **five previously unrelated domains**: real-time scheduling, taint analysis, model checking, battery scheduling, and thermal battery analysis.

This thrust laid the foundations [15, 17] for engineering CPS with multiple first-principles models and algorithms. However, with the advent of learning components, first-principles models tend to be highly uncertain, inaccurate, or unavailable. In Thrust 2, I apply my lessons from model integration to bridge the growing **gaps** between non-determinstic, probabilistic, and statistical models in presence of learning components.

Thrust 2: Evaluation of Trustworthiness for Learning-Enabled CPS

In this thrust, I focus on an important aspect of *trustworthiness:* a trustworthy autonomous system either performs its task successfully or fails clearly, safely, and predictably. By evaluating whether the system is trustworthy in this sense, we establish its operational bounds, satisfy legal requirements, and contribute to the trust of the users. Recently, the use of learning components has made the **trustworthiness** of CPS **difficult to evaluate** in two ways. First, end-to-end system analysis and verification do not extend naturally to components that introduce difficult-to-quantify uncertainty and lack low-dimensional descriptions. Second, due to their black-box nature, learning components limit engineers in using their intuition and domain knowledge to evaluate trustworthiness. Thus, for safety-critical learning-enabled CPS, it is crucial to define suitable measures of trustworthiness and ways to estimate them conservatively (i.e., without overestimating them) at both design time and run time.

As an early step in addressing the above challenges, my postdoctoral research investigates three questions:

(2A) How to evaluate the **trustworthiness** of **run-time monitors** at design time without testing data?
(2B) How to compute a **run-time measure** of **system trustworthiness** based on a verified model?
(2C) How to take advantage of less-than-certain **domain knowledge** to analyze trustworthiness?

When answering these questions, I represented uncertainty probabilistically with the notions of **detection rate** (precision, recall, specificity, etc.) and **confidence** (a calibrated probability estimate). To quantify trustworthiness and provide uncertainty-aware guarantees, I developed **compositional reasoning** that **integrates** exhaustive, probabilistic, and statistical techniques. This reasoning was informed by probabilistic domain knowledge in the form of **intuitive heuristics** and **declarative specifications**.

(2A) **Estimating detection rates without testing data.** A key measure of trustworthiness for discrete-output run-time monitors is their *detection rates*. However, these rates are usually unknown because the monitors use unpredictable outputs of learned perception and labeled testing data is not available. I investigated two ways of overcoming this challenge: exploiting the monitor's compositional structure and using domain-specific labeling heuristics. In the first case, I infer the detection rates of safety monitors specified in linear temporal logic (LTL) from the rates of their constituent, atomic detectors and conditional independence assumptions. To perform this inference, I developed an analytic **theory** of **detector composition** [8] that enhanced LTL with the semantics of stochastic three-value detectors and related the rates of composite and atomic detectors via logical, algebraic, and independence theories. Experiments on underwater sonar perception (Figure 1c) showed that this theory enables **accurate** and **data-efficient symbolic reasoning** about detection rates of LTL monitors. In the second case, we assume that a representative sample of the monitor's inputs is available, but the correct monitor outputs (i.e., the true labels) are not known. Instead of time-consuming manual labeling, we elicited domain-specific heuristics and used them as weak labeling functions in *data programming* to estimate labels with varied confidence. To account for the sampling and labeling uncertainties, we used high-confidence labels to infer the Probably Approximately Correct (**PAC**) **statistical bounds** for the true detection rate [3]. These bounds were shown to be **narrow** and contain the true rate **with high probability** in a case study of clinical alarm suppression. Looking forward, both estimation techniques are promising for quantifying broader notions of trustworthiness, such as probabilistic run-time confidence in Thrust (2B).

(2B) **Composition of confidence monitors for verification assumptions.** Exhaustive reachability verification of systems with neural networks promises strong guarantees of safety and performance, but it is computationally feasible only on simplified models. These models may turn out to be invalid at run time due to distribution shift, model inaccuracy, and unexpected noise – which can be detected with run-time monitors. To improve the trustworthiness of a system with neural networks, we aimed to **quantify** the extent to which the **verification guarantees** hold at run time. This quantification is challenging because it needs to take into account the verification results and multiple outputs of potentially inaccurate monitors. Our key insight was that *a verified system will uphold its guarantees as long as the assumptions behind its verification are satisfied*. Building on this insight, I led the development of a three-step **confidence composition framework** [1, 7]: (i) verify a safety property under explicit *logical assumptions*, (ii) build a calibrated *probabilistic confidence monitor* for each assumption, and (iii) *compose* the outputs of the monitors into a *single confidence measure.* The framework determines the sufficient conditions under which the composed confidence is **guaranteed** to be **calibrated** and **conservative** with respect to the safety chance, up to a bounded error. In our initial experiments on a mountain car and an underwater vehicle (Figure 1c) controlled by neural networks, the composed confidence **predicted safety violations** and **outperformed** the individual monitors.

(2C) **Leveraging domain knowledge in trustworthiness analysis.** Engineers often have substantial domain knowledge that could improve the analysis of trustworthiness. However, much of this knowledge is challenging to utilize because it is less-than-certain and does not match the typical inputs or constraints of the analysis. We investigated two probabilistic ways of encoding this knowledge: safety heuristics and temporal sampling scenarios. In the first case, the approximate and partial intuitions about behavior safety, such as "it is safer to drive slower" in the Carla simulator (Figure 1d), were represented as **monotonic safety** (MoS) **heuristics** [2]. We used MoS heuristics to remove the *safer* behaviors from the system's model and, hence, improve the scalability of probabilistic/statistical model checking for systems with DNN-based perception. When MoS heuristics are correct, this simplification is **provably conservative**. In our experiments, MoS led to **order-of-magnitude improvements** in the run times and data efficiency of model checking, as well as empirical conservatism even when the MoS heuristic is not perfectly accurate. In the second case, we helped engineers combine their heterogeneous knowledge about the desired temporal sampling scenarios in a **declarative domain-specific language** (DSL) [4]. This DSL captures constraints on marginal/joint/conditional probabilities, conditional independence, and temporal relations between categorical simulation variables. Our algebraic tool *PROSPECT* is guaranteed to find a unique distribution that satisfies those constraints, if it exists. Three case studies showed that our DSL specifications were **more succinct** than imperative probabilistic programs and **avoided** mis-specification **errors**.

Future Directions

Sharpening Run-Time Confidence with Hybrid Speculative Monitoring. To guard against the sim-to-real gap, run-time monitors flag unseen or unverified situations, such as out-of-distribution samples and violations of the verified model. Some of these situations are, in fact, *beneficial* to the system's goals; for instance, the classical mountain car may encounter a less steep hill (which violates the existing model) and succeed despite the monitor alarms. Thus, run-time monitoring of the sim-to-real gap can be **excessively conservative**: the run-time confidence is substantially lower than the actual mission success rate. To detect beneficial perturbations at run time, I propose **hybrid speculative confidence monitoring** that combines learned predictive models similar to how our hybrid planning framework combined fast and slow planners [9, 12]: the models that are expected to be less valid are down-weighted, whereas the ones consistent with the observations are up-weighted. When the valid models predict a beneficial outcome, increasing the confidence would make it more accurate on average. The main challenge in this direction is finding the constraints and augmentations that would provably prevent overconfidence.

Augmenting Exhaustive Verification with Targeted Testing. A significant amount of the monitoring conservatism mentioned above can stem from the uncertainty in exhaustive verification at design time. For example, run-time monitors have low confidence in the regions of state space where the verification has been inconclusive. My goal is to make run-time confidence more accurate with respect to the mission success rate by augmenting the verification with **targeted testing**. This testing would explore the model in the regions of potentially excessive conservatism, for instance, in the states where high uncertainty is due to the complexity in the DNN controller. This testing would clarify whether these regions are indeed unsafe or just poorly verifiable. To use these results at run time, I plan to construct their **statistical abstractions** and incorporate them into confidence reasoning. I hypothesize that relatively few tests can significantly improve monitoring calibration. In the longer term, the confidence can be further improved by treating run-time observations as unlabeled tests, connecting to our work on label-free evaluation [3].

Guaranteed Confidence-Based Recovery. A typical use of run-time monitoring is to trigger a recovery or adaptation guaranteed to preserve safety or minimal functionality. However, if the *very models* behind the system analysis are violated, how can the recovery provide any guarantees? This is an instance of **circular analytic dependency**, which I had encountered in my work on analysis contracts [16]. I hypothesize that this circularity can be broken in two steps: (i) each recovery gives guarantees on simplified models *only under* violations of certain assumptions, and (ii) the recovery is correctly chosen based on the accurate confidence in each assumption. The former step calls for abstract **models** with **plausible violations** of the original model's assumptions and, based on them, providing a recovery guarantee with bounded uncertainty. This analysis is expected to be robust to the invalidations of the original, more complex models. The latter step, in my opinion, can be accomplished by **bridging** the **detector** and **confidence composition frameworks** that I developed in Thrusts (2A) and (2B): detector composition would provide guarantees on the discrete recovery decisions by thresholding the composed confidence, which is subject to guarantees on continuous probabilities (e.g., the expected calibration error). Then we can compositionally assess each combination of assumptions and the expected outcome of each recovery, picking the one with the best expected outcome.

Specification and Synthesis of Modeling Assumptions for Automated Design. Today, modeling assumptions in CPS remain largely *informal* and are *manipulated manually* by engineers. Furthermore, in learning-enabled CPS, they may be *implicit*, hidden in the training data and process. This expensive and error-prone handling of assumptions is a major and daunting barrier to large-scale automation of iterative CPS design. In the short term, by building on my modeling experience, I plan to develop domain-specific languages for **specifying assumptions** of probabilistic and exhaustive models with **model-independent semantics**. These specifications would enable the synthesis of assumption checkers/monitors (similar to those for analysis contracts [20] and verification assumptions [1, 7]) and inform the system's design and analysis (similar to how monotonic safety improves model checking [2]). In the longer term, I plan to investigate how modeling assumptions can be **automatically synthesized, evaluated, used, refined,** and **retracted**. If developed, these capabilities would transform the modeling process beyond what is possible today.

Funding Opportunities

There is significant interest in engineering trustworthy autonomous systems in National Science Foundation (**NSF**), particularly in the Cyber-Physical Systems, Robust Intelligence, and National Robotics Initiative programs, National Aeronautics and Space Administration (**NASA**), particularly in the Ames Research Center and the Jet Propulsion Laboratory, and across the Department of Defense (**DoD**) agencies, including the Office of Naval Research (**ONR**), in particular the Science of Autonomy program, Defense Advanced Research Projects Agency (**DARPA**), in programs such as BRASS and Assured Autonomy, as well as the Air Force Research Lab (**AFRL**) and the Army Research Office (**ARO**). Applications to medical CPS can be funded through the National Institutes

of Health (**NIH**), particularly through programs related to Smart Health. The interdisciplinary nature of my research enables my participation in broad programs such as the Multidisciplinary University Research Initiatives (**MURI**) program and Collaborative Technology Alliances (**CTA**). Targeted applications can receive funding from industrial partners such as Microsoft, Intel, and ABB.

References

[1] **Ivan Ruchkin**, Matthew Cleaveland, Radoslav Ivanov, Pengyuan Lu, Taylor Carpenter, Oleg Sokolsky, Insup Lee. Confidence Composition for Monitors of Verification Assumptions. In Proc. of the International Conference on Cyber-Physical Systems (ICCPS), Milan, Italy, 2022.

[2] Matthew Cleaveland, **Ivan Ruchkin**, Oleg Sokolsky, Insup Lee. Monotonic Safety for Scalable and Data-Efficient Probabilistic Safety Analysis. In Proceedings of the International Conference on Cyber-Physical Systems (ICCPS), Milan, Italy, 2022.

[3] Sydney Pugh, **Ivan Ruchkin**, Christopher Bonafide, Sara DeMauro, Oleg Sokolsky, Insup lee, James Weimer. High-Confidence Data Programming for Evaluating Suppression of Physiological Alarms. In Proceedings of the Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), Washington, DC, 2021.

[4] Alan Ismaiel, **Ivan Ruchkin**, Jason Shu, Oleg Sokolsky, Insup Lee. Data Generation with PROSPECT: a Probability Specification Tool. In Proceedings of the Winter Simulation Conference (WinterSim), Phoenix, AZ, 2021. **Best Contributed Theoretical Paper**.

[5] Ankica Barišić, **Ivan Ruchkin**, Dušan Savić, Mustafa Abshir Mohamed, Rima Al-Ali, Letitia Li, Hana Mkaouar, Raheleh Eslampanah, Moharram Challenger, Dominique Blouin, Oksana Nikiforova, Antonio Cicchetti. Multi-paradigm modeling for cyber–physical systems: A systematic mapping review. In Journal of Systems and Software, vol. 183, 2021.

[6] Danny Weyns, Tomas Bures, Radu Calinescu, Barnaby Craggs, John Fitzgerald, David Garlan, Bashar Nuseibeh, Liliana Pasquale, Awais Rashid, **Ivan Ruchkin**, Bradley Schmerl. Six Software Engineering Principles for Smarter Cyber-Physical Systems. In Proceedings of the 8th Workshop on Self-Improving System Integration (SISSY) (in conjunction with ACSOS 2021). Washington, D.C., 2021.

[7] **Ivan Ruchkin**, Matthew Cleaveland, Oleg Sokolsky, and Insup Lee. Confidence Monitoring and Composition for Dynamic Assurance of Learning-Enabled Autonomous Systems. In Formal Methods in Outer Space: Essays Dedicated to Klaus Havelund on the Occasion of His 65th Birthday (in conjunction with ISoLA), 2021.

[8] **Ivan Ruchkin**, Oleg Sokolsky, James Weimer, Tushar Hedaoo, Insup Lee. Compositional Probabilistic Analysis of Temporal Properties Over Stochastic Detectors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol. 39, issue 11, 2020. In Proceedings of the International Conference on Embedded Software (EMSOFT).

[9] Ashutosh Pandey, **Ivan Ruchkin**, Bradley Schmerl, David Garlan. Hybrid Planning Using Learning and Model Checking for Autonomous Systems. In Proc. of the International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS), Washington, DC, USA, 2020.

[10] Jonathan Aldrich, David Garlan, Christian Kästner, Claire Le Goues, Anahita Mohseni-Kabir, **Ivan Ruchkin**, Selva Samuel, Bradley Schmerl, Christopher Steven Timperley, Manuela Veloso, Ian Voysey, Joydeep Biswas, Arjun Guha, Jarrett Holtz, Javier Cámara, Pooyan Jamshidi. Model-based adaptation for robotics software. In IEEE Software, 2019.

[11] **Ivan Ruchkin**, Joshua Sunshine, Grant Iraci, Bradley Schmerl, David Garlan, IPL: An Integration Property Language for Multi-Model Cyber-Physical Systems. In Proceedings of the 22nd International Symposium on Formal Methods (FM), Oxford, UK, 2018.

[12] Ashutosh Pandey, **Ivan Ruchkin**, Bradley Schmerl, Javier Camara. Towards a Formal Framework for Hybrid Planning in Self-Adaptation. In Proceedings of the 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) (in conjunction with ICSE). Buenos Aires, Argentina, 2017.

[13] **Ivan Ruchkin**, Ashwini Rao, Dionisio De Niz, Sagar Chaki, David Garlan. Eliminating Inter-Domain Vulnerabilities in Cyber-Physical Systems: An Analysis Contracts Approach. In the First ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC) (in conjunction with CCS). Denver, CO, 2015.

[14] **Ivan Ruchkin**, Selva Samuel, Bradley Schmerl, Amanda Rico, David Garlan. Challenges in Physical Modeling for Adaptation of Cyber-Physical Systems. In the First Workshop on Models at Runtime & Networked Control for Cyber Physical Systems (MARTCPS) (in conjunction with WF-IoT). Reston, VA, 2016. **The IEEE World Forum on the Internet of Things Best Paper Award**.

[15] **Ivan Ruchkin**. Integration Beyond Components and Models: Research Challenges and Directions. In Proceedings of the 3th Architecture Centric Virtual Integration Workshop (ACVI) (in conjunction with WICSA/CompArch). Venice, Italy, 2016.

[16] **Ivan Ruchkin**, Bradley Schmerl, David Garlan. Analytic Dependency Loops in Architectural Models of Cyber-Physical Systems. In the 8th International Workshop on Model-based Architecting of Cyber-Physical and Embedded Systems (ACES-MB) (in conjunction with MODELS). Ottawa, Canada, 2015.

[17] **Ivan Ruchkin**. Architectural and Analytic Integration of Cyber-Physical System Models. In the MODELS ACM Student Research Competition 2015. Ottawa, Canada. **ACM Student Research Competition Gold Medal.**

[18] **Ivan Ruchkin**, Bradley Schmerl, David Garlan. Architectural Abstractions for Hybrid Programs. In Proceedings of the 18th International Symposium on Component-Based Software Engineering (CBSE), Montreal, Canada, 2015. **ACM SIGSOFT Distinguished Paper Award**.

[19] **Ivan Ruchkin**, Dionisio De Niz, Sagar Chaki, David Garlan. ACTIVE: A Tool for Integrating Analysis Contracts. In Proceedings of the 5th Analytic Virtual Integration of Cyber-Physical Systems (AVICPS) (in conjunction with RTSS) Workshop, Rome, Italy, 2014.

[20] **Ivan Ruchkin**, Dionisio De Niz, Sagar Chaki, David Garlan. Contract-Based Integration of Cyber-Physical Analyses. In Proceedings of the 14th International Conference on Embedded Software (EMSOFT), New Delhi, India, 2014.

[21] Akshay Rajhans, Ajinkya Bhave, **Ivan Ruchkin**, Bruce Krogh, David Garlan, Andre Platzer, Bradley Schmerl. Supporting Heterogeneity in Cyber-Physical Systems Architectures. In IEEE Transactions on Automatic Control (TAC), Vol. 59, issue 12, 2014.