Model-Centric Cyber-Physical Computing

Robert Cartwright*, Kevin Kelly+, Farinaz Koushanfar+, Walid Taha*
*Department of Computer Science and
+Department of Electrical and Computer Engineering
Rice University, Houston, TX 77025

In *Cyber-physical* systems, digital computing components control or monitor physical components in real-time. Our goal is to develop novel methods to accelerate the development of highly reliable software and hardware solutions for cyber-physical systems. Our past research efforts have addressed numerous aspects of such systems, including robotics, nanoscale devices, sensor networks, and environmental monitoring and prediction infrastructures. Leadership in the development of cyber-physical systems not only helps maintain and improve US economic competitiveness, but contributes to protecting and extending human life.

The key lesson learned from our collective experience is that the structure of the physical model used to understand a particular problem has a dramatic effect on how solutions to this problem are developed.

We are particularly interested in bottleneck problems that arise in task-specific robotic and nanoscale devices. For a large class of these problems, developing the cyber-physical system critically depends on rapid prototyping and testing of new control algorithms. Whether the application is oil exploration or surgery, developing new, task-specific control algorithms is a costly and time-consuming task and is often the largest contributor to overall system cost.

The fundamental limitations of current technology for developing cyber-physical systems include:

Fidelity: Models for cyber-physical systems involve digital (discrete) and physical (continuous) elements, but there is a conspicuous absence of software tools and methods for establishing the correctness and validity of such cyber-physical models. For example, a wide variety of single molecule devices including switches, motors, shuttles, bearings, and elevators have recently been created. The synthesis of these molecules is extremely costly and time consuming. At the same time, modeling their behavior from first-principles (quantum mechanics) can be both computational expensive and potentially inaccurate. There is a pressing need in this domain for a way to rapidly develop ad hoc models and to establish their validity via experimentation and static checks (such as units analysis). Today, there is no mechanism for building executable models of the various chemical bonding structure into nano-scale electro-mechanical motion to accurately predict their individual behavior or their collective behavior in ensembles.

Reuse: Due to the enormous diversity of their applications, cyber-physical systems are fertile grounds for unique, creative innovations. Vivid examples can be seen in the task-specific robotics domain, where individual successful designs radically differ from one another. The specialized nature of such systems makes the developed canned solutions akin to numerical methods libraries impossible. Successful models are often ad hoc, and depend greatly on the details of the system being designed. As a result, the timely development of cyber-physical systems hinges on the ability to quickly simulate the physical nature of both the technology and the environment. At the same time, the development of cyber-physical systems spawns significant innovations in software and hardware design. Unfortunately, existing methods and tools for building hardware and software solutions do not provide the technology for expressing these innovations in a form than can be easily tailored (or specialized) to other cyber-physical problems.

Cost: Because many critical cyber-physical systems cannot benefit from the economics of mass production, the cost of human labor in design and validation can be extremely high. We need to develop tools and methods that allow new models to be prototyped quickly and efficiently, leveraging when possible similar features from other models.

The most important research challenges are:

Modeling languages for (hybrid) cyber-physical systems: While the difficulty in modeling cyber-physical systems comes from the diversity of these systems, the most promising approach to mitigating this problem is to develop very expressive, general modeling languages. For example, and as noted above, modeling molecular systems from the quantum chemical point of view is particularly challenging. Regardless of their exact molecular structure, scientists know how to recognize and map specific bonding configurations to mechanical equivalents. For examples, alkynes as act as bearings, azonitriles as actuating joints, and fullerenes as wheels. Scientists can exploit these insights if they have a flexible modeling language that has a clear, rigorously defined semantics, and is expressive enough to capture cyber-physical models in direct, natural form.

Effective methods for analysis, simulation and validation of models: Once a model has been built, there must be tools and methods to transform it an efficient machine-executable form. State-the-art compilation techniques can be used to perform this translation, but new technologies suchas multi-core architectures and reconfigurable computing architectures can only be exploited effectively if we take into account the nature of the class of models being executed. At the same time, tools are needed for both the manual and the automated analysis of various properties of these models (including stability, accuracy, and fidelity with respect to preservation of energy). Finally, once the models are built, they are validated against the real-life data and the existing analytical and rational relationship among the underlying phenomena.

Methods and techniques for guaranteeing real-time properties: Cyber-physical systems enable their users to monitor and manipulate the physical world. Many such systems involve real-time constraints. While recent advances in the theory of hybrid systems have laid a foundation for studying cyber-physical systems, these techniques must be adapted and extended to handle the modeling concepts that naturally arise in physics, and integrated with a wide range of analysis techniques to verify real-time properties of cyber-physical models.

The most promising innovations and abstractions are:

Techniques for staging well-established abstraction mechanisms: While existing programming abstraction mechanisms have been criticized as ill-suited for new computational challenges, we believe that the abstraction mechanisms in advanced language like ML and Scala are both quite powerful and under-utilized in practice. Instead, we believe that the real problem lies in how efficiently these abstractions are mapped to computational platforms. A particularly important enabling technology for reducing or elimination the runtime cost for abstraction is *staging*, the key technology underlying techniques such as program specialization or partial evaluation.

Gradual typing methods: Industrial practice indicates that the question of static versus dynamic typing is likely to never be resolved, and developers of computational artifacts, at best, must chose between the lesser of two evils. Recent work on *soft or gradual typing* provides evidence that combining the best of both worlds may yet be possible. The basic idea there is that computations can be initially described in an untyped form, and that typing information can be gradually added to establish higher-quality standards for the final product.

Expressive static type systems: Static type systems have now moved significantly beyond the scope of what is available in mainstream languages, as well as in terms of the guarantees that can be provided about the underlying computation. Significant engineering and technological challenges, however, must be addressed in order to make this technology available in mainstream languages and to adapt it to a framework that supports gradual typing.

Reconfigurable and heterogeneous computing infra-structure: Field Programming Gate Arrays (FPGAs) as well as a wide range of relating configurable computing architectures have significant potential for both increasing the computational power as well as reducing the energy cost for both embedded computation as well as offline simulation of cyber-physical systems. The challenge is to develop languages and tools to support the rapid and efficient coding of FPGAs for use in such systems.

Important milestones for the next 5 to 10 years will be:

Cyber-physical modeling environments that provide fidelity guarantees: Simulation systems need to tell us how accurate their results are.

Synthesis techniques that provide safety guarantees: Mappings from models that have been tested in simulators need to come with guarantees about how the software and hardware components will operate when embedded in the real-world.

Order of magnitude improvement in productivity for developers of cyber-physical systems: The ultimate measure of success of new technologies for cyber-physical computing will be in improved productivity in the development of reliable cyber-physical innovations and products.

Biographies:

Robert Cartwright: Professor Cartwright's principal research interests are programming language design and implementation, program specification, program testing and analysis, and software engineering.

Kevin Kelly: Prof. Kelly's research is focused on the physics and chemistry of surfaces at the nanometer level. He has investigated self-assembly of thiol molecules on gold and single-molecule electronic devices. His current research interests include the investigation of conducting polymers, carbon nanotubes, and single molecule devices such as the Nanocar.

Farinaz Koushanfar: Prof. Koushanfar's research interests is in the area of embedded networked systems, with a particular emphasis on sensor networks where she develops statistical modeling and combinatorial optimization tools and methods to address challenging real-life problems.

Walid Taha: Prof. Taha leads the Resource Aware Programming research group at Rice University, where he advises two postdoctoral associates, six PhD students, and several undergraduate students. His interest is in developing and applying programming language techniques that can improve the productivity of software and hardware developers.