Challenges in Transformation of Existing Real-Time Embedded Systems to Cyber-Physical Systems

Borzoo Bonakdarpour
Department of Computer Science and Engineering
Michigan State University
East Lansing MI 48824 USA
Email: borzoo@cse.msu.edu
Web: http://www.cse.msu.edu/~borzoo

Abstract

Most research directions in the context of cyber-physical systems have focused on developing computing foundations for designing, analyzing, and reasoning about systems being constructed from scratch. However, an equally important issue is to develop sound methods for transforming existing real-time embedded systems to ones that conform with standards of modern cyber-physical systems. This essay proposes several research directions for achieving such transformation techniques.

1. Motivation

The recent integration of embedded real-time systems with large collections of sensors and actuators in hostile physical environments, called *cyber-physical systems*, is posing many new and challenging problems to the computer science community. This is mostly because unlike existing deployments of embedded systems, in cyber-physical systems, *computational* processes are tightly coupled with a collection of *physical* processes. Thus, abstracting away physical-world factors from computational processes results in losing information crucial to decision making units of system. Indeed, in order to model and reason about cyber-physical systems, fundamentally new theories that host both *analytical* and *computational* models are needed. Clearly, developing such theories for modeling, analyzing, and constructing cyber-physical systems requires proposing entirely new levels of abstractions, which in turn introduces many new research directions [5, 6]. In fact, scientific study of problems regarding cyber-physical systems involves incorporating techniques from various disciplines and research areas including communication, control, composition, programming languages, modeling theory, formal methods, real-time, fault-tolerance, distributed systems and middleware, information theory, data mining, machine learning, hardware-software co-design, concurrency analysis, game theory, etc.

In this context, it is easy to perceive that the above research directions address problems on modeling, analyzing, building, and reasoning about cyber-physical systems that are being constructed from scratch. However, while investigating solutions to these problems should undoubtedly be carried out, I believe it is equally important to consider the following fundamental question as well:

> *How should we cure existing vulnerable deployments of real-time embedded systems that exhibit failures and time unpredictability in the face of physical processes?*

In order to address this question, we need to explore new avenues for *sound* transformation of existing embedded real-time systems to ones that conform to standards of modern cyber-physical systems, in an automated and formal manner. Developing such automated sound transformation methods for synthesizing new systems that are *correct-by-construction* is extremely crucial in cases where existing critical infrastructures must be protected. For instance, millions of people depend on the current US's air traffic control system everyday. This system consists of 500 FAA-managed ATC towers, along with 180 low-altitude radar control systems and 20 enroute centers [6]. Obviously, reconstructing such systems from scratch is infeasible.

Moreover, requirements of computational processes and behavior of physical processes in cyber-physical systems evolve constantly. For example, it is infeasible to identify all security measures or timing constraints of computational processes at requirements analysis time. In addition, it is impossible to anticipate all faults that a system is subject to at design time. These observations, as well, lead us to the importance of automated transformation techniques in the context of cyber-physical system.

With this motivation, in this essay, I propose several research directions for developing automated transformation techniques for *redesigning* existing deeply embedded real-time systems.

2. Challenges and Demands on Solutions

There exist three challenges in automated transformation of deeply embedded systems: (1) achieving *time predictability* in the face of openness of computational systems to concurrent physical processes, (2) resolving *conflicts* between concerns in cyber-physical systems (for instance, redundancy is key to fault-tolerance, but makes systems less secure), and (3) ensuring the *correctness* of the transformed system. In order to overcome the above challenges, I argue that any solution to the outlined transformation problem must meet the following criteria:

- *Scalability.* Transformation methods must scale to realistic-sized systems composed of several components to allow the construction of complex systems.
- *Reasonable level of abstraction.* A level of abstraction for the system at hand must be maintained. Specifically, the notion of *system* should not deal with low-level details such as sensing or node-to-node communication. Thus, systems can simply be specified by nondeterministic state-transition functions.
- *Multitolerance. Levels of fault-tolerance* specify the satisfaction of safety, liveness, and timing constrains in the presence of faults [1]. In cyber-physical systems, where physical processes impose different *types of faults* to computational

components, each computational component may require a different level of fault-tolerance with respect to each type of faults. Thus, the notion of fault-tolerance should be generalized to multitolerance in cyber-physical systems.

- *Hybridity.* Transformation methods must recognize the hybridity of cyber-physical systems. Put it another way, during transformation, both physical and computational processes must be treated as first-class citizens of the system at hand.
- *Extensive use of formal methods.* Obviously, applying unsound transformation methods to deal with safety-critical systems (e.g., air traffic control) is unacceptable. Thus, elicitation of formal methods from informal requirements such as expressing *functionality* of the system at hand using models of computation (e.g., transition systems and real-time temporal logic formulae), is a necessity. Consequently, the notion of automated transformation will be equivalent to synthesizing new systems by revising their existing version according to a set of requirements (e.g., fault-tolerance, time predictability, etc). In this context, one may argue that synthesis techniques are not known to be scalable. There are two answers to this argument. First, a brief survey of the literature of verification techniques shows that advances in this area in the past three decades have made it possible to scale up verification of systems with a few hundred states within a minute, to systems with 10^{624} states under a second [3]. Indeed, patience and continuous enormous research efforts have made model checking a fixed step of development process in many industrial places. Secondly, recent advances in synthesis of distributed fault-tolerant programs with 10^{30} states [2] also shows that the success in model checking may potentially take place in the context of automated synthesis as well. Therefore, I argue that incorporating automated formal analysis techniques such as program synthesis seems to be not only feasible, but also extremely beneficial to construct *correct* cyber-physical systems.

3. Research Directions

Synthesizing multitolerant cyber-physical systems. Synthesis of multitolerant systems has only been studied in the context of *untimed* systems. In order to develop synthesis algorithms for transforming fault-intolerant real-time embedded systems to multitolerant cyber-physical systems, first, one needs to formally define what it means for a real-time embedded system to be multitolerant. Such definitions are excepted to be quite different from untimed systems, as ensuring *bounded-time recovery* in the presence of different classes of faults makes problems considerably more complex. Subsequently, we need to address the following questions: What are the time and space complexity of automated synthesis of multitolerant real-time systems? Under what constraints is it possible to synthesize multitolerant systems in a *stepwise* manner by adding one level of fault-tolerance at a time? Under what conditions is it possible to devise *sound* and *complete* polynomial-time algorithms for synthesizing multitolerant systems? In what cases (if any) exponential complexity of synthesis algorithms is inevitable? Solutions to these problems and supporting tools will assist system designers to incrementally *add* multitolerance to existing real-time embedded systems.

Synthesizing fault-tolerant hybrid systems. A few blended theories of analytical and computational models have emerged to capture the hybridity of deeply embedded systems. In particular, the use of *hybrid automata* [4] makes it possible to model both analytical and computational behaviors of embedded systems at the same time. Thus, by developing a framework for defining the concept of levels of fault-tolerance in terms of safe and live semantics of hybrid automata, one can specify and reason about fault-tolerance properties of cyber-physical systems in a formal and elegant fashion. Meanwhile, the fundamental research problem in this context is to identify the possibilities and limitations in developing synthesis algorithms that transform fault-intolerant hybrid systems to fault-tolerant ones. Moreover, the formal framework can be generalized to capture multitolerant hybrid systems as well; building a rich theory of dependable cyber-physical systems.

Incorporating machine learning and data mining techniques. As mentioned in Section 2, the input to the transformation problem is a real-time embedded system in terms of a state-transition function and the goal is to generate another system which satisfies a set of properties. Since state-transition functions can be represented by (possibly weighed) directed graphs, the transformation problem can be formulated as a graph transformation problem as well. In fact, the output is the transformed input graph that satisfies the set of properties of interest.
In recent years, an increasing interest in the use of *graph mining* algorithms has been emerged in the graph transformation community. Interestingly, transformation of real-time embedded systems can be elegantly formulated as a graph mining problem as follows. Having a database of real-time embedded systems (i.e., directed graphs) that satisfy a property of interest and another database containing systems that do not satisfy the property, the problem is to identify possibilities and limitations of deciding the exitance of a solution to the transforation problem for a particular input embedded real-time system with respect to the property of interest. Furthermore, if the answer to the decision problem is affirmative, another research problem is to devise algorithms that efficiently find a witness to the decision problem.

I believe solutions to the above problems are as instrumental as the current research on cyber-physical systems. I envision the research agenda to be enormous, spanning many disciplines including formal methods, theory of computation, graph theory, supervisory control, dependability, real-time computing, data mining, and machine learning. Collaboration with the systems theory community is also essential to conduct case studies for benchmarking and evaluating the theoretical results.

References

[1] B. Bonakdarpour and S. S. Kulkarni. Incremental synthesis of fault-tolerant real-time programs. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, LNCS 4280, pages 122–136, 2006.

[2] B. Bonakdarpour and S. S. Kulkarni. Exploiting symbolic techniques in automated synthesis of distributed programs with large state space. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 3–10, 2007.

[3] G. Ciardo, G. Lüttgen, and R. Siminiceanu. Saturation: An efficient iteration strategy for symbolic state-space generation. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 328–342, 2001.

[4] T. A. Henzinger. The theory of hybrid automata. In *IEEE Symposium on Logic in Computer Science (LICS)*, pages 278–292, 1996.

[5] E. A. Lee. Cyber-physical systems - are computing foundations adequate? In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, October 2006.

[6] J. A. Stankovic, I. Lee, A. K. Mok, and R. Rajkumar. Opportunities and obligations for physical computing systems. *IEEE Computer*, 38(11):23–31, 2005.