

Modeling and Verification of Distributed Cyber-Physical Systems: Challenges, Research Needs, and Possible Roadmap

D. M. Tilbury
The University of Michigan
Department of Mechanical Engineering
Ann Arbor, MI 48109-2125
(734) 936-2129
tilbury@umich.edu

October 5, 2006

As sensors and computers increase in power and decrease in price, controllers are becoming ubiquitous. In most applications, an actuator has always been present, although it may have been “controlled” by a person or a mechanical linkage. The addition of a simple sensor and microprocessor allows the loop to be closed in real time. Feedback controllers are present in everything from basic appliances to building automation systems and high-performance aircraft.

As networking technology continues to increase in bandwidth and decrease in price, these ubiquitous controllers are becoming interconnected. This widespread connectivity can be advantageous: sensor data can be shared to improve control performance, greater operator awareness is possible by monitoring multiple systems simultaneously, and design redundancy is facilitated.

On the other hand, systems connected by networks can exhibit surprising, and often undesirable, behavior. Although each subsystem may be well-modeled and extensively tested, the interactions between the subsystems may not be well-understood. Since these systems may be part of a critical infrastructure, their performance must be guaranteed under both normal and abnormal operating conditions. Performance guarantees require a model of the physical system, a model of the controller (and associated software), and a formal specification.

1 Important challenges

In the manufacturing industry today there is a trend towards “converged modular automation.” In this vision, machines and systems are built up of cyber-physical modular components, each with its own embedded controller, and connected by a network. The nodes on the network are intelligent, and the network is used not only for a supervisory controller to collect information but also for peer-to-peer information exchange. These types of modular automation systems are more reconfigurable than standard custom-designed systems. However, since a module that contains an embedded controller is more complex than a “dumb” module, the assembled system is more complex than a typical system. Thus, the system may require more extensive testing and validation before it can be deployed.

In this section, we outline some of the challenges in modeling and verification of these cyber-physical in the manufacturing industry. Similar challenges are expected to arise in other domains.

1.1 Modeling the physical system and interface

The performance of the automation system cannot be verified without a model of the physical system that it is controlling. Traditionally, physical laws (such as Newton’s) have been used to derive differential equations that describe the behavior of system components. These approaches can still be useful when designing the modular controllers.

Instead of (or in addition to) servo control, the modular controllers will often be doing sequence or logic control. In these cases, the behavior of the subsystems are often more usefully described by discrete-event or logical models. Often, simple controllers are defined as a set of rules implemented as “if-then” statements in software, instead of as a discrete-time approximation of a differential equation.

Although the sequence of operations of a slide or a conveyor is usually quite simple and can be written down directly in a state diagram or Petri net, the normal operation is only a small fraction of what the controller must handle. Ideally, the module must operate correctly and safely under any type of fault conditions (including loss of power).

Techniques are needed to automatically extract — from physical system descriptions — appropriate models for the interface between the physical system and a discrete or logical controller implemented in software. This is accomplished today for many critical systems, through painstaking descriptions by human model-builders. This manual model-building process is too time-consuming and error-prone to be possible for the wide variety of cyber-physical systems that will be built in the coming years.

1.2 Model validation and abstraction

Once interface models are built for the physical system, they must be validated. Model validation is typically a time-consuming and expensive proposition. It is not cost-effective for most manufacturing systems, which are built as one-off solutions.¹ Even though high-fidelity 3D CAD models of the factory may be constructed during the design phase, they are rarely used after the factory is built. Changes are often made during the construction process, and so the model is no longer accurate. This makes reconfiguration difficult.

However, with the move to modular automation components, it may be cost-effective to not only build high-fidelity models, but also to validate them. Because the components will be built in quantity and used in many different designs, the cost required to perform the validation can be amortized. The validated models of the modules will facilitate future reconfigurations.

Definitions and theories exist for system identification and uncertainty modeling in the time and frequency domains. I am not aware of analogous definitions in the discrete-event and logical domain, especially as related to physical systems behavior and fault modeling. These definitions are needed for certification and validation of the component models.

1.3 Modular verification of the closed-loop behavior

Many verification techniques have been proposed in such diverse domains as logic control, discrete event systems, software systems, and databases. Important properties to verify include internal consistency, absence of deadlock, and the satisfiability of certain logical formulae or specifications. Most verification techniques require the explicit enumeration of all possible states in the system. Although this is theoretically possible when the state space is finite, it is not practically possible when the number of states in the system approaches the number of atoms in the universe. Thus, key limitations arise in the scale and complexity of systems that can be handled.

In the context of logic controllers modeled as discrete event systems, we have demonstrated that some key properties can be verified modularly, without constructing the entire state space of the system. However, these results are quite limited. There are many more properties that cannot currently be verified modularly. Indeed, there is not a good understanding of what properties could be verified modularly, and which (if any) are “global” properties and inherently depend on the entire state space.

1.4 Implementation and standards considerations

When considering the control technology to be implemented in these modular automation systems, there is a tradeoff to be considered between COTS open-architecture solutions and proprietary offerings. The open-architecture solutions may be less expensive initially, but there is less trust in them on the plant floor. There are also security concerns,

¹This is beginning to change; for example, GEMA (a joint venture of DaimlerChrysler, Hyundai, and Mitsubishi) plans 5 identical engine plants, 2 in the US and 3 in Asia.

especially when using widely-available software and hardware. On the other hand, proprietary solutions may not have the same upgrade or reconfiguration path available, which is a longer-term concern. Thus, the types of hardware and software that will be used for these modular automation components remains open for debate.

Another issue with these new cyber-physical systems is their conformance with international standards. There are dozens of “open” standards for networks (CIP, Modbus, . . .), several standards for logic programming (IEC 61131, 61499; note that most of these do not guarantee interoperability, or even unified execution semantics), and national and international safety standards (NFPA, RIA, IEC 61508, . . .). Automation modules should comply with at least some of these standards. In addition, in order for them to work together, they will need to communicate with a standard protocol (XML, Web services, . . .). Defining the correct standards that will guarantee interoperability, and certifying that the modules comply with the appropriate standards, remain open challenges.

Each automation module will need its own HMI, but when they are all connected together in a system, a unified HMI should control the entire system. Thus, standard techniques are needed to build and operate a unified HMI. This will be enabled through the formal models of the systems.

2 Important information technology research needs

The important research needs are in modeling and verification of combined mechanical and computing systems.

- Automatically generating models of physical systems that can interface with models of computing system operation
- Specifications of model correctness and uncertainty, and methods for validating these specifications
- Abstraction methods for modules, to manage complexity
- Specifications for reconfigurability, and methods to validate these specifications
- Specifications (interface models) that guarantee interoperability
- Certification methods to guarantee conformance with international standards
- Verification techniques for modular cyber-physical systems

3 Possible roadmap

1. Survey of existing techniques for modular design in mechanical and software domains
2. Mapping potential synergies from mechanical and software design domains to cyber-physical systems
3. Definition/identification of “challenge problem”
4. Design and construction of testbed, with physical modules distributed at multiple sites (simulations at all sites)
5. Definition of interface specifications for modules
6. Demonstration of module operations, validation of conformance with interface specification
7. Validation of simulation model in normal and fault modes
8. Demonstration of module interoperation in normal mode
9. Demonstration of module interoperation in fault mode, and fault recovery using unified HMI
10. Definition of reconfiguration scenario that was not originally envisioned
11. Demonstration of reconfiguration through adding a new module and/or rearranging existing modules

Biography of D. M. Tilbury

Dawn M. Tilbury received the B.S. degree in Electrical Engineering, *summa cum laude*, from the University of Minnesota in 1989, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1992 and 1994, respectively. In 1995, she joined the faculty of the Mechanical Engineering Department at the University of Michigan, Ann Arbor, where she is currently an Associate Professor. She won the EDUCOM Medal (jointly with Professor William Messner of Carnegie Mellon University) in 1997 for her work on the web-based Control Tutorials for Matlab. An expanded version, *Control Tutorials for Matlab and Simulink*, was published by Addison-Wesley in 1999. She is co-author (with Joseph Hellerstein, Yixin Diao, and Sujay Parekh) of the textbook *Feedback Control of Computing Systems*. She received an NSF CAREER award in 1999, and is the 2001 recipient of the Donald P. Eckman Award of the American Automatic Control Council. She was a member of the 2004–2005 class of the Defense Science Study Group (DSSG) and is a current member of DARPA's Information Science and Technology Study Group (ISAT). Her research interests include distributed control of mechanical systems with network communication, logic control of manufacturing systems, and uncertainty modeling in cooperative control. She belongs to ASEE, ASME, IEEE, and SWE, and is an elected member of the IEEE Control Systems Society Board of Governors.