Methods and Tools for Verification of Cyber-Physical Systems (CPS)

Chris J. Myers, Jian Wu, Zhen Zhang
University of Utah

Hao Zheng, Yingying Zhang, Ping Hou
University of South Florida

Cyber-Physical Systems (CPS) are deployed in a wide variety of safety critical applications from avionics, medical, and automotive domains. For these applications, it is essential to create a precise specification and formally verify that the implementation behaves as specified. The formal verification of these systems presents a wide variety of challenges. Models of these systems must represent the physical world, analog sensors and actuators, computer hardware and software, networks, and feedback control. These models must deal with the fact that correctness may depend on timing, concurrency, system dynamics, and stochastic behavior. To address this, our research project has developed a general hybrid system modeling formalism that is capable of representing continuous and discrete dynamics, timing, and stochastic behavior [2].

The complexity of CPS models makes the formal verification of them using techniques such as *model checking* extremely difficult. To address this, it is essential to develop more scalable verification methods which is the primary aim of this research project. In particular, we have developed automated *abstraction* methods to reduce model complexity [1, 4], *partial order* methods to address model concurrency [5], *symbolic methods* to utilize efficient data representations [6], and *compositional reasoning* methods to exploit model structure [7].

In order to validate our new verification methodologies, our group, in collaboration with Professor Tomohiro Yoneda at the National Institute of Informatics in Tokyo, has been developing a new fault tolerant routing algorithm for *networks-on-chip* (NoC) [3]. One CPS application that can leverage this router is in automotive electronic systems which often require more than 50 *electronic control units* (ECUs) to operate everything from the entertainment system to the anti-lock breaks. Currently, each ECU is statically tied to specific sensors and actuators which means that processing power of each ECU cannot be shared, and when an ECU fails, it causes a malfunction in the corresponding sensor/actuator. With a NoC approach, it makes the mapping between ECUs and sensors/actuators flexible allowing for a sharing of processing power and enabling fault tolerance by having spare units. Using a NoC approach, however, does have its challenges. Namely, a NoC must be carefully designed to avoid deadlock and be fault-tolerant while still achieving latency and throughput goals. This verification of the NoC router is a perfect case study since its correctness is highly dependent on concurrent, timing, and stochastic behavior.

References

[1] R. Thacker, K. Jones, C. Myers, and H. Zheng. Automatic abstraction for verification of cyber-physical systems. In *The 1st ACM/IEEE International Conference on Cyber-Physical Systems*, April 2010.

[2] R. Thacker, C. Myers, K. Jones, and S. Little. A new verification method for embedded systems. In *The 27th IEEE International Conference on Computer Design*, October 2009.

[3] J. Wu, Z. Zhang, and C. Myers. A fault-tolerant routing algorithm for a network-on-chip using a link fault model. In *2011 Virtual Worldwide Forum for PhD Researchers in EDA*, November 2011.

[4] H. Yao, H. Zheng, and C. Myers. State space reductions for scalable verification of asynchronous designs. In *2010 IEEE Int. High Level Design Validation and Test Workshop*, November 2010.

[5] Y. Zhang, E. Rodriguez, H. Zheng, and C. Myers. An improvement in partial order reduction using behavioral analysis. In *IEEE Computer Society Annual Symp. on VLSI*, August 2012.

[6] H. Zheng, A. Price, and C. Myers. Using decision diagrams to compactly represent state space for explicit model checking. In *2012 IEEE Int. High Level Design Validation and Test Workshop*, November 2012.

[7] H. Zheng, E. Rodriguez, Y. Zhang, and C. Myers. A compositional minimization approach for large asynchronous design verification. In *19th Int. SPIN Workshop on Model Checking of Software*, July 2012.