Formal Verification of Transportation Systems

Chris J. Myers
University of Utah

Hao Zheng
University of South Florida

Cyber-Physical Systems (CPS) are being deployed in a wide variety of safety critical transportation systems from autonomous vehicles to flight traffic control. 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. Currently, many system modeling languages exist each with unique modeling capabilities ranging from SystemC [2] for software-centered systems to SystemVerilog [3] for hardware-centered systems to Modelica [1] for physical-centered systems. While good for their domain, they are awkward to use outside their primary domain. To address this problem, the community must develop a standard general system modeling language that is capable of representing continuous and discrete dynamics, timing, and stochastic behavior. A standard representation is essential to enable the development of benchmarks and to exchange models between software tools and research groups.

The complexity of CPS models makes the formal verification of them using techniques such as *model checking* extremely difficult. Many verification methods are limited to only check functional correctness. While some verification methods exist to check timing, continuous dynamics, or stochastic behavior, they often do not scale as well as functional approaches. There have been some attempts to map these more complex problems to functional verification tools, but they are typically not accurate enough. A critical problem, therefore, is the development of a compositional verification methodology that enables system designers to decompose their system models into components which can be verified with the appropriate verification engine. A mechanism then needs to be constructed to integrate these results between the engines. A standard system modeling language would facilitate this "divide and rule" approach enabling dramatic scaling in the systems that can be formally verified.

References

[1] Tiller M., Bowles P., and Dempsey M. Development of a vehicle model architecture in modelica. In *Proceedings of the 3rd International Modelica Conference*, 2003.

[2] Elvinia Riccobene, Patrizia Scandurra, Sara Bocchio, Alberto Rosti, Luigi Lavazza, and Luigi Mantellini. Systemc/c-based model-driven design for embedded systems. *ACM Trans. Embed. Comput. Syst.*, 8(4):30:1–30:37, July 2009.

[3] Chris Spear. *SystemVerilog for Verification, Second Edition: A Guide to Learning the Testbench Language Features.* Springer Publishing Company, Incorporated, 2nd edition, 2008.