Formalisms for Cyber-Physical Systems in Transportation

Gul Agha
Department of Computer Science
University of Illinois, Urbana, Illinois, USA
http://osl.cs.illinois.edu
email: agha@illinois.edu

December 16, 2013

Modern automative systems involve dozens of embedded computers and hundreds of sensors that are networked together. The software running these nodes is hundreds of thousands of lines of code. The computing platforms used today are difficult to reason about or update as hardware and software environments are updated to add new functionalities. We envision building cps applications on *open computing platforms* which concurrently execute large numbers of actors [1] performing both coordinated and autonomous tasks, and potentially sharing resources between different applications [4]. We believe that a looser coupling between middleware, OS, and applications, can be facilitate the rapid evolution of hardware and software without sacrificing efficiency or safety.

We propose to build on our prior research in models and programming tools for large-scale concurrent and distributed systems using the actor model. Each actor is a reactive entity whose behavior can be abstractly represented by a big step semantics [2]. The system as a whole may be viewed as a probabilistic program in a distributed space and time, where some events are asynchronous while others may be probabilistic or continuous in nature [3]. Such a system may be formalized by a two level probabilistic actor-based programming model: the applications level and the meta-level representing an open computing platform (cf [5]). Different formalisms are needed to capture different aspects of the behavior of such systems. We propose the use of a number of formalism to specify, derive and reason about the behavior of such systems, including statistical model checking, euclidean model checking, constraint solving, predictive runtime verification, and testing.

References

[1] G. Agha. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, 1986.

[2] G. Agha, I. A. Mason, S. F. Smith, and C. L. Talcott. A foundation for actor computation. *J. Funct. Program.*, 7(1):1–72, 1997.

[3] G. A. Agha, J. Meseguer, and K. Sen. Pmaude: Rewrite-based specification language for probabilistic object systems. *Electr. Notes Theor. Comput. Sci.*, 153(2):213–239, 2006.

[4] K. Mechitov and G. Agha. An architecture for dynamic service-oriented computing in networked embedded systems. In M. Heisel, editor, *Software Service and Application Engineering*, volume 7365 of *Lecture Notes in Computer Science*, pages 147–164. Springer, 2012.

[5] N. Venkatasubramanian, C. L. Talcott, and G. Agha. A formal model for reasoning about adaptive qos-enabled middleware. *ACM Trans. Softw. Eng. Methodol.*, 13(1):86–147, 2004.