Demo: CyPhySim — A Cyber-Physical Systems Simulator*

Christopher Brooks
EECS Department
UC Berkeley
Berkeley, CA, USA
cxh@eecs.berkeley.edu

Edward A. Lee
EECS Department
UC Berkeley
Berkeley, CA, USA
eal@eecs.berkeley.edu

David Lorenzetti
Lawrence Berkeley
National Laboratory
Berkeley, CA, USA
dmlorenzetti@lbl.gov

Thierry S. Nouidui
Lawrence Berkeley
National Laboratory
Berkeley, CA, USA
tsnouidui@lbl.gov

Michael Wetter
Lawrence Berkeley
National Laboratory
Berkeley, CA, USA
mwetter@lbl.gov

ABSTRACT

This demo provides a preview of a pre-release version of CyPhySim, an open-source simulator for cyber-physical systems. This simulator supports discrete-event models, quantized-state simulation of continuous dynamics, the Functional Mockup Interface (FMI), classical (Runge-Kutta) simulation of continuous dynamics, modal models (hybrid systems), discrete-time (periodic) systems, and algebraic loop solvers. CyPhySim provides a graphical editor, an XML file syntax for models, and an open API for programmatic construction of models.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: General

1. INTRODUCTION

The heterogeneity of cyber-physical systems presents considerable challenges to software simulation techniques. CyPhySim is a new open-source software simulator (BSD license) that supports the most promising combination of technologies. The core of the system is a discrete-event simulation engine from the open source Ptolemy II system [17].

*This work was supported in part by the iCyPhy Research Center (Industrial Cyber-Physical Systems, supported by IBM and United Technologies), by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231, and by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley (supported by the National Science Foundation, NSF award #0931843 (ActionWebs), the Naval Research Laboratory (NRL #N0013-12-1-G015), and the following companies: Denso, National Instruments, and Toyota.

HSCC '15 Apr 14-16, 2015, Seattle, WA, USA
ACM 978-1-4503-3433-4/15/04.
http://dx.doi.org/10.1145/2728606.2728641

2. MODEL OF TIME

For models that mix discrete and continuous behaviors, it is well established that the model of time that is used must support sequences of causally-related instantaneous events [9, 2]. CyPhySim uses superdense time [13, 11].

3. DISCRETE-EVENT SIMULATION

In the style of discrete-event (DE) modeling realized in CyPhySim, a model is a network of actors with input and output ports. The actors send each other time-stamped events, and the simulation processes these events in time stamp order. This style of DE is widely used for simulation of large, complex systems [3, 20, 5]. CyPhySim builds on the particular implementation in Ptolemy II, which has a sound, deterministic semantics [8, 14].

4. QUANTIZED-STATE SYSTEMS

A relatively recent development in numerical simulation of ordinary differential equations is the emergence of so-called *quantized-state systems* (QSS) [19, 7, 4, 6, 1]. In a classical ODE simulator, a step-size control algorithm determines sample times, and a sample value is computed at those times for all states in the model. In a QSS simulator, each state has its own sample times, and samples are processed using a DE simulation engine in time-stamp order. The sample time of each state is determined by quantizing the value of each state and producing samples only when the value changes by a pre-determined tolerance, called the quantum. Higher-order variants incorporate knowledge of higher-order derivatives of a state to predict trajectories and produce samples only when these higher order prediction differs by more than the quantum [15]. For some systems, QSS yields efficient simulation by producing samples only when predicted state trajectories exceed the quantum. Moreover, state events can be scheduled using an explicit equation, avoiding iteration in time. Unlike classical ODE solvers, QSS solvers never require backtracking, greatly simplifying simulation.

5. CLASSICAL ODE SOLVERS

Some continuous systems are still better simulated using classical ODE solvers, so CyPhySim makes available the Continuous domain of Ptolemy II, which provides Runge-Kutta solvers. A model can have hierarchical mixtures of such classical solvers, QSS models, and DE models.

6. FMI

The Functional Mockup Interface (FMI) is a standard for model exchange and co-simulation of dynamical models [16]. CyPhySim supports importing FMI components designed for model exchange, and provides a QSS simulation engine for numerical integration. This strategy also enables co-simulation of DE models with FMI, a combination also described in [18].

7. MODAL MODELS

CyPhySim imports the modal models of Ptolemy II [10], which provide hierarchical combinations of state machines and continuous and DE subsystems. This combination supports hybrid system modeling with rigorous deterministic semantics [11].

8. DISCRETE-TIME SYSTEMS

Many CPS applications include sampled-data subsystems with regular, periodic sample rates. CyPhySim leverages the synchronous-reactive (SR) domain of Ptolemy II to provide such models, which permits specification of a sample rate and enables structured multi rate systems. Such SR models interoperate well with DE, QSS, and Continuous models [12]. CyPhySim incorporates a new innovation that enables arbitrary hierarchical nesting of these models. Sampled data systems can contain continuous or DE subsystems and vice versa.

9. ALGEBRAIC LOOP SOLVERS

Finally, CyPhySim includes a mechanism for specifying algebraic loop solvers, including a simple successive substitution mechanism, a Newton-Raphson solver, and a homotopy method. The model builder is given explicit control over the solution method and initial guesses in order to ensure deterministic results.

10. REFERENCES

[1] Bliudze, S., and Furic, S. An operational semantics for hybrid systems involving behavioral abstraction. In *Modelica Conference* (Lund, Sweden, 2014).

[2] Broman, D., Greenberg, L., Lee, E. A., Masin, M., Tripakis, S., and Wetter, M. Requirements for hybrid cosimulation standards. In *Hybrid Systems: Computation and Control (HSCC)* (Seattle, 2015).

[3] Cassandras, C. G. *Discrete Event Systems, Modeling and Performance Analysis*. Irwin, 1993.

[4] Cellier, F. E., and Kofman, E. *Continuous System Simulation*. Springer, 2006.

[5] Fishman, G. S. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag, 2001.

[6] Floros, X., Cellier, F. E., and Kofman, E. Discretizing time or states? a comparative study between dassl and qss - work in progress paper -. In *Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (EOOLT)* (Oslo, 2010), Linköping University.

[7] Kofman, E., and Junco, S. Quantized-state systems: A DEVS approach for continuous system simulation. *Transactions of The Society for Modeling and Simulation International 18*, 1 (2001), 2–8.

[8] Lee, E. A. Modeling concurrent real-time processes using discrete events. *Annals of Software Engineering 7* (1999), 25–45.

[9] Lee, E. A. Constructive models of discrete and continuous physical phenomena. *IEEE Access 2*, 1 (2014), 1–25.

[10] Lee, E. A., and Tripakis, S. Modal models in Ptolemy. In *3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools (EOOLT)* (Oslo, Norway, 2010), vol. 47, Linköping University Electronic Press, Linköping University, pp. 11–21.

[11] Lee, E. A., and Zheng, H. Operational semantics of hybrid systems. In *Hybrid Systems: Computation and Control (HSCC)* (Zurich, 2005), M. Morari and L. Thiele, Eds., vol. LNCS 3414, Springer-Verlag, pp. 25–53.

[12] Lee, E. A., and Zheng, H. Leveraging synchronous language principles for heterogeneous modeling and design of embedded systems. In *EMSOFT* (Salzburg, Austria, 2007), ACM, pp. 114 – 123.

[13] Maler, O., Manna, Z., and Pnueli, A. From timed to hybrid systems. In *Real-Time: Theory and Practice, REX Workshop* (1992), Springer-Verlag, pp. 447–484.

[14] Matsikoudis, E., and Lee, E. A. On fixed points of strictly causal functions. In *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)* (Buenos Aires, Argentina, 2013), vol. LNCS 8053, Springer-Verlag, pp. 183–197.

[15] Migoni, G., Bortolotto, M., Kofman, E., and Cellier, F. E. Linearly implicit quantization-based integration methods for stiff ordinary differential equations. *Simulation Modelling Practice and Theory 35* (2013), 118–136.

[16] Modelisar Consortium and the Modelica Association. Functional mock-up interface for model exchange and co-simulation. Report Version 2.0, July 25 2014. `https://www.fmi-standard.org/downloads`.

[17] Ptolemaeus, C., Ed. *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, Berkeley, CA, 2014.

[18] Savicks, V., Butler, M., and Colley, J. Co-simulating Event-B and continuous models via FMI. In *Summer Simulation Multi-Conference (SummerSim)* (San Diego, 2014), pp. 259–256.

[19] Zeigler, B. P., and Lee, J. S. Theory of quantized systems: Formal basis for DEVS/HLA distributed simulation environment. In *SPIE Conference on Enabling Technology for Simulation Science* (Orlando, 1998), vol. SPIE Vol. 3369, pp. 49–58.

[20] Zeigler, B. P., Praehofer, H., and Kim, T. G. *Theory of Modeling and Simulation*, 2nd ed. Academic Press, 2000.