Exploring Models of Computation with Ptolemy II

Christopher Brooks
EECS Department
University of California, Berkeley
Berkeley, CA, USA
cxh@eecs.berkeley.edu

Edward A. Lee
EECS Department
University of California, Berkeley
Berkeley, CA, USA
eal@eecs.berkeley.edu

Stavros Tripakis
EECS Department
University of California, Berkeley
Berkeley, CA, USA
stavros@eecs.berkeley.edu

ABSTRACT

The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, embedded systems. The focus is on assembly of concurrent components. The key underlying principle in the project is the use of well-defined models of computation that govern the interaction between components. A major problem area being addressed is the use of heterogeneous mixtures of models of computation.

Ptolemy II takes a component view of design, in that models are constructed as a set of interacting components. A model of computation governs the semantics of the interaction, and thus imposes an execution-time discipline. Ptolemy II has implementations of many models of computation including Synchronous Data Flow, Kahn Process Networks, Discrete Event, Continuous Time, Synchronous/Reactive and Modal Models

This hands-on tutorial explores how these models of computation are implemented in Ptolemy II and how to create new models of computation such as a "non-dogmatic" Process Networks example and a left-to-right execution policy example.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: *real-time and embedded systems*; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*cyber-physical systems*; I.6.5 [**Simultation and Modeling**]: Model Development[models cyber-physical interactions]; J.7 [**Computers in Other Systems**][computer-based systems]

General Terms

Design, Experimentation

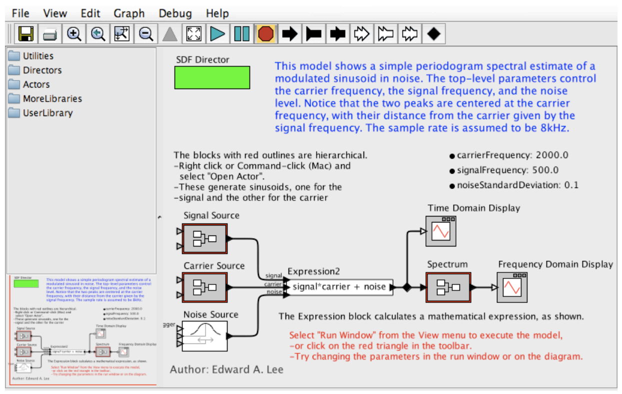
Keywords

Modeling, simulation, concurrency

1. INTRODUCTION

Ptolemy II [1] is an open-source software framework supporting experimentation with actor-oriented design. Actors are software components that execute concurrently and communicate through messages sent via interconnected ports. A model is a hierarchical interconnection of actors. In Ptolemy II, the semantics of a model is not determined by the framework, but rather by a software component in the model called a director, which implements a model of computation. The Ptolemy Project has developed directors supporting Kahn process networks (PN) [3], discrete-events (DE), dataflow (SDF), synchronous/reactive(SR), rendezvous-based models, 3-D visualization, and continuous-time models. Each level of the hierarchy in a model can have its own director, and distinct directors can be composed hierarchically. A major emphasis of the project has been on understanding the heterogeneous combinations of models of computation realized by these directors. Directors can be combined hierarchically with state machines to make modal models [4, 5]. A hierarchical combination of continuous-time models with state machines yields hybrid systems [6]; a combination of synchronous/reactive with state machines yields Statecharts [2] (the Ptolemy II variant [7] is close to SyncCharts).

Ptolemy II has been under development since 1996; it is a successor to Ptolemy Classic, which was developed since 1990. The core of Ptolemy II is a collection of Java classes and packages, layered to provide increasingly specific capabilities. The kernel supports an abstract syntax, a hierarchical structure of entities with ports and interconnections. A graphical editor called Vergil supports visual editing of this abstract syntax. An XML concrete syntax called MoML provides a persistent file format for the models. Various specialized tools have been created from this framework, including HyVisual (for hybrid systems modeling), Kepler (for scientific workflows), VisualSense (for modeling and simulation of wireless networks), Viptos (for sensor network design), and some commercial products. Key parts of the infrastructure include an actor abstract semantics, which enables the interoperability of distinct models of computation with a well-defined semantics; a model of time (specifically, super-dense time, which enables interaction of continuous dynamics and imperative logic); and a sophisticated type system supporting type checking, type inference, and polymorphism. The type system has recently been extended to support user-defined ontologies [8]. Various experiments with synthesis of implementation code and abstractions for verification are included in the project.



This model shows a simple periodogram spectral estimate of a modulated sinusoid in noise. The top-level parameters control the carrier frequency, the signal frequency, and the noise level. Notice that the two peaks are centered at the carrier frequency, with their distance from the carrier given by the signal frequency. The sample rate is assumed to be 8kHz.

The blocks with red outlines are hierarchical.
-Right click or Command-click (Mac) and select "Open Actor".
-These generate sinusoids, one for the
-signal and the other for the carrier

- carrierFrequency: 2000.0
- signalFrequency: 500.0
- noiseStandardDeviation: 0.1

The Expression block calculates a mathematical expression, as shown.

Select "Run Window" from the View menu to execute the model,
-or click on the red triangle in the toolbar.
-Try changing the parameters in the run window or on the diagram.

Author: Edward A. Lee

Figure 1: An example of a Vergil window. Vergil is a graphical editor for Ptolemy II.

2. SPEAKERS

- **Edward A. Lee**, (eal@eecs.berkeley.edu)
  `http://ptolemy.org/~eal`

  Edward A. Lee is the Robert S. Pepper Distinguished Professor of the Electrical Engineering and Computer Sciences (EECS) department at U.C. Berkeley. His research interests center on design, modeling, and simulation of embedded, real-time computational systems. He is the director of the Ptolemy project.

- **Stavros Tripakis** (stavros@eecs.berkeley.edu)
  `http://www-verimag.imag.fr/~tripakis`

  Stavros Tripakis is Associate Research Scientist at UC Berkeley. His work lies in the areas of embedded, real-time and distributed systems, model-based and component-based design, verification, testing and synthesis.

- **Christopher Brooks** (cxh@eecs.berkeley.edu)
  `http://ptolemy.org/~cxh`

  Christopher Brooks is the Ptolemy Project Software Manager and the Executive Director of the the Center for Hybrid and Embedded Software Systems (CHESS).

3. TOPICS

The tutorial is a hands-on tutorial of 3 to 3.5 hours on Sunday, October 24th, 2010.

- (13:00-14:00) Setting up Ptolemy II 8.0 and Eclipse: installing Eclipse; configuring Ptolemy; compiling Ptolemy; running a model within Eclipse.
- (14:00-15:00) Overview of Models of Computation in Ptolemy II: abstract semantics; actors and directors; currently implemented models of computation (Synchronous Data Flow, Dynamic Data Flow, Kahn Process Networks, Discrete Event, Continuous Time, Synchronous/Reactive and Modal Models); combining models of computation.
- (15:00 - 15:30) Break
- (15:30 - 17:00) Extending Ptolemy Models of Computation: Ptolemy execution semantics object model; using Eclipse to extend Ptolemy; building a non-dogmatic Kahn Process Networks model of computation; building a left to right model of computation.

The tutorial is similar to material presented at the February, 2009 Ptolemy Miniconference tutorial, (35 attendees, http://ptolemy.org/conferences/09/tutorial.htm) and at the February, 2007 Ptolemy Miniconference tutorial, (30 attendees, http://ptolemy.org/conferences/07).

4. REFERENCES

[1] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming heterogeneity—the Ptolemy approach. *Proceedings of the IEEE*, 91(2):127–144, 2003.

[2] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.

[3] G. Kahn and D. B. MacQueen. Coroutines and networks of parallel processes. In B. Gilchrist, editor, *Information Processing*, pages 993–998. North-Holland Publishing Co., 1977.

[4] E. A. Lee. Finite State Machines and Modal Models in Ptolemy II. Technical Report UCB/EECS-2009-151, EECS Department, University of California, Berkeley, Nov 2009.

[5] E. A. Lee and S. Tripakis. Modal Models in Ptolemy. In *EOOLT 2010 – 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools*. Linköping University Electronic Press, Oct. 2010. To appear.

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

[7] E. A. Lee and H. Zheng. Leveraging synchronous language principles for heterogeneous modeling and design of embedded systems. In *EMSOFT*, Salzburg, Austria, 2007. ACM.

[8] J. M.-K. Leung, T. Mandl, E. A. Lee, E. Latronico, C. Shelton, S. Tripakis, and B. Lickly. Scalable semantic annotation using lattice-based ontologies. In *12th International Conference on Model Driven Engineering Languages and Systems*, pages 393–407. ACM/IEEE, October 2009.