May 20, 1996

# System-Level Design Methodology for Embedded Signal Processors

**by *The Ptolemy Team*,[1] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1772, *http://ptolemy.eecs.berkeley.edu*/**

## 1. Objectives

This project focuses on design methodology for complex real-time systems in which a variety of design methodologies and implementation technologies must be combined. Design methodologies are encapsulated in models of computation, whereas implementation technologies are implemented as synthesis tools. Applications that use more than one model of computation and/or more than one synthesis tool are *heterogeneous*. Hardware/software codesign is one example of heterogeneous design.

## 2. Technical Approach

The key idea in the Ptolemy project is to mix models of computation, implementation languages, and design styles, rather than trying to develop one all-encompassing technique. The rationale is that specialized design techniques are (1) more useful to the system-level designer and (2) more amenable to high-quality high-level synthesis of hardware and software.

## 3. Technical Contributions

We support heterogeneity by defining formal models of computation such as dataflow and finite-state machines at arbitrary levels of granularity, hierarchical mechanisms to compose models into complex systems, and algorithms to synthesize systems in hardware and software. We address:

- *Algorithm Specification*: graphical/algebraic specification [1]; algorithm restructuring [1]
- *System Specification*: scalable systems; syntax management; integrated documentation; design methodology management [1]
- *Modeling*: multidimensional dataflow [2]; finite state machines (FSM); synchronous/reactive (SR) controllers; process networks; formal mathematical analysis of dataflow models
- *Scheduling of Dataflow Graphs*: joint minimization of program and data memory for embedded DSP processors [3]; optimization of synchronized communication between processors [3]; hierarchical scheduling [4]; incremental compilation [4]; dynamic scheduling
- *Simulation*: compiled simulation [4]; mixing models of control (FSM, SR), dataflow, and discrete-event subsystems [5]; mixed-signal analog/digital simulation
- *Synthesis*: converting (untimed) dataflow graphs into (timed) clocked circuits; fast partitioning algorithms for hardware/software codesign [6]

1 Over the course of the RASSP contract, the Ptolemy team has been directed by Prof. Edward A. Lee and co-directed by Prof. David Messerschmitt. Ptolemy software development was managed first by Alan Kamas and later by Christopher X. Hylands. Dr. Joseph T. Buck is the primary designer of the Ptolemy software architecture. The technical staff has included Dr. Brian L. Evans, Dr. Alain Girault, Dr. Takashi Miyazaki, Dr. H. John Reekie, Dr. Juergen Teich, and several industrial scholars. The Ptolemy team also consists of Dr. Shuvra S. Bhattacharyya, Wan-Teh Chang, William Chen, Kang Ngee Chia, Stephen A. Edwards, Ron Galicia, Michael Goodwin, Steve X. Gu, Dr. Soonhoi Ha, Sangjin Hong, Farhad Jalilvand, Asawaree Kalavade, Karim P. Khiar, Joel King, Allen Lao, Bilung Lee, William Li, Xiao Mei, Praveen K. Murthy, Dr. Thomas M. Parks, José Luis Pino, Farhana Shiekh, S. Sriram, Matthew Tavis, Warren W. Tsai, William Tsu, Patrick J. Warner, and Michael C. Williamson.

We test our ideas in the Ptolemy software environment. Ptolemy provides a framework for mixing tools with fundamentally different semantics and a laboratory for experimenting with these mixtures. Many different kinds of tools cooperate in the specification, simulation, and synthesis of systems. We mix our own custom tools with many existing ones to create a system-level design tool:

- *Block diagram specification*: octtools design database; vem schematic editor; Ptolemy higher-order functions for scalable systems
- *Parameter specification*: Ptolemy expression evaluator; Tcl; MATLAB; Mathematica; Tycho
- *Simulation*: Ptolemy dataflow, discrete-event, and finite state machine domains; Ptolemy kernel; MATLAB; Esterel; Tcl/Tk; Synopsys VHDL System Simulator; Model Technology VSIM VHDL Simulator; xgraph; xv; soundtool; Utah Raster Toolkit; GNU threads
- *Software synthesis*: Ptolemy C, C++, and DSP assembly code generators; GNU and cfront C and C++ compilers; make; Motorola 56000 assemblers and simulators
- *Hardware synthesis*: Ptolemy VHDL generation synthesized by Synopsys Design Analyzer.

## 4. Deliverables

During the RASSP contract, we published ninety papers and two books, released five versions of the Ptolemy software, initiated a news group `comp.soft-sys.ptolemy`, and created a Web site `http://ptolemy.eecs.berkeley.edu/` for rapid dissemination of papers, software, documentation, and tutorials. Several other RASSP participants, such as Sanders, MIT/BU, and BDTI, have leveraged the Ptolemy software environment for rapid prototyping. Cadence has commercialized our ideas on dataflow modeling and scheduling in their SPW tool and our ideas on heterogeneous simulation in their CONVERGENCE architecture to allow the SPW and Bones simulators to cooperate.

The Ptolemy Project is broader than the RASSP contract. The Ptolemy Web site has a tutorial on the TI C30 DSP processor, a C code documentation extractor, and Signal Processing Packages for Mathematica, which Wolfram Research Inc. commercialized as the *Signals and Systems Pack*.

## 5. Selected References

[1] B. L. Evans, S. X. Gu, A. Kalavade, and E. A. Lee, “Symbolic Computation in System Simulation and Design,” Invited Paper, *Proc. of SPIE Int. Sym. on Advanced Signal Processing Algorithms, Architectures, and Implementations*, July 9-16, 1995, San Diego, CA, pp. 396-407.

[2] P. K. Murthy and E. A. Lee, “An Extension of Multidimensional Synchronous Dataflow to Handle Arbitrary Sampling Lattices,” *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Atlanta, GA, May 7-10, 1996, vol. 6, pp. 3306-3309.

[3] Shuvra S. Bhattacharyya, Praveen K. Murthy, and Edward A. Lee, *Software Synthesis from Dataflow Graphs*, Kluwer Academic Press, ISBN 0-7923-9722-3, April, 1996.

[4] J. L. Pino, M. C. Williamson, and E. A. Lee, “Interface Synthesis in Heterogeneous System-Level DSP Design Tools,” *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Atlanta, GA, May 7-10, 1996, vol. 2, pp. 1268-1271.

[5] W.-T. Chang, S.-H. Ha, and E. A. Lee, “Heterogeneous Simulation-- Mixing Discrete-Event Models with Dataflow,” Invited Paper, *Journal on VLSI Signal Proc.*, RASSP special issue, to appear.

[6] A. Kalavade and E. A. Lee, “The Extended Partitioning Problem: Hardware/Software Mapping and Implementation-Bin Selection,” *Proc. IEEE Int. Work. on Rapid Sys. Prototyping*, June 1995, pp. 12-18.