DESIGN METHODOLOGY FOR DSP

Edward A. Lee, Principal Investigator

Department of Electrical Engineering and Computer Science
University of California, Berkeley CA 94720

Final Report 1994-95, Micro Project #94-075

Industrial Sponsors: Bell Northern Research, Dolby Laboratories,
Mentor Graphics, Philips Research, and Rockwell International.

ABSTRACT

The focus of this project is on design methodology for complex real-time systems where a variety of design methodologies and implementation technologies must be combined. Design methodologies are encapsulated in one or more models of computation, while implementation technologies are implemented as synthesis tools. Applications that use more than one model of computation and/or more than one synthesis tool are said to be heterogeneous. Hardware/software codesign is one example of heterogeneous design. Project results have been disseminated via the *Ptolemy* software system, in addition to papers. The overall Ptolemy project is fairly large, with additional support from ARPA, SRC, and a number of other companies, and is strongly collaborative. The MICRO portion of the project has focused on real-time signal processing, although the larger project is broader.

1. THE CONTEXT

The objectives of the Ptolemy Project include most aspects of designing signal processing and communications systems, ranging from designing and simulating algorithms to synthesizing hardware and software, parallelizing algorithms, and prototyping real-time systems. Research ideas developed in the project are implemented and tested in the Ptolemy software environment. The Ptolemy software environment, which serves as our laboratory, is a system-level design framework that allows mixing models of computation and implementation languages.

In designing digital signal processing and communications systems, often the best available design tools are domain specific. The tools must be able to interact. Ptolemy allows the interaction of diverse models of computation by using the object-oriented principles of polymorphism and information hiding. For example, using Ptolemy, a high-level dataflow model of a signal processing system can be connected to a hardware simulator that in turn may be connected to a discrete-event model of a communication network.

A part of the Ptolemy project concerns programming methodologies commonly called “graphical dataflow programming” that are used in industry for signal processing and experimentally for other applications. By “graphical” we mean simply that the program is explicitly specified by a directed graph where the nodes represent computations and the arcs represent streams of data. The graphs are typically hierarchical, in that a node in a graph may represent another directed graph. In Ptolemy the nodes in the graph are subprograms specified in C++.

It is common in the signal processing community to use a visual syntax to specify such graphs, in which case the model is often called “visual dataflow programming.” But it is by no means essential to use a visual syntax.

Hierarchy in graphical program structure can be viewed as an alternative to the more usual abstraction of subprograms via procedures, functions, or objects. It is better suited than any of these to a visual syntax, and also better suited to signal processing.

Some other examples of graphical dataflow programming environments intended for signal processing (including image processing) are Khoros, from the University of New Mexico (now distributed by Khoral Research, Inc.), the signal processing worksystem (SPW), from the Alta Group at Cadence (formerly Comdisco Systems), COSSAP, from Synopsys (formerly Cadis), and the DSP Station, from Mentor Graphics. MATLAB from The MathWorks, which is popular for signal processing and other applications, has a visual interface called SIMULINK. These software environments all claim variants of dataflow semantics.

Most graphical signal processing environments do not define a language in a strict sense. In fact, some designers of such environments advocate minimal semantics, arguing that the graphical organization by itself is sufficient to be useful. The semantics of a program in such environments is determined by the contents of the graph nodes, either subgraphs or subprograms. Subprograms are usually specified in a conventional programming language such as C. Most such environments, however, including Khoros, SPW, and COSSAP, take a middle ground, permitting the nodes in a graph or subgraph to contain arbitrary subprograms, but defining precise semantics for the interaction between nodes. We call the language used to define the subprograms in nodes the *host language*. We call the language defining the interaction between nodes the *coordination language*.

Many possibilities have been explored for precise semantics of coordination languages. Many of these limit expressiveness in exchange for considerable advantages such as compile-time predictability. In Ptolemy, a *domain* defines the semantics of a coordination language, but domains are modular objects that can be mixed and matched at will. Thus we gain flexibility without the sloppiness of unspecified semantics in the coordination language.

Graphical programs can be either interpreted or compiled. It is common in signal processing environments to provide both options. The output of compilation can be a standard procedural language, such as C, assembly code for programmable DSP processors, or even specifications of silicon implementations. We have put considerable effort into optimized compilation.

2. RESULTS OF MICRO SUPPORT

2.1. Methodology

We have made progress on several different models of computation and their interaction. In dataflow modeling, we have formalized relationships among actors using partial orders and characterized the various types of dataflow in terms of their mathematical properties. We have also developed a new efficient robust scheduler for dynamic dataflow and investigated efficient dataflow representations for multidimensional multirate systems. The Process Networks model is a superset of the dataflow models and schedules blocks as processes (threads) under the control of a multitasking kernel. In other areas of modeling, we have researched formal models for characterizing hierarchical finite state machine (FSM) controllers. For interaction between models of computation, we have studied the mixing of FSM controllers with dataflow models, FSM controllers with discrete-event models, and dataflow models with discrete-event models.

In implementation, we have made fundamental contributions in optimized synthesis of embedded software on both uniprocessor and multiprocessor architectures. For uniprocessor code generation, we can trade off program size, data size, and throughput. For parallel code generation, we have made progress in optimizing synchronized communication between processors. We have also made contributions in hierarchical scheduling and incremental compilation and developed methodologies for converting (untimed) models of computation into (timed) clocked circuits implemented in VHDL. And we have made progress in integrated multi-lingual design, where for example a subsystem specified in VHDL can interact with one specified in C, which in turn can interact with another specified in assembly code for a DSP.

In hardware/software codesign, we have derived an efficient algorithm to partition dataflow graphs into a combined hardware/software implementation. We have also made progress in design methodology management by defining an abstraction for capturing the dependencies between the data, design representations, and tools used in a complex design process. Having a formal design model allows heterogeneous design styles to cooperate in a constructive and predictable manner.

2.2. The Ptolemy software

We have successfully tested many theoretical ideas in modeling, implementation, system design, and user interfaces in the Ptolemy software environment. We have distributed two new releases of the software, which runs on ten different Unix architectures. These releases contain many improvements to the Ptolemy interactive graphical interface, including highly interactive, animated simulation capabilities. The GUI now supports a visual syntax for representing recursion and scalable systems, based on higher-order functions. During simulation, Ptolemy can now cooperate with MATLAB and Mathematica. In synthesis, Ptolemy can get feedback about hardware implementation cost from the high-level synthesis tools Hyper for Silage specifications and Synopsys tools for VHDL specifications.

We have increased the visibility of the Ptolemy Project and Ptolemy software environment. We have made a wealth of information available on our web server http://ptolemy.eecs.berkeley.edu, including on-line demonstrations and searchable hypertext versions of all four volumes of documentation for the software environment. We have a Usenet news group called comp.soft-sys.ptolemy. We gave a Ptolemy Mini-conference at U.C. Berkeley in March of 1995 that was attended by 50 people from 21 companies and three government agencies. We also conducted a full-day Ptolemy tutorial in Washington, D.C., in July of 1995. We have placed the transparencies from both all-day events on our web site.

The number of organizations actively using and extending Ptolemy continues to grow. It is used by a number of participants of the ARPA RASSP program (Rapid Prototyping of Application-Specific Signal Processors) as well as a number of companies and universities. We believe that it continues to provide an effective laboratory for research in design methodology. Moreover, it has been having considerable influence on commercial products. The Alta Group of Cadence Design Systems acknowledges considerable influence from Ptolemy in their latest software release, SPW 3.5. We believe that by influencing commercial products that are used by many of our sponsors, we enhance their return on investment.

Ptolemy 0.5.2 is the latest release. It consist of approximately 2000 files containing 300,000 lines and 8 Mb of source code, with executables distributed for the following architectures: Sun Sparc (SunOS and Solaris), HP (HP/UX), and SGI (Irix) computers. Releases are also contributed by outside users for the Dec Alpha (Ultrix), PC (Linux), IBM RS/6000 (AIX), and Power PC (AIX) computers. The source code has also been ported to the DecStation (Ultrix) and PC (NetBSD) computers. Ptolemy 0.6 is due out in April of 1996.

2.3. Plans

The top priority topics that we will be addressing over the next year are:

- hierarchical syntax management,
- hardware synthesis from dataflow graphs,
- mixed control and dataflow,
- interaction of synchronous languages and dataflow,
- “cyclo-static” dataflow model for scheduling,
- redesigned image processing infrastructure,
- integrated symbolic computation,
- graphical design, and
- automated regression tests for the software.

3. PUBLICATIONS — July 1994 - December 1995

[1] T. M. Parks, *Bounded Scheduling of Process Networks*, Technical Report UCB/ERL-95-105. **PhD Dissertation**. EECS Department, University of California. Berkeley, CA 94720, December 1995.

[2] Sundararajan Sriram, *Minimizing Communication and Synchronization Overhead in Multiprocessors for Digital Signal Processing*, Tech. Report UCB/ERL 95/90, **Ph.D. Dissertation**, Dept. of EECS, University of California, Berkeley, CA 94720, November 7, 1995

[3] J. L. Pino, S. S. Bhattacharyya and E. A. Lee, “A Hierarchical Multiprocessor Scheduling System for DSP Applications,”*Proc. IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 29 -

November 1, 1995. (http://ptolemy.eecs.berkeley.edu/~pino/papers/asilomar-95)

[4] T. M. Parks, J. L. Pino, and E. A. Lee, "A Comparison of Synchronous and Cyclo-Static Dataflow," *Proc. IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 29 - November 1, 1995.

[5] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Optimal Parenthesization of Lexical Orderings for DSP Block Diagrams," in *Proc. IEEE Workshop on VLSI Signal Processing*, Osaka, Japan, October 16-18, 1995. (http://ptolemy.eecs.berkeley.edu/papers/PganRpmcDppo/).

[6] S. S. Bhattacharyya, S. Sriram, and E. A. Lee, "Resynchronization for Embedded Multiprocessors," ERL Technical Report UCB/ERL M95/70, University of California, Berkeley, CA 94720, September, 1995. (http://ptolemy.eecs.berkeley.edu/papers/synchOpt/).

[7] Asawaree Kalavade, *System Level Codesign of Mixed Hardware-Software Systems*, Tech. Report UCB/ERL 95/88, **Ph.D. Dissertation**, Dept. of EECS, University of California, Berkeley, CA 94720, September, 1995.

[8] B. L. Evans, D. R. Firth, K. D. White, and E. A. Lee, "Automatic Generation of Programs That Jointly Optimize Characteristics of Analog Filter Designs," *Proc. of European Conference on Circuit Theory and Design*, August 27-31, 1995, Istanbul, Turkey, pp. 1047-1050.

[9] 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.

[10] S. S. Bhattacharyya, S. Sriram, and E. A. Lee, "Minimizing Synchronization Overhead in Statically Scheduled Multiprocessor Systems," *Proc. of IEEE Int. Conference on Application Specific Array Processors*, July 24-26, 1995. (http://ptolemy.eecs.berkeley.edu/papers/synchOpt/).

[11] W.-T. Chang, A. Kalavade, and E. A. Lee, "Effective Heterogeneous Design and Cosimulation,", NATO Advanced Study Institute Workshop on Hardware/Software Codesign, Lake Como, Italy, June 18 — 30, 1995 (http://ptolemy.eecs.berkeley.edu/papers/effective).

[12] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Converting Graphical DSP Programs into Memory-Constrained Software Prototypes," *Proc. of IEEE Int. Workshop on Rapid Systems Prototyping*, Chapel Hill, NC, June 7-9, 1995 (http://ptolemy.eecs.berkeley.edu/papers/PganRpmcDppo/).

[13] A. Kalavade and E. A. Lee, "The Extended Partitioning Problem: Hardware/Software Mapping and Implementation-Bin Selection," *Proc. of IEEE Int. Workshop on Rapid Systems Prototyping*, Chapel Hill, NC, June 7-9, 1995 (http://ptolemy.eecs.berkeley.edu/papers/extended_partitioning/).

[14] E. A. Lee and T. M. Parks, "Dataflow Process Networks, *Proceedings of the IEEE*, vol. 83, no. 5, pp. 773-801, May, 1995. (http://ptolemy.eecs.berkeley.edu/papers/processNets)

[15] J. L. Pino, S.S. Bhattacharyya and E. A. Lee, *A Hierarchical Multiprocessor Scheduling Framework for Synchronous Dataflow Graphs*, UCB/ERL M95/36, May 30, 1995.(http://ptolemy.eecs.berkeley.edu/papers/erl-95-36)

[16] R. H. Bamberger, B. L. Evans, E. A. Lee, J. H. McClellan, and M. A. Yoder, "Integrating Layout, Analysis, and Simulation Tools in Electronic Courseware for Teaching Signal Processing," Invited Paper, *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 8-12, 1995, Detroit, MI, pp. 2873-2876 (http://ptolemy.eecs.berkeley.edu/papers/education).

[17] A. Kalavade, J. L. Pino, and E. A. Lee, "Managing Complexity in Heterogeneous Specification, Simulation, and Synthesis," Invited Paper, *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 8-12, 1995, Detroit, MI, pp. 2833-2836 (http://ptolemy.eecs.berkeley.edu/papers/managing_complexity).

[18] K. Khiar and E. A. Lee, "Modeling Radar Systems Using Hierarchical Dataflow," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 8-12, 1995, pp. 3259-3262. (available locally) (http://ptolemy.eecs.berkeley.edu/papers/Radarsimu.ps.Z)

[19] T. M. Parks and E. A. Lee, ""Non Preemptive Real-Time Scheduling of Dataflow Systems," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 8-12, 1995, pp. 3235-3238. (available locally) (http://ptolemy.eecs.berkeley.edu/papers/real_time).

[20] J. L. Pino and E. A. Lee, "Hierarchical Static Scheduling of Dataflow Graphs onto Multiple Processors," *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 8-12, 1995, pp. 2643-2646 (http://ptolemy.eecs.berkeley.edu/papers/hierStaticSched). (available locally)

[21] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Renesting Single Appearance Schedules to Minimize Buffer Memory", UCB/ERL Technical Memorandum UCB/ERL M95/43, Electronics Research Lab., UC Berkeley, CA 94720, April, 1 1995. (http://ptolemy.eecs.berkeley.edu/papers/PganRpmcDppo/).

[22] P. K. Murthy, and E. A. Lee, "A Generalization of Multidimensional Synchronous Dataflow to Handle Arbitrary Sampling Lattices", ERL Technical Report UCB/ERL M95/59, Electronics Research Lab, UC, Berkeley, CA 94720, March 20, 1995. (http://ptolemy.eecs.berkeley.edu/papers/genMdsdf/).

[23] S. S. Bhattacharyya, J. T. Buck, S. Ha, and E. A. Lee, "Generating Compact Code from Dataflow Specifications of Multirate Signal Processing Algorithms", *IEEE Trans.*

on Circuits and Systems I: Fundamental Theory and Applications, vol. 42, no. 3, pp. 138-150, March 1995.

[24] J. L. Pino, S. Ha, E. A. Lee, and J. T. Buck, "Software Synthesis for DSP Using Ptolemy," *Journal on VLSI Signal Processing*, vol. 9, no. 1, pp. 7-21, Jan., 1995 (http://ptolemy.eecs.berkeley.edu/papers/jvsp_codegen).

[25] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Two Complementary Heuristics for Translating Graphical DSP Programs into Minimum Memory Software Implementations," Memorandum No. UCB/ERL M95/3, Electronics Research Laboratory, University of California, Berkeley, CA 94720, January 10, 1995. (http://ptolemy.eecs.berkeley.edu/papers/PganRpmcDppo/)

[26] S. S. Bhattacharyya, S. Sriram, and E. A. Lee, "Optimizing Synchronization in Multiprocessor Implementations of Iterative Dataflow Programs," ERL Technical Report UCB/ERL M95/2, University of California, Berkeley, CA 94720, January 5, 1995. (http://ptolemy.eecs.berkeley.edu/papers/synchOpt/). Also, to appear in *IEEE Tr. on Signal Processing*.

[27] P. K. Murthy, S. S. Bhattacharyya, and E. A. Lee, "Combined Code and Data Minimization for Synchronous Dataflow Programs," Memorandum No. UCB/ERL M94/93, Electronics Research Laboratory, University of California, Berkeley, CA 94720, November 29, 1994. (http://ptolemy.eecs.berkeley.edu/papers/jointCodeDataMinimize).

[28] J. T. Buck, "Static Scheduling and Code Generation from Dynamic Dataflow Graphs with Integer-Valued Control Systems," Invited Paper, *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994. (http://ptolemy.eecs.berkeley.edu/papers/IDF_Asilomar.ps.Z)

[29] M. J. Chen and E. A. Lee, "Design and Implementation of a Multidimensional Synchronous Dataflow Environment," Invited Paper, *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994. (http://ptolemy.eecs.berkeley.edu/papers/mdsdf_asilomar.ps.Z)

[30] B. L. Evans and J. H. McClellan, "Algorithms for Symbolic Linear Convolution," *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994, pp. 948-953. (http://ptolemy.eecs.berkeley.edu/papers/symbolic_convolution)

[31] B. L. Evans, S. X. Gu, and R. H. Bamberger, "Interactive Solution Sets as Components of Fully Electronic Signals and Systems Courseware," *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994, pp. 1314-1319. (http://ptolemy.eecs.berkeley.edu/papers/education)

[32] B. L. Evans, J. Teich, and T. A. Kalker, "Families of Smith Form Decomposition to Simplify Multidimensional Filter Bank Design," *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994, pp. 363-367. (http://ptolemy.eecs.berkeley.edu/papers/mD_filter_bank_design)

[33] P. K. Murthy and E. A. Lee, "Optimal Blocking Factors for Blocked, Non-Overlapped Multiprocessor Schedules", Invited Paper, *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994. (http://ptolemy.eecs.berkeley.edu/papers/blocFac.ps.Z)

[34] J. L. Pino, T. M. Parks and E. A. Lee, "Mapping Multiple Independent Synchronous Dataflow Graphs onto Heterogeneous Multiprocessors," *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 31 - Nov. 2, 1994. (http://ptolemy.eecs.berkeley.edu/papers/multiIndepGraph)

[35] S. Sriram and E. A. Lee, "Statically Scheduling Communication Resources in Multiprocessor DSP Architectures," Invited Paper, *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994. (http://ptolemy.eecs.berkeley.edu/papers/ordering_asilomar94.ps.Z)

[36] J. Teich, S. Sriram, L. Thiele, and M. Martin, "Performance Analysis of Mixed Asynchronous-Synchronous Systems", *Proc. of the IEEE Workshop on VLSI Signal Processing*, Oct. 26 - 28, 1994, pp. 103-112. Proceedings published as IEEE VLSI Signal Processing VII (http://ptolemy.eecs.berkeley.edu/papers/wvsp94.ps.Z)

[37] A. Kalavade and E. A. Lee, "A Global Criticality / Local Phase Driven Algorithm for the Constrained Hardware/Software Partitioning Problem," *Proc. of Codes/CASHE 94, Third IEEE International Workshop on Hardware/Software Codesign*, Grenoble, France, Sept. 22-24, 1994, pp 42-48. (http://ptolemy.eecs.berkeley.edu/papers/codesign)

[38] J. T. Buck, "A Dynamic Dataflow Model Suitable for Efficient Mixed Hardware and Software Implementations of DSP Applications,"" *Proc. of Codes/CASHE 94, Third International Workshop on Hardware/Software Codesign*, Grenoble, France, Sept. 22-24, 1994. (http://ptolemy.eecs.berkeley.edu/papers/DDF_codesign.ps.Z)

[39] B. L. Evans, A. Kamas, and E. A. Lee, "Design and Simulation of Heterogeneous Systems Using Ptolemy," *First Annual Rapid Prototyping of Application Specific Signal Processors (RASSP) Conference*, Arlington, VA, Aug. 15-18, 1994, pp. 97-105. (http://ptolemy.eecs.berkeley.edu/papers/heterogeneous)

[40] S. S. Bhattacharyya, *Compiling Dataflow Programs for Digital Signal Processing*, Tech. Report UCB/ERL 94/52, **Ph.D. Dissertation**, Dept. of EECS, University of California, Berkeley, CA 94720, July 12, 1994. (http://ptolemy.eecs.berkeley.edu/papers/compiling_dataflow.tar.Z)