DESIGN METHODOLOGY FOR DSP

Edward A. Lee, Principal Investigator

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

Final Report 1993-94, Micro Project #93-092

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

ABSTRACT

This ongoing project explores design methodology for simulation and real-time computation. The goal is to facilitate rapid prototyping by developing tools that are both efficient in their use of hardware and easy for an algorithm designer to learn and use. Project results have been disseminated via the *Ptolemy* software system, in addition to papers. Ptolemy is currently distributed by anonymous FTP and through our Industrial Liaison Program. It is also used as an integral part of our graduate courses in statistical signal processing (EE225a) and digital communications (EE224), our undergraduate signal processing course (EE123), and a research seminar (EE290T) investigating languages and design methodology for signal processing systems. 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 traditionally focused on real-time signal processing, although the larger project is broader.

1. THE CONTEXT

A large 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. A few graphical programming environments allow an arbitrary mixture of visual and textual specification, both based on the same language. For example, the Signal [38], Lustre [23], and Silage [24] languages all have a visual and a textual syntax, the latter available in the commercial Mentor Graphics DSP Station as DFL. Other languages with related semantics, such as Sisal [39], are used primarily or exclusively with textual 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 [50] (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. A survey of graphical dataflow languages for other applications is given by Hills [25]. These software environments all claim variants of dataflow semantics, but a word of caution is in order. The term "dataflow" is often used loosely for semantics that bear little resemblance to those outlined by Dennis in 1975 [13]. The relationship between such languages and functional languages and each other is studied in detail in [37].

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, including for example the computation graphs of Karp and Miller [31], the synchronous dataflow graphs of Lee and Messerschmitt, the cyclo-static dataflow of Lauwereins, *et al.* [33], the Processing Graph Method (PGM) of Kaplan, *et al.* [30], and many others. 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 [48], or even specifications of silicon implementations [12]. We have put considerable effort into optimized compilation (see publications overview below).

2. RESULTS OF MICRO SUPPORT

Our contributions fall into two overlapping categories: methodology and software.

2.1. The Ptolemy software

The latest version of Ptolemy is designated 0.5.1. It is a "multi-paradigm" and fully extensible design environment with both a visual ("block diagram") and textual interface. A number of models of computation have been built into it already, and experimentation with others continues. The host language is C++, although C, FORTRAN, and Tcl/Tk can all be used as well.

Ptolemy supports two distinct execution models, *interpreted* and *compiled*. Compilation in current domains is implemented using a simple code generation mechanism that simply stitches together code fragments defining each module [48].

We have two principal uses for Ptolemy. First, it is the laboratory within which we conduct our experiments in design methodology. Second, it is a vehicle for disseminating our results. All source code is distributed with minor restrictions on re-use (attribution and disclaimer). Of course, fundamental results are also published in more conventional forms.

2.2. Overview of recent publications

Between January 1994 and the present, the project produced

- 1 Ph.D. thesis,
- 3 masters theses,
- 1 patent,
- 5 journal papers published and 3 accepted,
- 16 conference papers published and 5 accepted, and
- 8 technical reports.

Our basic paper on the Ptolemy kernel has finally appeared in print after an absurd delay [9], and we continue to occasionally present general overviews of the project [17].

Quite a few of the publications pertain to synthesis of optimized embedded software from dataflow graphs [3][4][5][6][8][40][41][42][43].

A major accomplishment was the unification of dataflow with process network theory and functional languages [37]. This led to the application of higher-order functions to visual dataflow programming and system-level design. We have applied these techniques to a large radar signal processing problem [32] as well as to a number of smaller problems and have developed a process networks (PN) domain in Ptolemy.

We were granted a patent on our ordered transaction architecture, which dramatically reduces the cost of interprocessor communication for embedded systems [34]. A side effect of the ordered transactions principle is that it constrains the sequence in which interprocessor communications can occur. However, we have identified a procedure for ensuring that the performance penalty arising from these additional constraints is negligible [53]. Spurred in part by this work, we have also found methods for systematically removing unnecessary synchronizations in multiprocessor implementations [7].

We made a number of methodology advances:

- in real-time computing [44],
- in code generation [48],
- in mixed synchronous/asynchronous systems [54],
- in hardware/software codesign [26][27][28], and
- in heterogeneous scheduling and code generation [46][47][49].

We implemented an experimental multidimensional dataflow domain in Ptolemy [10][11] and a code generation mechanism for a real-time video signal processor [52].

Management of the design flow has been a major focus [29], and has led to the development of a "design methodology management" (DMM) domain in Ptolemy.

The use of symbolic computation (based on Mathematica) within Ptolemy will be reported in [22].

We have put considerable effort into carrying advanced design methodology techniques into the classroom. This includes both our fundamental work in models of computation [35] and our work in symbolic processing [1][19].

Fundamental applications work has been primarily concerned with multidimensional signal processing [51][14][15][16][18][20][21].

3. REFERENCES

[1] 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, **to appear** *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May, 1995, Detroit, MI.

[2] 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.

[3] S. S. Bhattacharyya and E. A. Lee, "Memory Management for Dataflow Programming of Multirate Signal Processing Algorithms," *IEEE Trans. on Signal Processing*, vol. 42, no. 5, May 1994.

[4] S. S. Bhattacharyya and E. A. Lee, "Looped Schedules for Dataflow Descriptions of Multirate Signal Processing Algorithms," *Formal Methods in System Design*, No. 5, No. 3, December, 1994.

[5] S. S. Bhattacharyya, P. K. Murthy, and E. A. Lee, "Converting Graphical DSP Programs into Memory-Constrained Software Prototypes," **to appear** in *Proc. of IEEE Int.Wkshp. on Rapid Systems Prototyping*, Chapel Hill, NC, June, 1995.

[6] S. S. Bhattacharyya, J. T. Buck, S. Ha, and E. A. Lee, "Generating Compact Code from Dataflow Specifications of Multirate Signal Processing Algorithms", **to appear** in

IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications.

[7] 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, January 5, 1995.

[8] 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, January 10, 1995.

[9] J. T. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: A Framework for Simulating and Prototyping Heterogeneous Systems," *Int. Journal of Computer Simulation*, special issue on "Simulation Software Development," vol. 4, pp. 155-182, April, 1994.

[10] M. J. Chen, "Developing a Multidimensional Synchronous Dataflow Domain in Ptolemy," **MS Report**, ERL Technical Report UCB/ERL No. 94/16, University of California, Berkeley, CA 94720, May 6, 1994.

[11] 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.

[12] H. De Man, F. Catthoor, G. Goossens, J. Vanhoof, J. Van Meerbergen, S. Note, J. Huisken, "Architecture-driven synthesis techniques for mapping digital signal processing algorithms into silicon," *Proceedings of the IEEE*, Vol. 78, No. 2, pp. 319-335, February, 1990.

[13] J.B. Dennis, "First Version Data Flow Procedure Language", Technical Memo MAC TM61, May, 1975, MIT Laboratory for Computer Science.

[14] B. L. Evans T. R. Gardos, and J. H. McClellan, "Imposing Structure on Smith Form Decompositions of Rational Resampling Matrices", *IEEE Trans. on Signal Processing*, vol. 42, no. 4, pp. 970-973, April, 1994.

[15] B. L. Evans, R. H. Bamberger, and J. H. McClellan, "Rules for Multidimensional Multirate Structures", *IEEE Trans. on Signal Processing*, vol. 42, no. 4, pp. 762-771, April, 1994.

[16] B. L. Evans and F. A. Sakarya, "Interactive Graphical Design of Two- Dimensional Compression Systems", *Proc. of Second National Workshop on Signal Processing*, pp. 173-178, Marmaris, Turkey, April, 1994.

[17] 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.

[18] 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.

[19] 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.

[20] B. L. Evans, J. Teich, and C. Schwarz, "Automated Design of Two-Dimensional Rational Decimation Systems," *Proc. of IEEE Asilomar Conf. on Signals, Systems, and Computers*, Oct. 31 - Nov. 2, Pacific Grove, CA, 1994, pp. 498-502.

[21] 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.

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

[23] N. Halbwachs, P. Caspi, P. Raymond, D. Pilaud, "The Synchronous Data Flow Programming Language LUSTRE," *Proceedings of the IEEE*, Vol. 79, No. 9, 1991.

[24] P. Hilfinger, "A High-Level Language and Silicon Compiler for Digital Signal Processing", Proceedings of the *Custom Integrated Circuits Conference*, IEEE Computer Society Press, Los Alamitos, CA 1985, pp 213-216.

[25] D. D. Hills, "Visual Languages and Computing Survey: Data Flow Visual Programming Languages," *J. of Visual Languages and Computing*, Vol. 3, p. 69-101.

[26] A. Kalavade and E. A. Lee, "Manifestations of Heterogeneity in Hardware/Software Codesign," *Proc. of Design Automation Conference*, San Diego, CA, June, 1994, pp. 437-438.

[27] A. Kalavade and E. A. Lee, "A Methodology for the Simulation and Synthesis of Mixed Hardware/Software Systems," October 1994, UCB Tech. Report.

[28] 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 International Workshop on Hardware/Software Codesign*, Grenoble, France, Sept. 22-24, 1994, pp 42-48.

[29] A. Kalavade, J. L. Pino, and E. A. Lee, "Managing Complexity in Heterogeneous Specification, Simulation, and Synthesis," Invited Paper, **to appear** *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May, 1995, Detroit, MI.

[30] D. J. Kaplan, *et al.*, "Processing Graph Method Specification Version 1.0," Unpublished Memorandum, The Naval

Research Laboratory, Washington D.C., December 11, 1987.

[31] R. M. Karp, R. E. Miller, "Properties of a Model for Parallel Computations: Determinacy, Termination, Queueing," *SIAM Journal*, Vol. 14, pp. 1390-1411, November, 1966.

[32] K. Khiar and E. A. Lee, "Modeling Radar Systems Using Hierarchical Dataflow," **to appear** in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995.

[33] R. Lauwereins, P. Wauters, M. Adé, J. A. Peperstraete, "Geometric Parallelism and Cyclo-Static Dataflow in GRAPE-II", *Proc. 5th Int. Workshop on Rapid System Prototyping*, Grenoble, France, June, 1994.

[34] E. A. Lee and J. Bier, "Multiprocessor System Having Statically Determining Resource Allocation Schedule at Compile Time and the Using o Static Schedule With Processor Signals To Control The Execution Time Dynamically", UNITED STATES PATENT, Number 5,367,678, Nov. 22, 1994

[35] E. A. Lee, "Computing and Signal Processing: An Experimental Multidisciplinary Course", *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. VI, pp. 45-48, Adelaide, Australia, April, 1994.

[36] E. A. Lee and D. G. Messerschmitt, "Synchronous Data Flow," *IEEE Proceedings*, September, 1987.

[37] E. A. Lee and T. M. Parks, "Dataflow Process Networks," to appear in *Proceedings of the IEEE*, May 1995.

[38] P. Le Guernic, T. Gauthier, M. Le Borgne, C. Le Maire, "Programming Real-Time Applications with SIGNAL," *Proceedings of the IEEE*, Vol. 79, No. 9, September 1991.

[39] J. McGraw, "Sisal: Streams and Iteration in a Single Assignment Language", *Language Reference Manual*, Lawrence Livermore National Laboratory, Livermore, CA.

[40] P. K. Murthy, S. Bhattacharyya, and E. A. Lee, "Minimizing Memory Requirements For Chain-Structured Synchronous Dataflow Programs," *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. II, pp. 453-456, Adelaide, Australia, April, 1994.

[41] P. K. Murthy and E. A. Lee, "On the Optimal Blocking Factor for Blocked, Non-Overlapped Schedules," ERL Technical Report UCB/ERL 94/46, University of California, Berkeley, CA 94720, June 6, 1994.

[42] 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.

[43] 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, November 29, 1994.

[44] T. M. Parks and E. A. Lee, "Non Preemptive Real-Time Scheduling of Dataflow Systems," **to appear** in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995.

[45] A. Peevers, *A Real-Time 3D Signal Analysis/Synthesis Tool Based on the Overlap-Add Short-Time Fourier Transform*, **MS Report**, Plan II, University of California, Berkeley, February 24, 1994.

[46] J. L. Pino, T. M. Parks, and E. A. Lee, "Automatic Code Generation for Heterogeneous Multiprocessors," *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Process*ing, vol. II, pp. 445-448, Adelaide, Australia, April, 1994.

[47] 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.

[48] J. L. Pino, S. Ha, E. A. Lee, J. T. Buck, "Software Synthesis for DSP Using Ptolemy", *Journal of VLSI Signal Processing*, Vol. 9, No. 1, pp 7-21, January 1995.

[49] J. L. Pino and E. A. Lee, "Hierarchical Static Scheduling of Dataflow Graphs onto Multiple Processors," **to appear** in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, MI, May 1995.

[50] J. Rasure and C. S. Williams, "An Integrated Visual Language and Software Development Environment", *Journal of Visual Languages and Computing*, Vol 2, pp 217-246, 1991.

[51] C. Schwarz, J. Teich, Alek Vainshtein, Emo Welzl, and B. L. Evans, "Minimal enclosing parallelogram with applications," *ACM Conf. on Computational Geometry*, June 5-7, 1995, Vancouver, Canada.

[52] S.-I. Shih, "Code Generation for VSP Software Tool in Ptolemy", **MS Report**, Plan II, ERL Technical Report UCB/ERL M94/41, University of California, Berkeley, CA 94720, May 25, 1994.

[53] 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.

[54] 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.