Ptolemy Seamlessly Supports Heterogeneous Design

DEPARTMENT OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCES
UNIVERSITY OF CALIFORNIA AT BERKELEY

RASSP Enterprise Newsletter, Vol. 1, No. 3, August 1994
— EXPANDED VERSION —

BERKELEY, CA — A technology base team at the University of California at Berkeley is developing a software environment called Ptolemy that supports heterogeneous design. In signal processing systems, heterogeneity arises in two ways:

- Diverse implementation technologies are combined (such as hardware, software, or subsystem integration).
- Diverse models of computation are used to describe the system being implemented.

The first of these dominates in later phases of the design, where validation is complicated by the mixture of implementation technologies. The second of these dominates during early phases of system-level design, where domain-specific, high-level tools are most effectively used. The objective of the Ptolemy project is to seamlessly support both forms of heterogeneity through carefully conceived software architecture, and to support seamless migration from system-level design to detailed design.

An example of the way that Ptolemy can support diverse implementation technologies is shown in figure 1. The design of each subsystem is carried out using the best available methodology and tools for that subsystem, at the highest level of abstraction possible. For example, signal processing software can be specified using a block diagram with dataflow semantics, and hardware can be specified at the architecture level as an assemblage of high-level building blocks. The subsystems are then combined for cosimulation and/or cosynthesis in a modular and transparent way.

An example of the way that Ptolemy can support diverse models of computation in a system-level design is shown in figure 2. There, a model of a broadband packet network includes the interactions between three key elements: signal processing (video and audio compression), transport (cell-relay or ATM), and control (signaling and call processing). The signal processing is modeled again using block diagrams with dataflow semantics, while the networking is modeled using discrete-event semantics.

Ptolemy is an object-oriented system and achieves its goals using the principle of polymorphism. The Ptolemy kernel, written in C++, defines the basic classes that allow the components of the system to function together, and from the classes, application-specific objects are derived. A high-level domain-specific design tool is a collection of such objects. Information hiding and data abstraction are key to the design; the system is extensible in many dimensions without the need to modify the kernel.

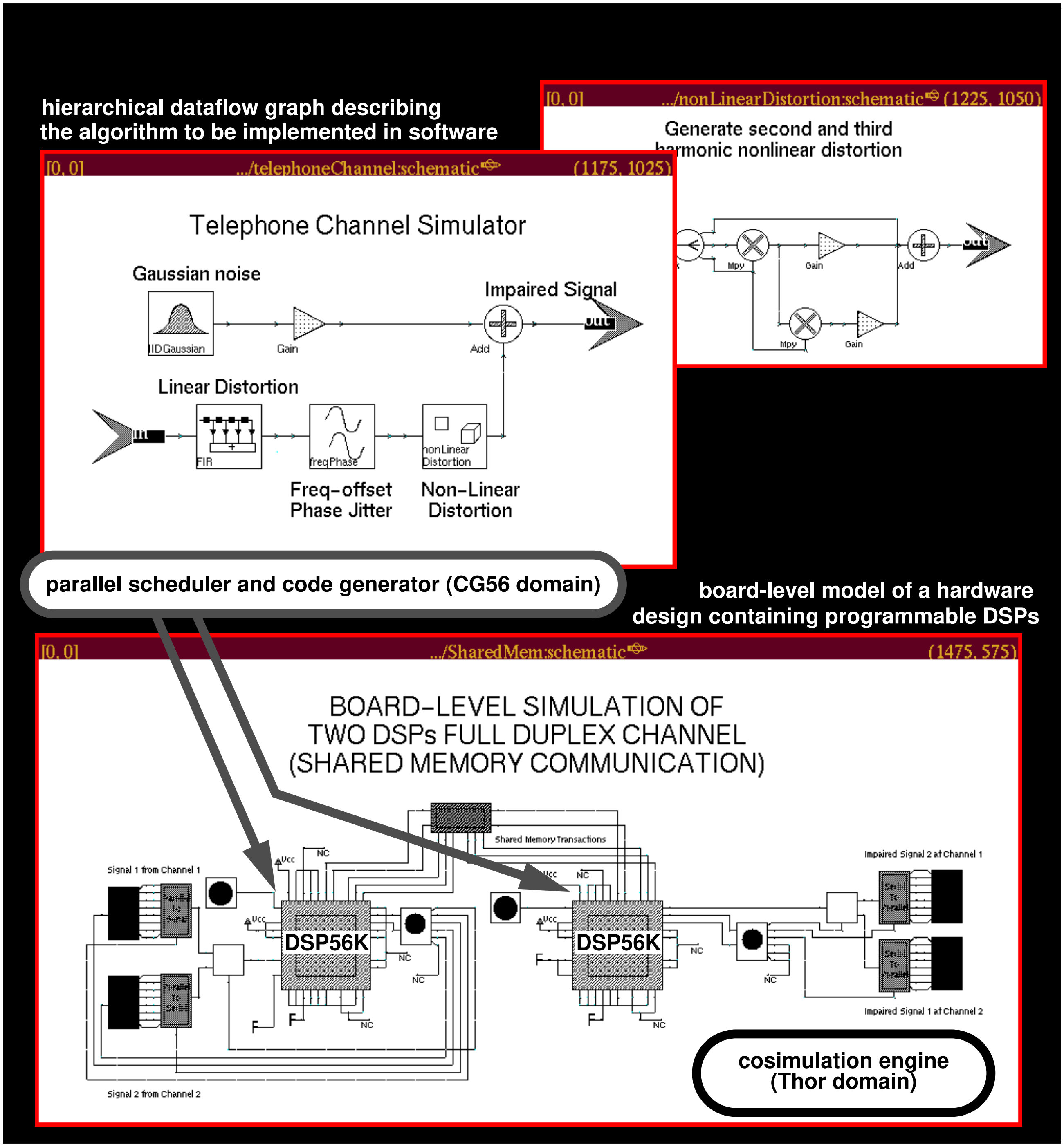

FIGURE 1. A board-level specification of a hardware design containing programmable DSPs (bottom window) is combined with a high-level block-diagram representation of the algorithm to be executed by the DSPs. Ptolemy handles code generation for the DSPs, including partitioning for parallel execution, and hardware/software cosimulation.

Each design style is supported by one or more *domains*. A domain is an extensible library of functional blocks and model of computation. By “model of computation” we mean the operational semantics of a network of functional blocks. Key to Ptolemy is the ability to combine multiple domains in a single design. Table 1 summarizes some of the domains that Berkeley is currently working on.

Using heterogeneous design styles also requires using tools from different sources. The Berkeley team has already demonstrated varying levels of integration between Ptolemy and a number of externally developed tools, including Matlab (from The Math Works), Hyper (a VLSI hardware synthesis

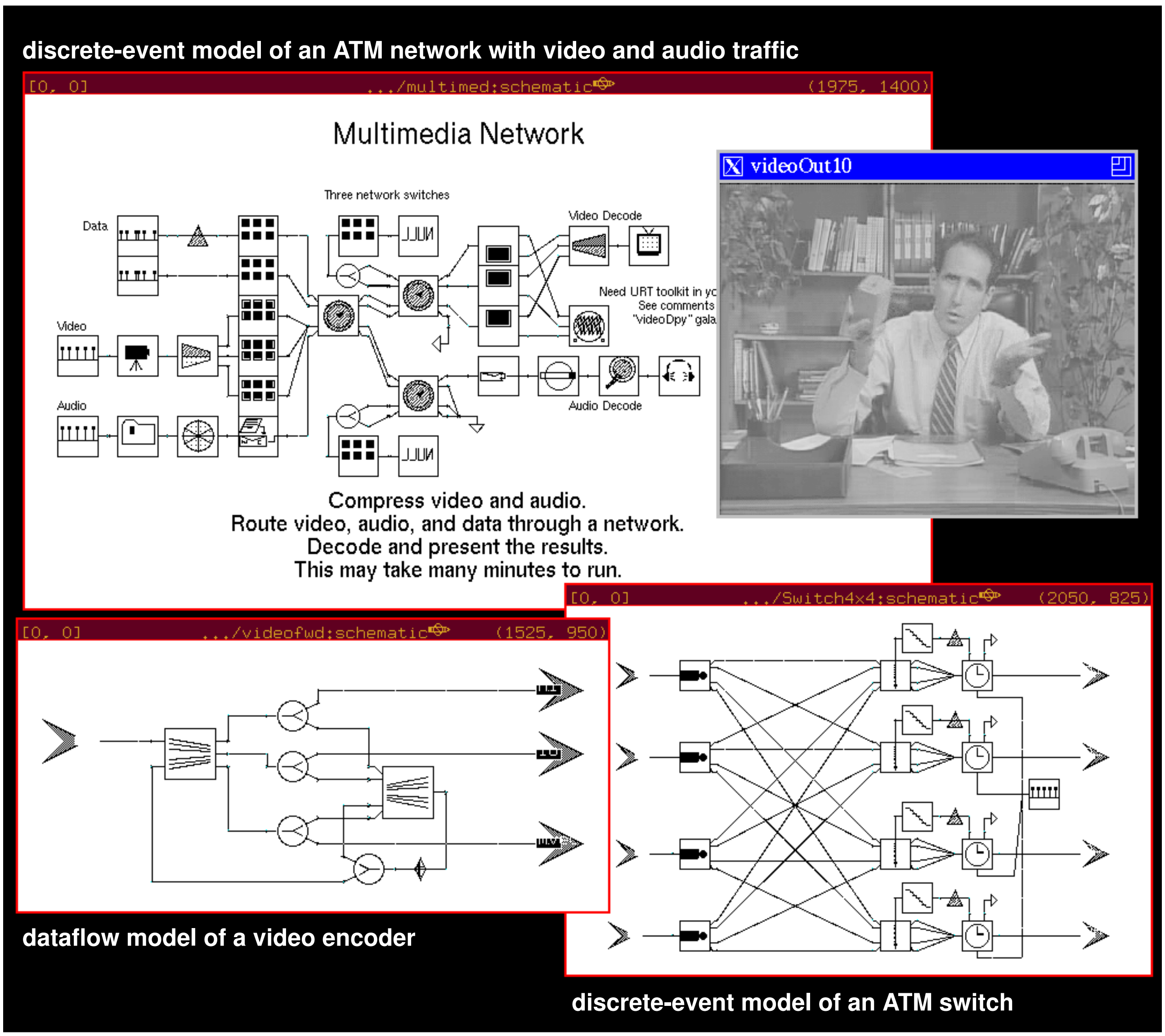

FIGURE 2. A multimedia networking application that combines discrete-event modeling of an ATM (asynchronous transfer mode) packet-switched network with video and audio coding and decoding (application developed by Paul Haskell).

tool from Berkeley), Thor (an RTL-level circuit simulator from Stanford), sim56000 and sim96000 (instruction-set simulators from Motorola for their programmable DSPs).

Key research problems being addressed by this broad project include:

- Hardware/software codesign methodology.
- Domain-specific signal processing design.
- Dataflow semantics (for signal processing).
- Hierarchical finite-state machine semantics (for control).
- Synthesis of embedded software.

Name	Expansion	Principal Use
SDF	synchronous dataflow	synchronous signal processing
DDF	dynamic dataflow	asynchronous signal processing
BDF	boolean dataflow	asynchronous signal processing
MDSDF	multidimensional dataflow	multidimensional signal processing
DE	discrete event	communication network modeling and determinate high-level system modeling
FSM	finite state machines	control
HOF	higher-order functions	graphical programming
Thor	(name given at Stanford)	RTL hardware simulation
MQ	message queue	telecommunications switching software
CP	communicating processes	communication network modeling nondeterminate system modeling
CGC	code generation - C	software synthesis (SDF or BDF model)
CG56	code generation - DSP56000	firmware synthesis (SDF model)
CG96	code generation - DSP96000	firmware synthesis (SDF model)
Silage	a functional language	VLSI hardware synthesis (SDF model)
VHDLF	VHDL - functional	high-level modeling and design (SDF)
VHDLB	VHDL - behavioral	hardware modeling and design (DE)
Sproc	a multiprocessor DSP from Star Semiconductor	firmware synthesis (SDF)

Table 1: Some domains that have been implemented in Ptolemy. The shaded area indicates simulation domains. The rest are code-generation domains.

In summary, the key idea in the Ptolemy project is to mix models of computation, rather than trying to develop one, all-encompassing model. The rationale is that specialized models of computation are (1) more useful to the system-level designer, and (2) more amenable to high-quality high-level synthesis of hardware and software. The Ptolemy kernel demonstrates one way to mix tools that have fundamentally different semantics, and provides a laboratory for experimenting with such mixtures.

More information about the Ptolemy project, plus access to all of the software and documentation, is available on the worldwide web via the URL “http://ptolemy.eecs.berkeley.edu”.