Overview of the Ptolemy Project

Brian L. Evans and H. John Reekie

Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720-1770

{ble, johnr}@eecs.berkeley.edu
http://ptolemy.eecs.berkeley.edu/

y[n] = h[n] * a^n u[n]

Project directed by Prof. Edward A. Lee and co-directed by Prof. David G. Messerschmitt
A Typical Embedded Signal Processing System

- control panel
- ASIC
- microcontroller
- real-time operating system
- controller process
- user interface process
- system bus
- host port
  - programmable DSP
  - memory interface
- dual-ported memory
- CODEC
- DSP assembly code
- analog interface
Heterogeneity in System-Level Design

- System-level modeling
  - Imperative
  - FSM
  - Dataflow
  - Discrete event

- Synthesis
  - Partitioning
  - Compiler
  - Software synthesis
  - ASIC synthesis
  - Logic synthesis

- Detail modeling and simulation
  - Execution model
  - ASIC model
  - Logic model

- Cosimulation
  - Symbolic
Ptolemy Project

Design Methodologies for Heterogeneous Systems

• Formal models of computation
• Hierarchical compositions of models form complex systems
• Synthesis and partitioning algorithms
• Laboratory to test design methodology is the Ptolemy software environment

Personnel

• Directors: Profs. Edward Lee and David Messerschmitt
• Staff: 4 post-doctoral, 1 software manager, 2 administrative
• Students: 12 graduate and 3 undergraduate
Hierarchical Graphs As Underlying Abstract Syntax

Attach semantics

Dataflow

Discrete-Event

Finite State Machine
Computational Models (Domains) in Ptolemy

- Ptolemy Kernel
  - untimed
  - timed
  - control

- MDSD
  - multidimensional synchronous dataflow

- DE
  - discrete-event

- CP
  - communicating processes
  - synchronous/reactive

- FSM
  - finite state machine

- SR
  - synchronous/reactive

- C
  - 56000
  - 96000
  - C50

- VHDL

- VHDLB

- Silage

- CG
Impact on Industrial CAD Tools

- **Synchronous Dataflow model plus multirate schedulers** (1992) → **Signal Processing WorkSystem from Cadence**
- **Heterogeneous simulation support by Ptolemy kernel** (1995) → **CONVERGENCE environment from Cadence**
- **Synchronous Dataflow and Discrete-Event domains** (1995) → **Ptolemy HSIM from Berkeley Design Technology, Inc.**
- **Ptolemy kernel plus dataflow domains** (1995) → **Sanders ENTIRE framework (multiprocessor architecture mapping and FPGA synthesis)**
- **Ptolemy kernel plus dataflow domains** (1996) → **Thomson-CSF radar simulator**

http://ptolemy.eecs.berkeley.edu/third-party.html
<table>
<thead>
<tr>
<th>Specification</th>
<th>Topic</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>System optimization</td>
<td>System rearrangement</td>
</tr>
<tr>
<td></td>
<td>Scalable systems</td>
<td>Higher-order functions</td>
</tr>
<tr>
<td></td>
<td>Visual languages</td>
<td>Tycho</td>
</tr>
<tr>
<td></td>
<td>Formal semantics</td>
<td>Tagged signal model</td>
</tr>
<tr>
<td>Simulation</td>
<td>Dataflow models of computation</td>
<td>Multidimensional dataflow</td>
</tr>
<tr>
<td></td>
<td>Control models of computation</td>
<td>Synchronous/reactive</td>
</tr>
<tr>
<td></td>
<td>Cosimulation</td>
<td>Mixed signal simulation</td>
</tr>
<tr>
<td></td>
<td>Native signal processing</td>
<td>UltraSparc visual instruction set</td>
</tr>
<tr>
<td>Synthesis</td>
<td>Partitioning</td>
<td>Hardware/software codesign</td>
</tr>
<tr>
<td></td>
<td>Uniprocessor scheduling</td>
<td>Program/data code minimization</td>
</tr>
<tr>
<td></td>
<td>Multiprocessor scheduling</td>
<td>Hierarchical scheduling</td>
</tr>
<tr>
<td></td>
<td>Distributed systems</td>
<td>Network of workstations</td>
</tr>
</tbody>
</table>
Synchronous/Reactive Model of Computation

- **Synchronous** means that computations occur “instantaneously” at integer clock ticks (Lustre, Signal)
- **Reactive** means that the model responds to the environment at the speed of the environment (Esterel, StateCharts)
- Modules are **monotonic functions** operating on complete partial orders
- Execution proceeds by iterating towards the fixed-point — compile-time analysis finds an execution order guaranteed to produce the least fixed-point
- Schedules determined at compile-time in polynomial time

Zero-delay blocks | Instantaneous, bidirectional communication
---|---

Heterogenous hierarchical system construction
Mixing FSMs and Dataflow Models of Computation

- Each Finite State Machine (FSM) has a single thread of control
- Arranging FSMs in a nested tree specifies hierarchical FSMs
- Arbitrary mixing of FSMs and dataflow models of computation captures the 21 variants of Statecharts
- Formal analysis (verification) possible due to finite state
Vision for Common Operating Environments

Requirements to avoid

• Avoid one monolithic standard
• Avoid standardizing on one general purpose language
• Avoid one specification format

Requirements to include

• Support of domain-specific models of computation/tools
• Support imperative and declarative styles of programming
• Support multiple specification formats, such as directed acyclic graphs, textual languages, algebraic descriptions
• Support back annotation
• Support general frameworks for tools to interface with
• Support cosimulation of arbitrary levels of abstraction