A Brief Tutorial on Models of Computation

The MESCAL Team
UC Berkeley
Fall 2001

- The laws that govern communication and concurrency

- What do you get?
  - Description that is natural to the application domain
  - Description that is expressive
  - Descriptive approach that is not cumbersome
  - Description amenable to formal analysis
  - Description should be executable and a high-performance simulation should be able to be generated from it
Brief Tutorial on Models of Computation

- CSP – concurrent threads with rendezvous
- PN – process networks
- TM – Timed multithreading, prioritization
- FSM – finite state machines
- SR – Synchronous Reactive
- CT – continuous-time modeling
- DE – discrete-event systems
- DDE – distributed discrete-event systems
- Giotto – time driven cyclic models
- Petri Nets – places and transitions
- GR – graphics
- SDF – synchronous dataflow
- xDF – other dataflow
- DT – discrete time, DF with time
- MescalPE – VLIW datapaths
- CLICK – Router packet dataflow
- Meta – function calls

Process Domains

- CSP
  - Components are sequential processes that run concurrently
  - Synchronous message passing
  - Good for resource management problems
    - Dining Philosophers
    - Hardware bus contention
    - Nondeterminism
    - Liveness
    - Fairness
    - Deadlock
Process Domains

- **PN**
  - Kahn-MacQueen Process Network
  - Components are sequential processes that run concurrently
  - Communication channels are unbounded FIFOs
    - Get operation blocks until data is available.
    - Processes cannot poll for data
  - Deterministic execution
  - Bounded memory with blocking writes
  - Good for streaming signal processing applications

Control Domains

- **FSM**
  - *charts formalism
  - Hierarchical decomposition of states
  - Concurrent FSMs with broadcast communication
  - Good for:
    - Sequential control logic
    - Modal models
CT - Continuous Time

- Components perform mathematical operations on signals
  - Includes Integrators
- Executing a model means solving ODEs numerically
- Good for modeling physical systems
  - Simulate analog circuit
  - Control systems
  - SPICE

DE - Discrete Event

- Components communicate with events that are ordered on a global timeline
- Send operation transmits an event with a timestamp
- Get operation only sees events that are available at the current time
- Good for:
  - Digital logic - VHDL and Verilog simulators are DE simulators
  - Less detailed than a CT simulation, and faster
- DDE - Distributed Discrete Event
  - Each component has a local notion of time
  - Global synchronization preserved
Petri Nets

• C.A. Petri, 1966
• Places
• Transitions
• Tokens
• Good for:
  • Modeling scheduling systems
  • Industrial control systems
  • Analyzing network protocols

Dataflow Models

• SDF – Synchronous Dataflow
  • Components execute according to a predetermined schedule
  • Communications is strictly controlled
    • Components must always consume a specified number of tokens on their input ports
    • Produce a specified number of tokens on their output ports
  • No time
  • Good for applications without complex control flow
    • Digital signal processing
Dataflow Variants

• HDF – Heterochronous Dataflow
  • A generalization of SDF
  • Components can change their token production/consumption rates between iterations
  • Must choose from a fixed selection of type signatures
  • Dataflow with different modes

• DT – Discrete Time
  • SDF with time
  • A model has a period and a global time
  • Each communication channel has a local time
    \[ \text{global time} \leq \text{local time} \leq \text{global time} + \text{period} \]
  • Compose DF with other timed domains like DE or CT

Dataflow Variants

• Boolean-controlled dataflow (BDF)

• Integer-controlled dataflow (IDF)

• Dynamic dataflow (DDF)
Dataflow Variants

- MescalPE
  - Uses a BDF foundation to describe VLIW datapaths
  - Implicit pipeline registers
  - Extract reservation table
  - Generate a compiler

---

CLICK

- Process Network
- Processes are IDF-like blocks
- Hierarchical composition of PN and IDF
- Good for network routers
Meta

• Components are objects that communicate with imperative function call semantics
• An output port indicates that an object performs function calls
• An input port indicates that an object accepts function calls
• Control follows communication from component to component
• No strong guarantees or provisions like other MoCs
• Good for:
  • Metamodelling – making models of other models

Ptolemy II

• Java framework for experimenting with models of computation
• Models are composeable hierarchically
• Actors are type polymorphic and domain polymorphic
The MESCAL Team

- Kurt Keutzer
- Matthias Gries
- Christian Sauer
- Kees Vissers
- Yujia Jin
- Andrew Mihal
- Matt Moskewicz
- Will Plishker
- Kaushik Ravindran
- Niraj Shah
- Scott Weber