

Ptolemy II

Models of Computation

Domains

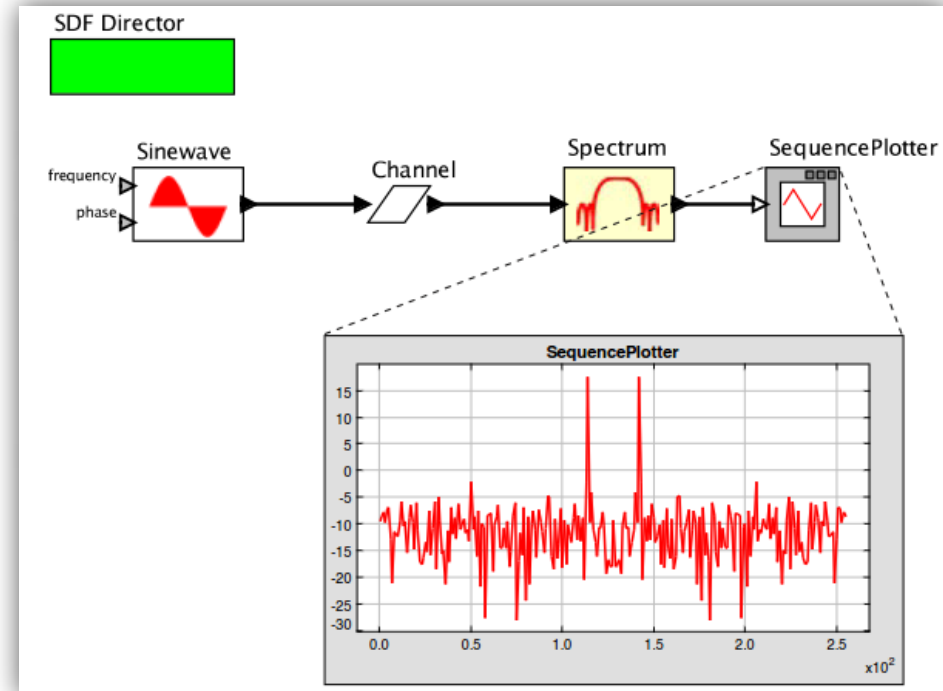
Dataflow

- Used for **pipe-and-filter** models
- Actors are fired when their required inputs become available
- Issues:
 - Buffer size
 - Scheduling
 - Liveness
- SDF: synchronous dataflow
 - Data rates statically specified
- Example:

SDF Example

- The Spectrum actor requires 256 tokens to fire

All other actors require one token each.

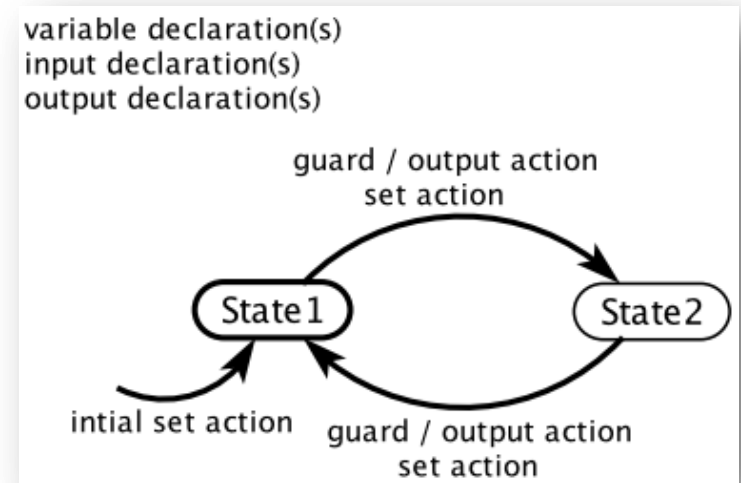


Synchronous Reactive

- Useful for state machines
- Logical time ticks
 - A tick represents a reaction
 - Each actor performs one iteration per tick
- And SR model has simultaneous and instantaneous reaction of all actors at each tick
- Actors may react to *absence* of inputs
- Fixed point semantics (see the state machine lectures)

Finite State Machines

- *Initialize* of an FSM actor:
 - Executes the initial set action
 - Sets the initial state
- *Fire* of an FSM actor:
 1. Reads **inputs**
 2. Evaluates outgoing **guards**
 3. Chooses an enabled transition
 4. Executes the **output** actions (if any)
- *Postfire* of an FSM actor:
 1. executes the set actions
 2. Changes the current state



FSM example: thermostat actor

The screenshot shows a software interface for editing a Finite State Machine (FSM) actor. The window title is "file:/C:/Data/work/department/teachin.../olemy/ECPS2014/Hysteresis.xml#FSMActor". The interface includes a menu bar (File, View, Edit, Graph, Debug, Help) and a toolbar with various icons for file operations and simulation control. On the left, a tree view shows project components: DocViewerAttribute, LocalPreferences, RepaintController, Decorative, Parameters, Analysis, and State. The main workspace displays a state machine diagram with two states: "heating" and "cooling".

heating state:
guard: temperature < heatOffThreshold
output: heat = heatingRate

cooling state:
guard: temperature > heatOnThreshold
output: heat = coolingRate

Transitions:
- From heating to cooling: guard: temperature >= heatOffThreshold, output: heat = coolingRate
- From cooling to heating: guard: temperature <= heatOnThreshold, output: heat = heatingRate

Parameters:

- heatOnThreshold: 18.0
- heatOffThreshold: 22.0
- heatingRate: 0.1
- coolingRate: -0.05

Text description:
This state machine models hysteresis in a simple thermostat. Each time it gets a temperature input, it produces a heat output depending on its current state and the value of the input, and possibly takes a transition to the other state.

Usage of the thermostat actor

This example illustrates the use of finite state machines (FSMs) to model systems with hysteresis. The TemperatureModel takes heat as input and produces temperature as output. The FSMActor models a simple thermostat that turns the heat on and off. When the heat is off, a negative heating rate is provided to model cooling.

Run the model to see its behavior. Open the FSMActor to see the implementation.

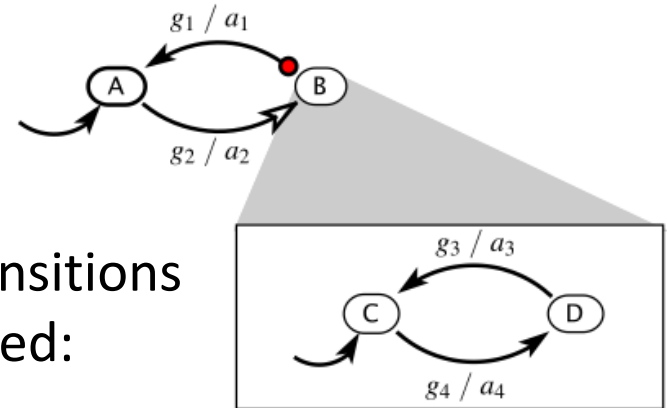
Modal Models

- *Fire* of a ModalModel:

1. Reads **inputs**
2. Evaluates **guards** of preemptive transitions
3. If no preemptive transition is enabled:
 1. Fires the refinement of current state
 2. Evaluates guards of non-preemptive outgoing transitions
4. Chooses an enabled transition
5. Executes the **output** actions (if any)

- *Postfire* of an FSM actor:

1. Postfires the refinement of current state, if fired
2. executes the set actions of chosen transition
3. Changes the current state
4. Initializes refinement of destination state if reset transition



Discrete Event Director (DE)

- Has the notion of *model time*
- Maintains an event queue, where events are stored in increasing timestamp order
- At each iteration processes the events with the smallest timestamp in the queue (by iterating their target actors)
- Event are introduced in the queue by
 - An actor calling the *fireAt* method of the director:
The calling actor will be the target and the event timestamp may be in the future
 - A token received in an input port: the *put* method of the DE receiver inserts an event having as target the container of the input port and as timestamp the current model time