

# SBML Event Semantics

Chris Myers

University of Utah

SBML Hackathon  
May 3, 2008

- ID / Name (optional)
- Trigger - expression evaluating to a Boolean
- Delay (optional) - expression evaluating to a non-negative real number
- UseValuesAtTriggerTime - Boolean
- List of event assignments

# Current SBML Event Semantics

- Event enabled when the trigger expression changes from false to true.
- Delay evaluated and event scheduled at current time plus delay.
- If `UseValuesAtTriggerTime` is true:
  - Event assignment expressions are evaluated at trigger time.
  - Event assignments are performed when the event fires.
- If `UseValuesAtTriggerTime` is false:
  - Event assignment expressions are evaluated and performed when the event fires.

# Problems (features?) with Current Semantics

- Events once enabled cannot be disabled.
- No prescribed semantics when events are scheduled to fire simultaneously.
- Events enabled at time = 0 never fire.

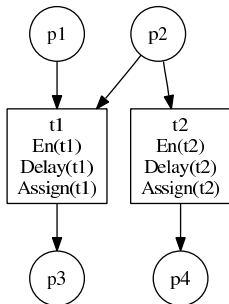
# Event Disabling: Proposal

- Add Boolean TriggerCanBeDisabled
- If TriggerCanBeDisabled is true, an event that has its trigger disabled before it fires, will not fire.
- This only has a minimal impact on simulators.
  - Since triggers already must be evaluated during each simulation cycle to determine which have become newly enabled, detecting those which are disabled requires only a few additional lines of code.

# Event Disabling: Motivation

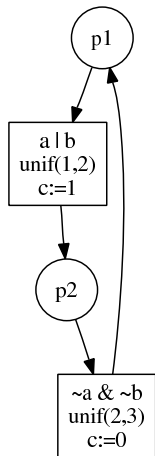
- Enables both transport and inertial delays.
- Enables Petri net, State Machine, and Boolean models to be created in SBML without a new package.

# Petri Nets (or Finite State Machines)



ID	t1	t2
Trigger	$(t > 0) \wedge (p1 = 1)$ $\wedge (p2 = 1) \wedge En(t1)$	$(t > 0) \wedge (p2 = 1)$ $\wedge En(t2)$
Delay	$Delay(t1)$	$Delay(t2)$
TCBD	true	true
Event assignments	$p1 := 0$ $p2 := 0$ $p3 := 1$ $Assign(t1)$	$p2 := 0$ $p4 := 1$ $Assign(t2)$

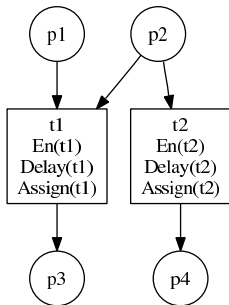
# Boolean Logic



ID	t1	t2
Trigger	$(t > 0) \wedge (p1 = 1) \wedge (a \vee b)$	$(t > 0) \wedge (p2 = 1) \wedge (\neg a \wedge \neg b)$
Delay	unif(1,2)	unif(2,3)
TCBD	true	true
Event assignments	$p1 := 0$ $p2 := 1$ $c := 1$	$p2 := 0$ $p1 := 1$ $c := 0$



# Simultaneous Events



ID	t1	t2
Trigger	$(t > 0) \wedge (p1 = 1) \wedge (p2 = 1) \wedge \text{En}(t1)$	$(t > 0) \wedge (p2 = 1) \wedge \text{En}(t2)$
Delay	Delay(t1)	Delay(t2)
TCBD	true	true
Event assignments	$p1 := 0$ $p2 := 0$ $p3 := 1$ Assign(t1)	$p2 := 0$ $p4 := 1$ Assign(t2)

# Possible Semantics for Simultaneous Events

- Stop
- Execute all in a defined order
- Execute all in a random order
- Execute 1 in a defined order
- Execute 1 randomly

# Possible Semantics for Simultaneous Events

- Stop ← not likely the desired behavior
- Execute all in a defined order
- Execute all in a random order
- Execute 1 in a defined order
- Execute 1 randomly

# Possible Semantics for Simultaneous Events

- Execute **all** in a defined order
- Execute **all** in a random order
- Execute **1** in a defined order
- Execute **1** randomly

# Possible Semantics for Simultaneous Events

- Execute **all** in a defined order
- Execute **all** in a random order
- Execute **1** in a defined order
- Execute **1** randomly
- All, 1, or something in-between can be determined by `TriggerCanBeDisabled`.

# Possible Semantics for Simultaneous Events

- Execute in a **defined** order
- Execute in a **random** order

# Possible Semantics for Simultaneous Events

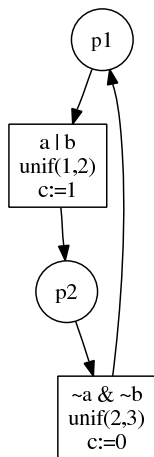
- Execute in a **defined** order
- Execute in a **random** order
- Stochastic simulators should use a random order.

# Possible Semantics for Simultaneous Events

- Execute in a **defined** order
- Execute in a **random** order
- Stochastic simulators should use a random order.
- Deterministic simulators should produce same outcome each time, but different simulators can produce different outcomes (perhaps).

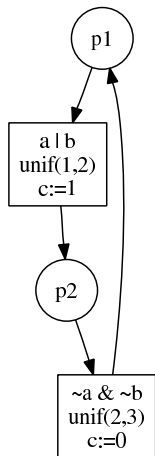


# No Events at Time = 0



ID	t1	t2
Trigger	$(t > 0) \wedge (p1 = 1) \wedge (a \vee b)$	$(t > 0) \wedge (p2 = 1) \wedge (\neg a \wedge \neg b)$
Delay	unif(1,2)	unif(2,3)
TCBD	true	true
Event assignments	$p1 := 0$ $p2 := 1$ $c := 1$	$p2 := 0$ $p1 := 1$ $c := 0$

# Initial Value for Trigger



ID	t1	t2
Trigger	$(p1 = 1) \wedge (a \vee b)$	$(p2 = 1) \wedge (\neg a \wedge \neg b)$
InitialValue	false	false
Delay	$\text{unif}(1,2)$	$\text{unif}(2,3)$
TCBD	true	true
Event assignments	$p1 := 0$ $p2 := 1$ $c := 1$	$p2 := 0$ $p1 := 1$ $c := 0$

# Timed Events

- Timed events complicate stochastic simulation.

ID	t1
Trigger	geq(t,100)
Delay	
TCBD	false
Event assignments	<i>Cro</i> := 50

# Timed Events

ID	t1
Trigger	true
InitialValue	false
Delay	100
TCBD	false
Event assignments	<i>Cro</i> := 50

# Timed Events

ID	t1
Trigger	true
InitialValue	false
Delay	<i>uniform(75, 125)</i>
TCBD	false
Event assignments	<i>Cro := 50</i>

# Summary

- TriggerCanBeDisabled is **essential** for PN, FSM, and logical models.
- Simultaneous events should have a defined semantics.
- Having an initial value for triggers at  $t < 0$  is useful.