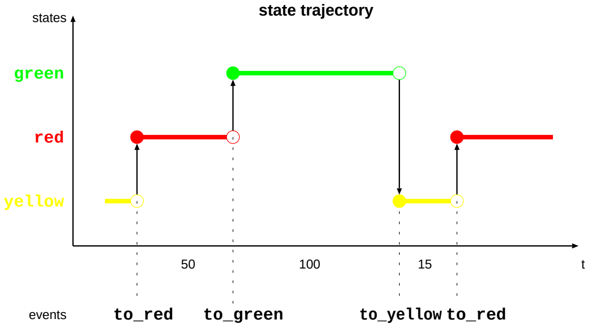
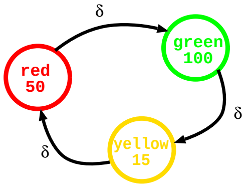
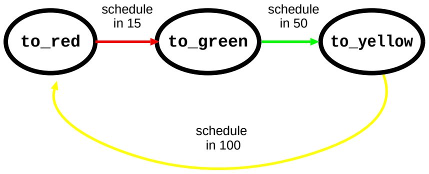
Event Graphs



Finite State Automaton



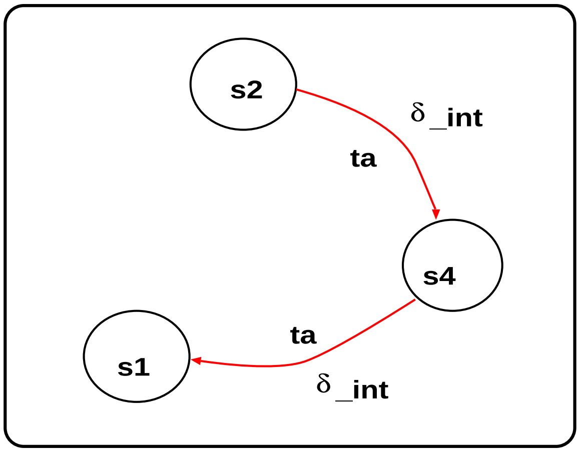
Event (Scheduling) Graph



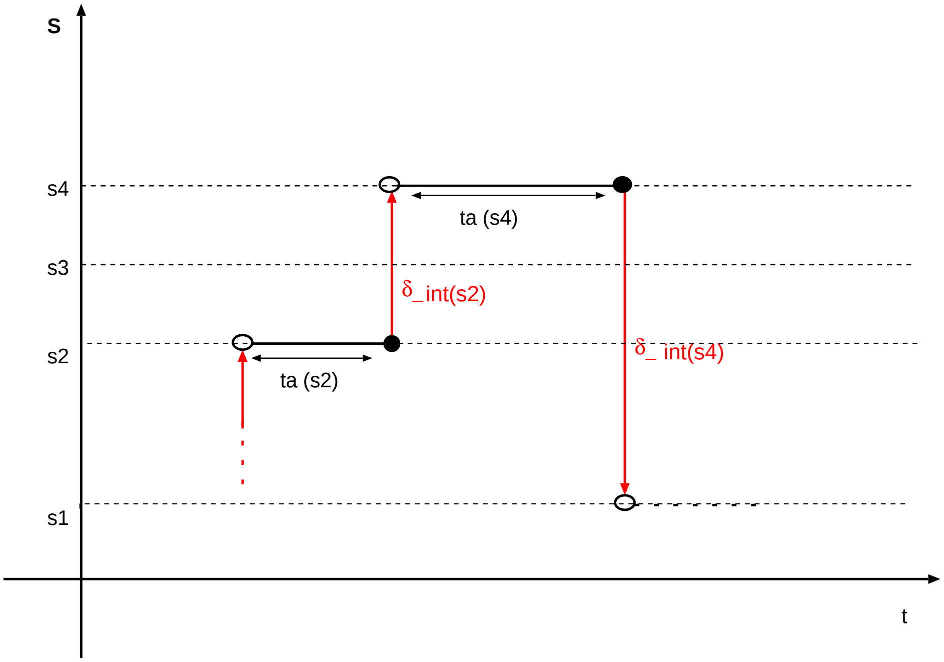
Discrete EVent System specification (DEVS)

A formal basis
for (low-level) representation
of *all* discrete event modelling formalisms
and simulator implementations

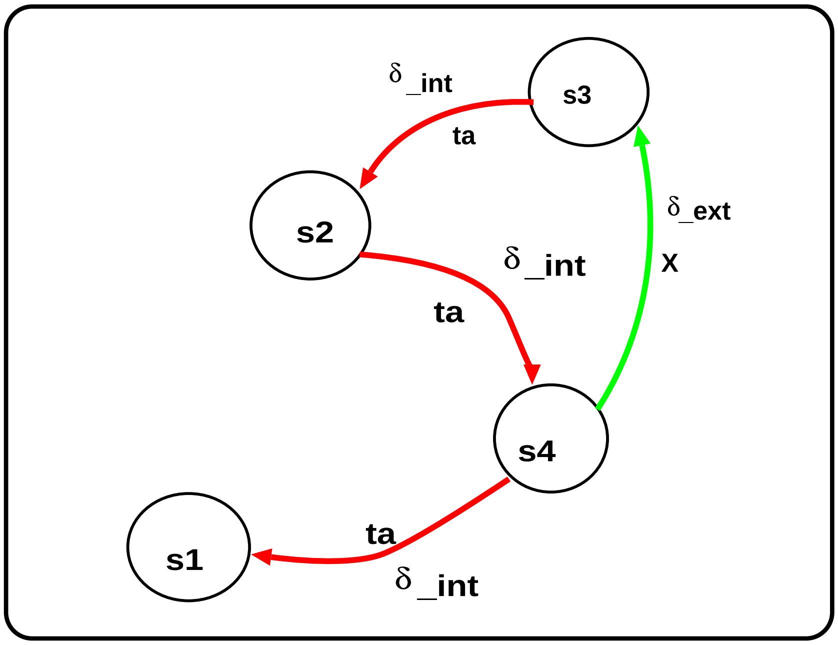
DEVS without external events (FSA)



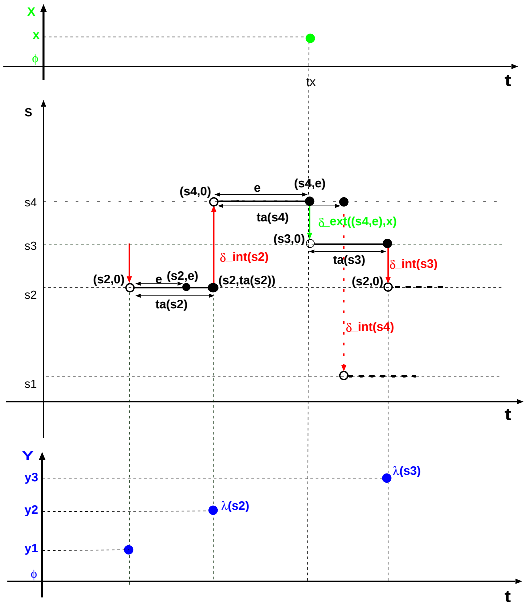
DEVS without external events (states)



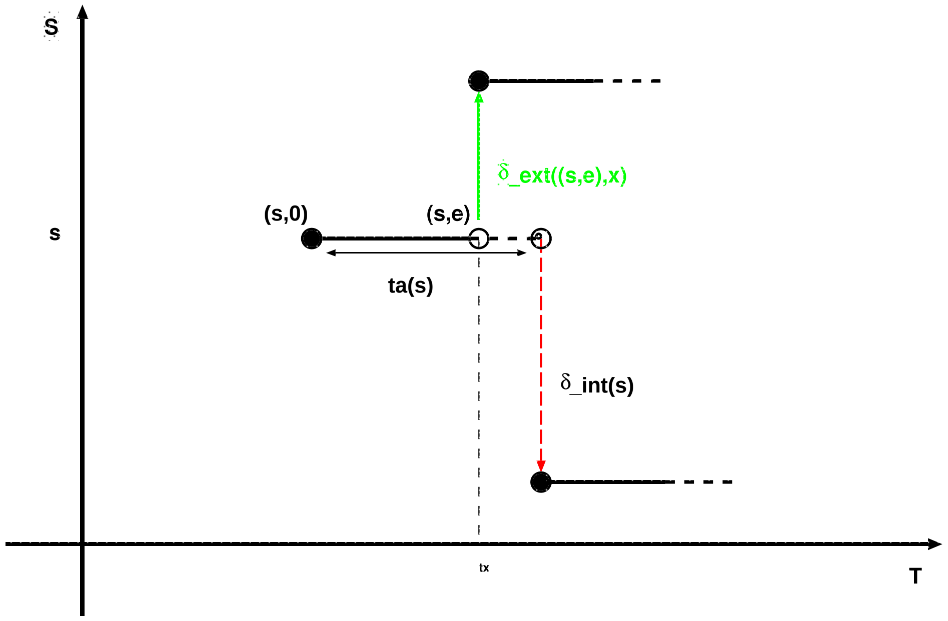
DEVS with external events



DEVS with external events (states)



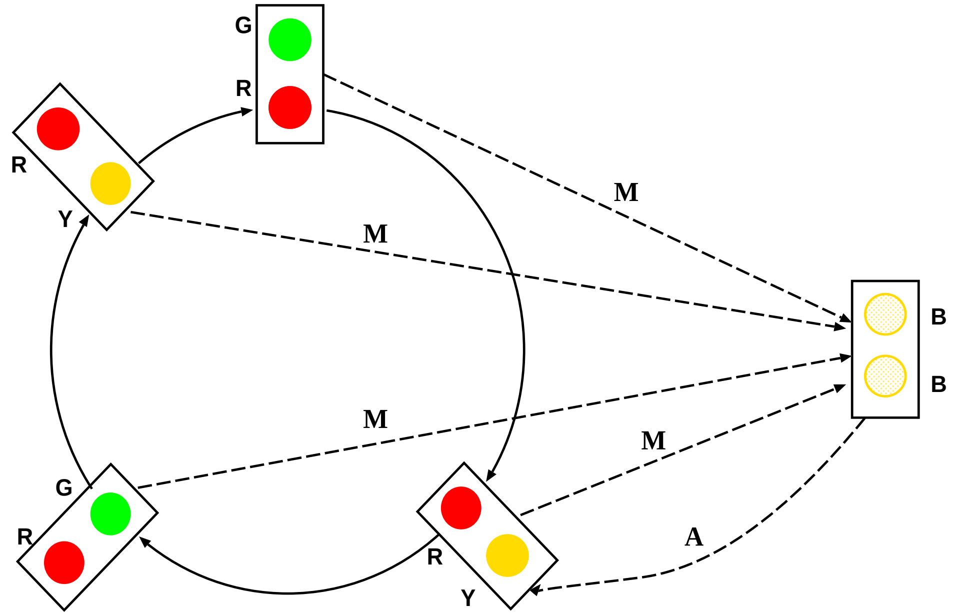
DEVS essence



$$DEVS = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$$

$T = \mathbb{R}$	time base
X	input set
$\omega : T \to X \cup \{\phi\}$	input segment
S	state set
Y	output set
$\delta_{int} : S \to S$	internal transition function
$ta : S \to \mathbb{R}^{+}_{0,\infty}$	time advance function
$Q = \{(s,e) \mid s \in S, 0 \leq e \leq ta(s)\}$	total state, *e* is elapsed time
$\delta_{ext} : Q \times X \to S$	external transition function
$\lambda : S \to Y$	output function

Traffic Lights



$$trafficDEVS =< X,S,Y,\delta_{int},\delta_{ext},\lambda,ta >$$

$$T = \mathbb{R}$$

$$X = \{M,A\}$$

$$\omega : T \to X \cup \{\phi\}$$

$$S = \{RG,RY,GR,YR,BB\}$$

$$\delta_{int}(RG) = RY;\ \delta_{int}(RY) = GR$$

$$\delta_{int}(GR) = YR;\ \delta_{int}(YR) = RG$$

$$ta(RG) = 60s;\ ta(RY) = 10s$$

$$ta(GR) = 50s;\ ta(YR) = 10s$$

$$ta(BB) = +\infty$$

$$trafficDEVS = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta >$$

$$\delta_{ext}((RG, e), M) = BB$$

$$\delta_{ext}((RY, e), M) = BB$$

$$\delta_{ext}((GR, e), M) = BB$$

$$\delta_{ext}((YR, e), M) = BB$$

$$\delta_{ext}((BB, e), A) = RY$$

$$Y = \{GREY, YELLOW, BLINK\}$$

$$\lambda(RG) = \lambda(RY) = \lambda(GR) = GREY$$

$$\lambda(YR) = YELLOW$$

$$\lambda(BB) = BLINK$$

Coupled DEVS

$$coupledDEVS \equiv \langle X_{self}, Y_{self}, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle$$

$$\{M_i | i \in D\}.$$

$$M_i = \langle S_i, ta_i, \delta_{int,i}, X_i, \delta_{ext,i}, Y_i, \lambda_i \rangle, \forall i \in D.$$

$$\{I_i | i \in D \cup \{self\}\}.$$

$$\forall i \in D \cup \{self\} : I_i \subseteq D \cup \{self\}.$$

$$\forall i \in D \cup \{self\} : i \notin I_i.$$

$Z_{i,j}$ output-to-input translation

$$\{Z_{i,j} | i \in D \cup \{self\}, j \in I_i\},$$

$$\begin{aligned} Z_{self,j} &: X_{self} \rightarrow X_j &&, \forall j \in D, \\ Z_{i,self} &: Y_i \rightarrow Y_{self} &&, \forall i \in D, \\ Z_{i,j} &: Y_i \rightarrow X_j &&, \forall i, j \in D. \end{aligned}$$

Together, I_i and $Z_{i,j}$ completely specify the coupling (structure and behaviour)

Tie-breaking among simultaneous events

$$select : 2^D \rightarrow D$$

Choose a unique component from any non-empty subset E of D:

$$select(E) \in E.$$

E corresponds to the set of all components having a state transition simultaneously (*collisions*).

Closure under coupling

From the coupled DEVS

$$\langle X_{self}, Y_{self}, D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select\rangle,$$

with all components M_i atomic DEVS models

$$M_i = \langle S_i, ta_i, \delta_{int,i}, X_i, \delta_{ext,i}, Y_i, \lambda_i\rangle, \forall i \in D$$

the atomic DEVS

$$\langle S, ta, \delta_{int}, X, \delta_{ext}, Y, \lambda\rangle$$

is constructed.

Closure: state and time-advance

$$S = \times_{i \in D} Q_i,$$

where

$$Q_i = \{(s_i, e_i) | s \in S_i, 0 \leq e_i \leq ta_i(s_i)\}, \forall i \in D.$$

$$ta : S \to \mathbb{R}^+_{0,+\infty}$$

Select the *most imminent* event time, $=$ smallest time *remaining* until internal transition, of all the components

$$ta(s) = min\{\sigma_i = ta_i(s_i) - e_i | i \in D\}.$$

Dealing with simultaneous events

Imminent components:

$$IMM(s) = \{i \in D | \sigma_i = ta(s)\}.$$

select one component i^* of the coupled model

$$\begin{array}{rcccl} select & : & 2^D & \rightarrow & D \\ & & IMM(s) & \rightarrow & i^* \end{array}$$

Output (at internal transition time)

$$\begin{array}{rcll} \lambda(s) & = & Z_{i^*,self}(\lambda_{i^*}(s_{i^*})) & \text{,if } self \in I_{i^*}, \\ & & \phi & \text{,if } self \notin I_{i^*}. \end{array}$$

Conceptually, the non-event ϕ is generated if i^* is not connected to the output of the coupled model.

Internal transition function

$$\delta_{int}(s) = (\ldots, (s'_j, e'_j), \ldots), \text{ where}$$

$$\begin{array}{rcll}
(s'_j, e'_j) & = & (\delta_{int,j}(s_j), 0) & ,\text{for } j = i^*, \\
 & = & (\delta_{ext,j}(s_j, e_j + ta(s), Z_{i^*,j}(\lambda_{i^*}(s_{i^*}))), 0) & ,\text{for } j \in I_{i^*} \\
 & & & (\text{and } Z_{i^*,j}(\lambda_{i^*}(s_{i^*})) \neq \phi), \\
 & = & (s_j, e_j + ta(s)) & ,\text{otherwise.}
\end{array}$$

External transition function

$$\delta_{ext}(s, e, x) = (\ldots, (s'_i, e'_i), \ldots), \text{ where}$$

$$\begin{array}{rcll}
(s'_i, e'_i) & = & (\delta_{ext,i}(s_i, e_i + e, Z_{self,i}(x)), 0) & ,\text{for } i \in I_{self}, \\
 & = & (s_i, e_i + e) & ,\text{otherwise.}
\end{array}$$

DEVS limitations

- a conflict due to simultaneous internal and external events is resolved by ignoring the internal event. It should be possible to explicitly specify behaviour in case of conflicts;
- there is limited potential for parallel implementation;
- the $select$ function is an artificial legacy of the semantics of traditional sequential simulators based on an event list;
- it is not possible to explicitly describe variable structure.

Some of these are resolved in *parallel DEVS*

DEVS Solver

- Iterative simulation of DEVS model
- Possibly distributed implementation

message m	**simulator**	**coordinator**
$(*, from, t)$	simulator correct only if $t = t_N$	
	$y \leftarrow \lambda(s)$	**send** $(*, self, t)$ to i^*, where
	if $y \neq \phi$:	$i^* = select(imm\_children)$
	send $(\lambda(s), self, t)$ to *parent*	$imm\_children = \{i \in D \mid M_i.t_N = t\}$
	$s \leftarrow \delta_{int}(s)$	$active\_children \leftarrow active\_children \cup \{i^*$
	$t_L \leftarrow t$	
	$t_N \leftarrow t_L + ta(s)$	
	send $(done, self, t_N)$ to *parent*	

message m	**simulator**	**coordinator**
$(x, from, t)$	simulator correct only if $t_L \leq t \leq t_N$ (ignore δ_{int} to resolve a $t = t_N$ *conflict*)	
	$e \leftarrow t - t_L$	$\forall i \in I_{self}$:
	$s \leftarrow \delta_{ext}(s, e, x)$	**send** $(Z_{self,i}(x), self, t)$ to i
	$t_L \leftarrow t$	$active\_children \leftarrow active\_children \cup \{$
	$t_N \leftarrow t_L + ta(s)$	
	send $(done, self, t_N)$ to *parent*	

message m	**simulator**	**coordinator**

$(y, from, t)$		$\forall i \in I_{from} \setminus \{self\}$:
		  send $(Z_{from,i}(y), from, t)$ to i
		  $active\_children \leftarrow active\_children \cup \{i\}$
		if $self \in I_{from}$:
		  send $(Z_{from,self}(y), self, t)$ to $parent$

$(done, from, t)$		$active\_children \leftarrow active\_children \setminus \{from\}$ **if** $active\_children = \emptyset$: $t_L \leftarrow t$ $t_N \leftarrow min\{M_i.t_N \mid i \in D\}$ **send** $(done, self, t_N)$ to $parent$

DEVS simulator main loop

```
$t \leftarrow t_N$ of topmost coordinator
repeat until $t \geq t_{end}$ (or some other termination condition)
    send $(*, meta, t)$ to topmost coupled model $top$
    wait for $(done, top, t_N)$
    $t \leftarrow t_N$
```