Discrete Event Simulation

SYS 611: Systems Modeling and Simulation

Paul T. Grogan, Ph.D.
Assistant Professor
School of Systems and Enterprises

Agenda

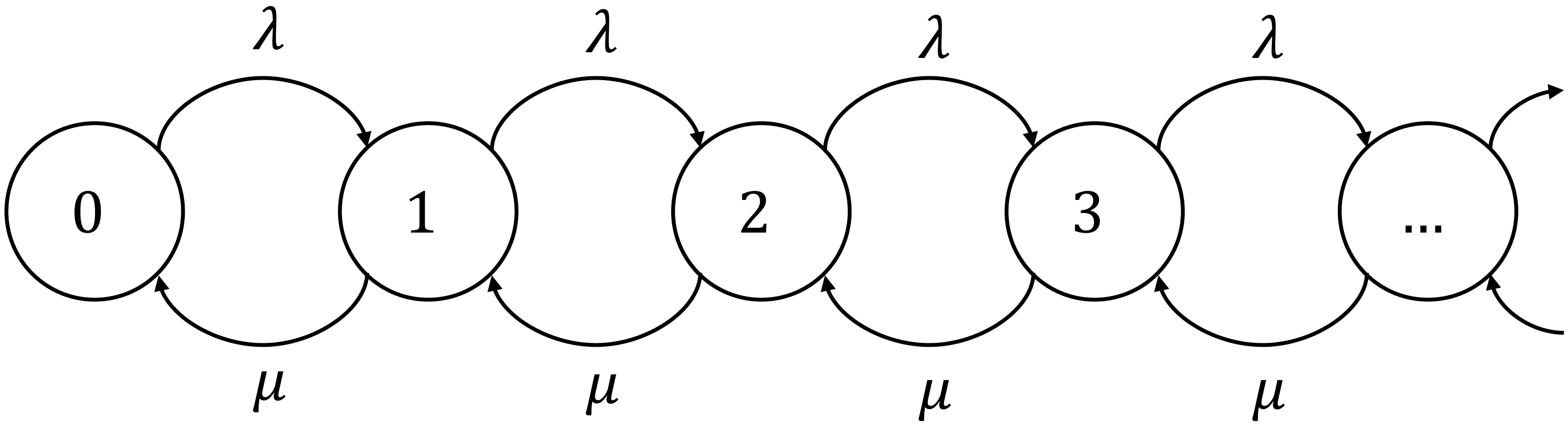
1. Queuing Model, Revisited
2. DES Model Constructs

Reading: S.M. Ross "The Discrete Event Simulation Approach," Ch. 7 in *Simulation*, 5th Edition, 2013.

Queuing Model, Revisited

Queuing Model Review

- State $q(t)$: number customers in system at time t
- Inter-arrival periods (arrivals), $X \sim \text{exponential}(\lambda)$
- Service times (departures), $Y \sim \text{exponential}(\mu)$



M/M/1 Queuing Theory

- Steady-state assumption helps to solve for the stationary stochastic distribution P_i in terms of P_0
- Geometric series converges to analytical result

$$\bar{L}_q = \frac{\lambda^2}{\mu(\mu-\lambda)}, \qquad \bar{L} = \frac{\lambda}{\mu-\lambda}$$

$$\bar{W}_q = \frac{\lambda}{\mu(\mu-\lambda)}, \qquad \bar{W} = \frac{1}{\mu-\lambda}$$

M/M/1 Model Simulation

- For each event i:
  - $x = -\ln(1 - r_x)/\lambda$
  - $y = -\ln(1 - r_y)/\mu$
  - if $q(i) = 0$ or $x < y$:
    - $\Delta t = x$
    - $q(i+1) = q(i) + 1$
  - else:
    - $\Delta t = y$
    - $q(i+1) = q(i) - 1$
  - $t(i+1) = t(i) + \Delta t$

- Exponentially-distributed arrivals, service times
  - Independent with long-term average rates
  - Memoryless: can resample x, y for every event
- Single queue, single server
  - Simplifies state representation
- No human behaviors (reneging, balking, etc.)
  - Simplifies state transitions

G/G/1 Model Simulation

- For general distributions (G) without the *memoryless* property:
  - Cannot simply re-sample x and y every event
  - Once sampled, x and y must be recorded and used
- Generate x and y in advance for each customer
  - Acceptable if x and y are independent of simulation state
    - x is *not* a function of time (no time-varying arrival rate)
    - y is *not* a function of time or queue length
  - "Customer-centric" simulation model

Customer-centric Simulation

Index state variables by customer i instead of event i

t_{enter}^{i}	time customer i enters the system
L_{q}^{i}	length of the queue when customer i enters
t_{served}^{i}	time customer i service starts
W_{q}^{i}	waiting time in the queue for customer i
t_{exit}^{i}	time customer i exists the system
W_{i}	total waiting time for customer i

Customer-based Simulation

x: Inter-arrival Period
y: Service Time
t_{enter}: Entry Time
L_q: Queue Length
t_{served}: Time Served
W_q: Queue Wait Time
t_{exit}: Exit Time
W: Total Wait Time

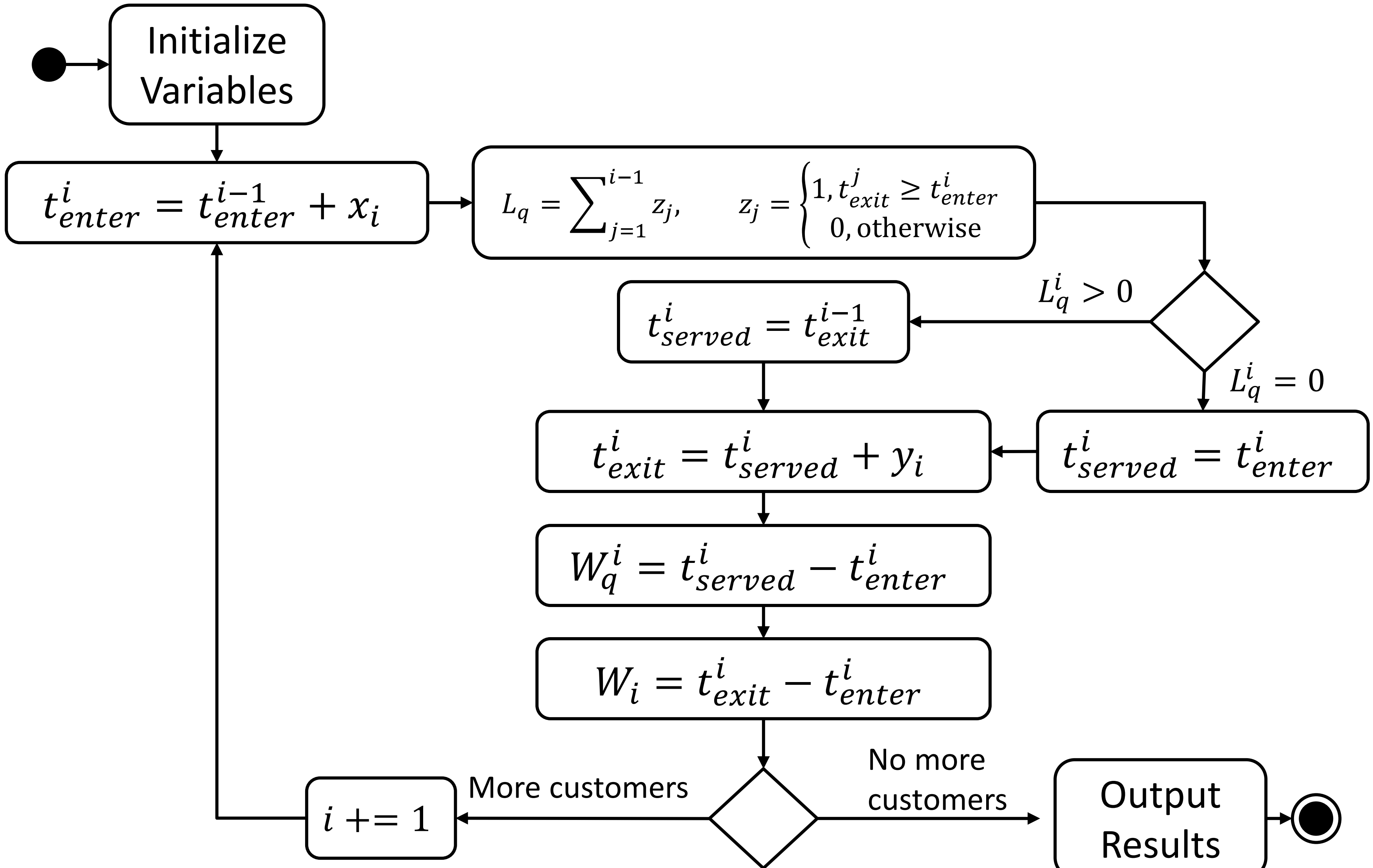
i	x	t_{enter}	L_q	t_{served}	W_q	y	t_{exit}	W
1	**0.43**					**0.90**		
2	**4.49**					**1.66**		
3	**0.95**					**0.02**		
4	**0.03**					**0.46**		
5	**0.28**					**0.51**		

Customer-based Simulation

x: Inter-arrival Period t_{enter}: Entry Time t_{served}: Time Served t_{exit}: Exit Time
y: Service Time L_q: Queue Length W_q: Queue Wait Time W: Total Wait Time

i	x	t_{enter}	L_q	t_{served}	W_q	y	t_{exit}	W
1	**0.43**	0.00+0.43= 0.43	0	0.43	0.43-0.43= 0	**0.90**	0.43+0.90= 1.33	1.33-0.43= 0.90
2	**4.49**	0.43+4.49= 4.92	0	4.92	4.92-4.92= 0	**1.66**	4.92+1.66= 6.58	6.58-4.92= 1.66
3	**0.95**	4.92+0.95= 5.87	1	6.58	6.58-5.87= 0.71	**0.02**	6.58+0.02= 6.60	6.60-5.87= 0.73
4	**0.03**	5.87+0.03= 5.90	2	6.60	6.60-5.90= 0.70	**0.46**	6.60+0.46= 7.06	7.06-5.90= 1.16
5	**0.28**	5.90+0.28= 6.18	3	7.06	7.06-6.18= 0.88	**0.51**	7.06+0.51= 7.57	7.57-6.18= 1.39

Customer-based Sim Behavior



Customer-based Sim (Excel)

	A	B	C	D	E	F	G	H	I	J	
1	Customer	x	t_enter	L_q	t_served	y	t_exit	W_q	W		L_q
2	1	0.44	0.44	0	0.44	2.21	2.64	2.21	2.21		
3	2	4.25	4.69	0	4.69	0.13	4.81	0.13	0.13		
4	3	0.74	=B4+C3	0	5.43	0.57	6.00	0.57	0.57		

$$t_{enter}^{i} = t_{enter}^{i-1} + x_i$$

	A	B	C	D	E	F	G	H	I
1	Customer	x	t_enter	L_q	t_served	y	t_exit	W_q	W
2	1	0.44	0.44	0	0.44	2.21	2.64	2.21	2.21
3	2	4.25	4.69	0	4.69	0.13	4.81	0.13	0.13
4	3	0.74	5.43	=COUNTIF(G2:G3,">="&C4)			6.00	0.57	0.57

$$L_q = \sum_{j=1}^{i-1} z_j, \qquad z_j = \begin{cases} 1, t_{exit}^{j} \geq t_{enter}^{i} \\ 0, \text{otherwise} \end{cases}$$

	A	B	C	D	E	F	G	H	I
1	Customer	x	t_enter	L_q	t_served	y	t_exit	W_q	W
2	1	0.39	0.39	0	0.39	0.71	1.11	0.71	0.71
3	2	0.24	0.63	1	1.11	0.24	1.35	0.24	0.71
4	3	3.05	3.68	0	=IF(D4=0,C4,G3)		3.73	0.05	0.05
5	4	1.91	5.59	0	5.59	0.44	6.03	0.44	0.44

$L_q^i > 0$: $t_{served}^{i} = t_{exit}^{i-1}$

$L_q^i = 0$: $t_{served}^{i} = t_{enter}^{i}$

	A	B	C	D	E	F	G	H	I	J	
1	Customer	x	t_enter	L_q	t_served	y	t_exit	W_q	W		L_q
2	1	0.92	0.92	0	0.92	0.64	1.55	0.64	0.64		
3	2	0.39	1.31	1	1.55	0.26	1.82	0.26	0.51		
4	3	2.13	3.44	0	3.44	1.69	=F4+E4	1.69	1.69		

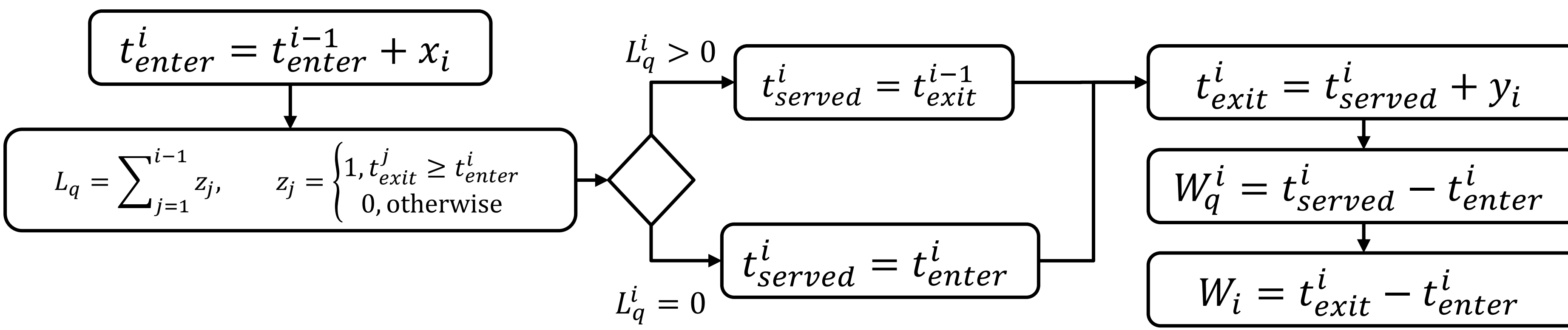
$$t_{exit}^{i} = t_{served}^{i} + y_i$$

	A	B	C	D	E	F	G	H	I	J
1	Customer	x	t_enter	L_q	t_served	y	t_exit	W_q	W	
2	1	1.08	1.08	0	1.08	1.50	2.58	0.00	1.50	
3	2	0.65	1.74	1	2.58	0.95	3.54		1.80	
4	3	0.32	2.06	2	3.54	0.21	3.75	=E4-C4	1.69	

$$W_q^i = t_{served}^{i} - t_{enter}^{i}$$

Customer-based Sim (Python)

```
for i in range(num_customers):
    t_enter[i] = t_enter[i-1] + x[i] if i > 0 else x[i]
    q_length[i] = np.sum(t_exit[0:i] > t_enter[i]) if i > 0 else 0
    if q_length[i] > 0:
        t_served[i] = t_exit[i-1]
    else:
        t_served[i] = t_enter[i]
    t_exit[i] = t_served[i] + y[i]
    wait_q[i] = t_served[i] - t_enter[i]
    wait_total[i] = t_exit[i] - t_enter[i]
```



Customer-based Simulation

$\lambda = 2/3$ $\mu = 4/3$

1000-customer Sim.:

- $\overline{W} = 1.49$ min.
- $\overline{W}_q = 0.74$ min.
- $\overline{L}_q{}^* = 1.05$
- $\overline{L}^* = \overline{L}_q{}^* + 1 = 2.05$

Queuing Theory:

- $\overline{W} = \frac{1}{\mu - \lambda} = 1.50$ min.
- $\overline{W}_q = \frac{\lambda}{\mu(\mu - \lambda)} = 0.75$ min.
- $\overline{L}_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = 0.50$
- $\overline{L} = \frac{\lambda}{\mu - \lambda} = 1.00$

- Length observations *only* recorded at arrivals
- Biased sample, more customers arrive in long lines (by definition)
- Issue of "**random incidence**": need to take independent samples

Customer-based Simulation

Strengths

- Can use arbitrary process generators for all arrival times and service times (if state independent)
- Straight-forward, intuitive state updating

Limitations

- Need to maintain history of all customers
- Difficult to calculate time-average results
- Difficult to extend to new problems:
  - Balking, reneging

DES Model Constructs

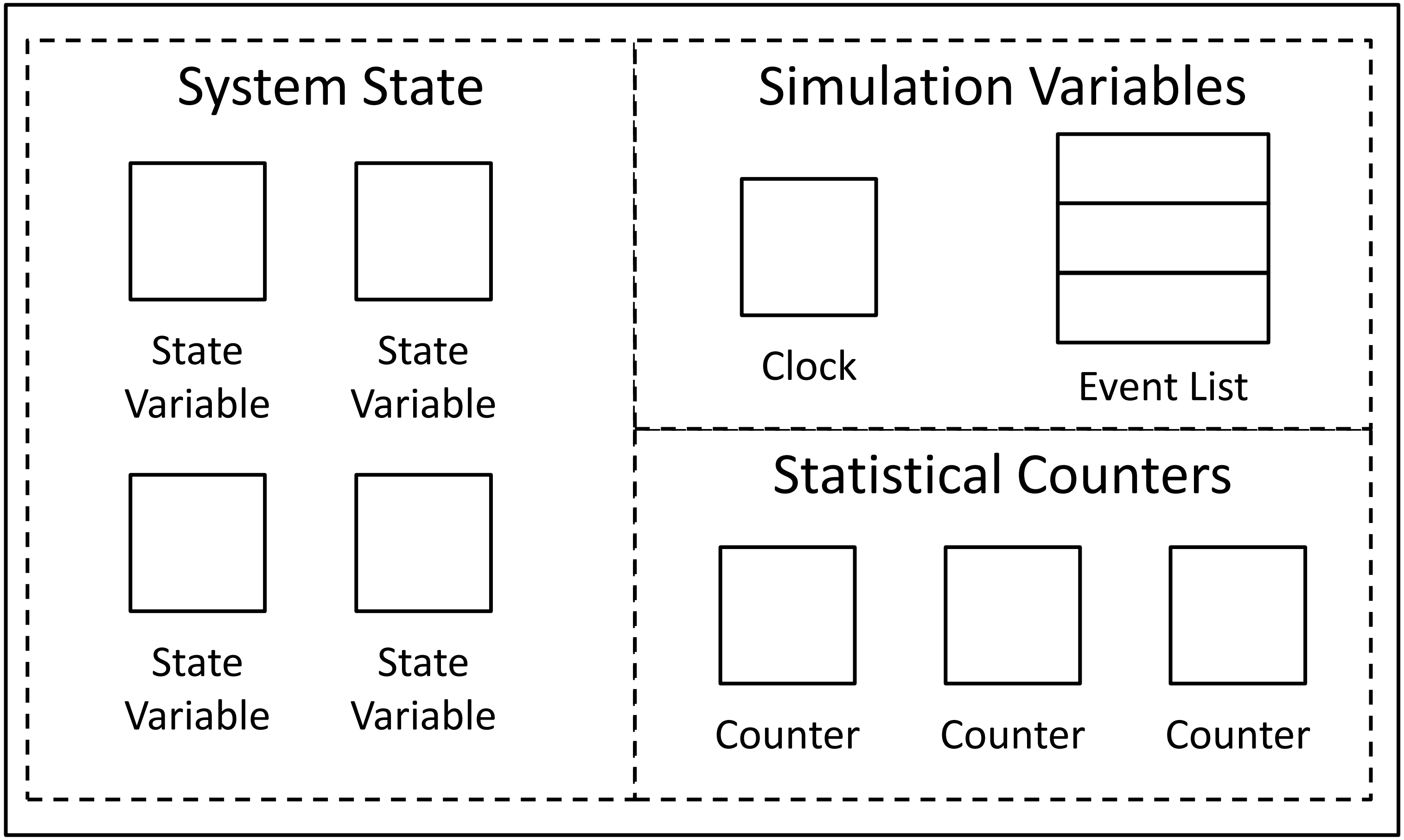
Discrete Event Simulation

- **Discrete event simulation (DES)** is a modeling approach which represents state transitions at "interesting" instants in time (events)
  - Efficiently models systems with occasional state transitions (e.g., logistics, operations)
  - Example: queuing model with general (non-Markov) arrivals and services
- Discrete time simulation is a special type of DES with evenly-distributed events (clock ticks)

DES Model Constructs

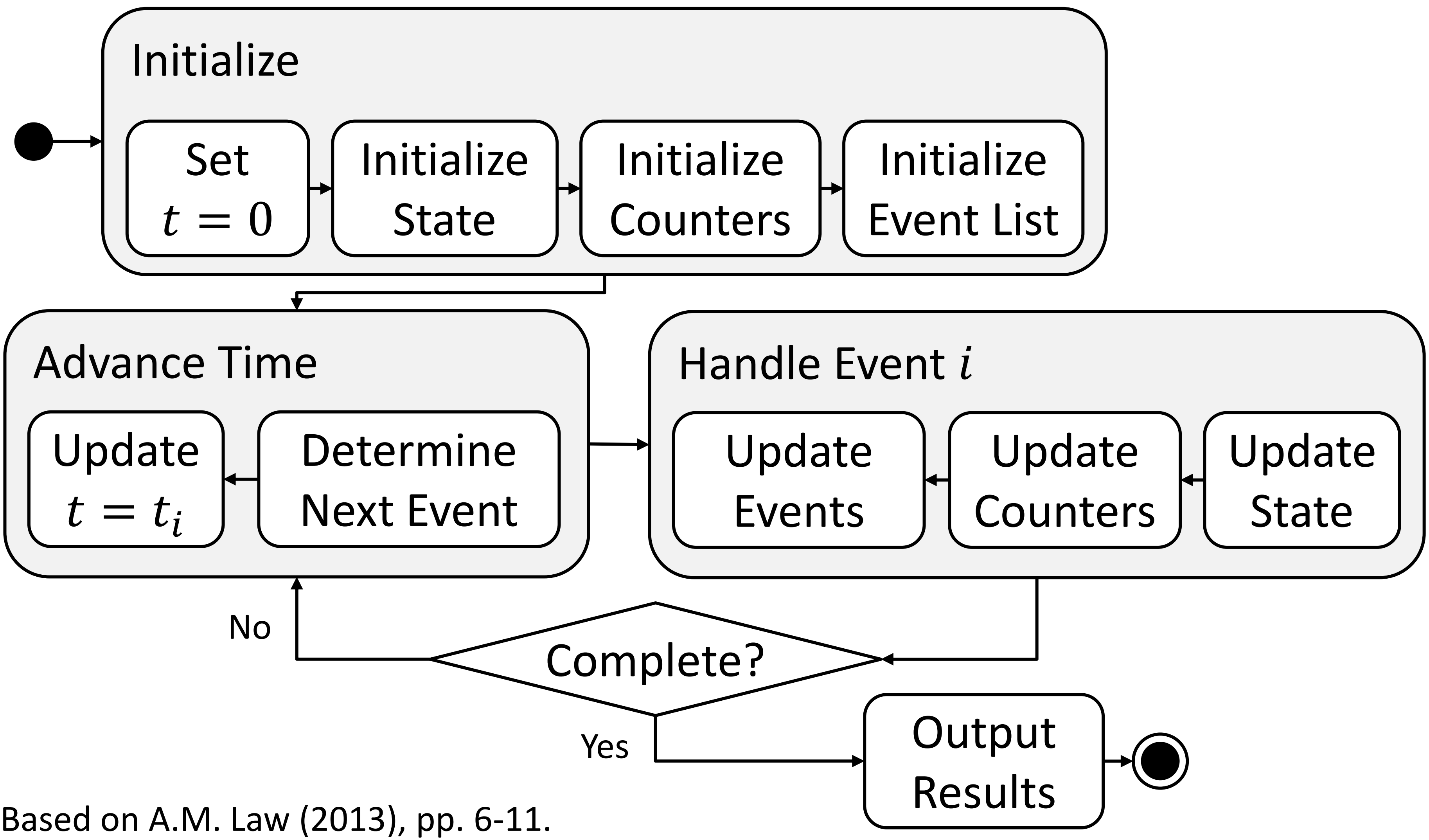
- **Time variable**: Store the current simulation time
- **Event list/stack**: List of future events with associated execution times
- **System state variables**: Store state variables which persist across time
- **Counter variables**: Store derived state variables that record useful observations for analysis

DES Model Structure



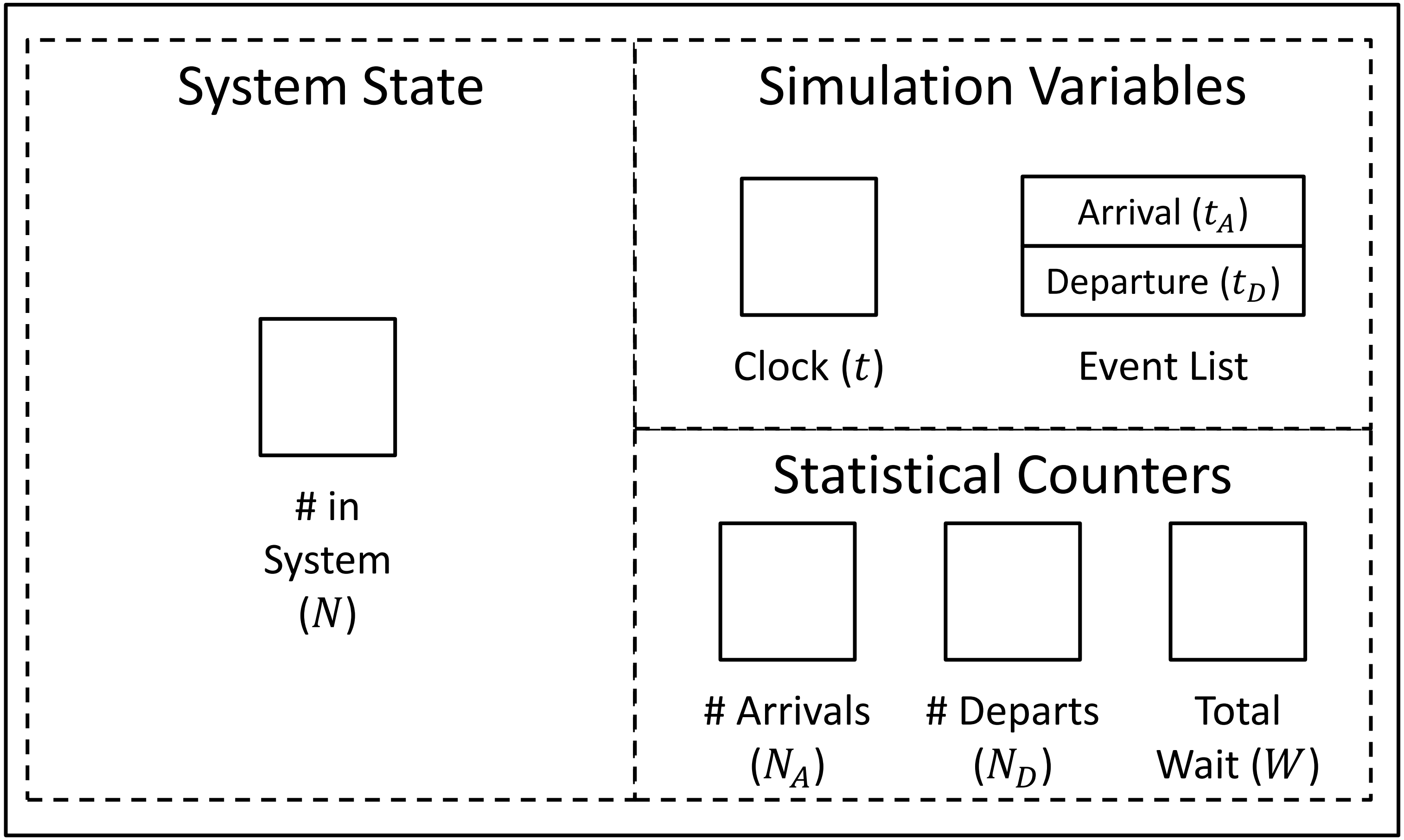
Based on A.M. Law (2013), pp. 6-11.

DES Model Behavior



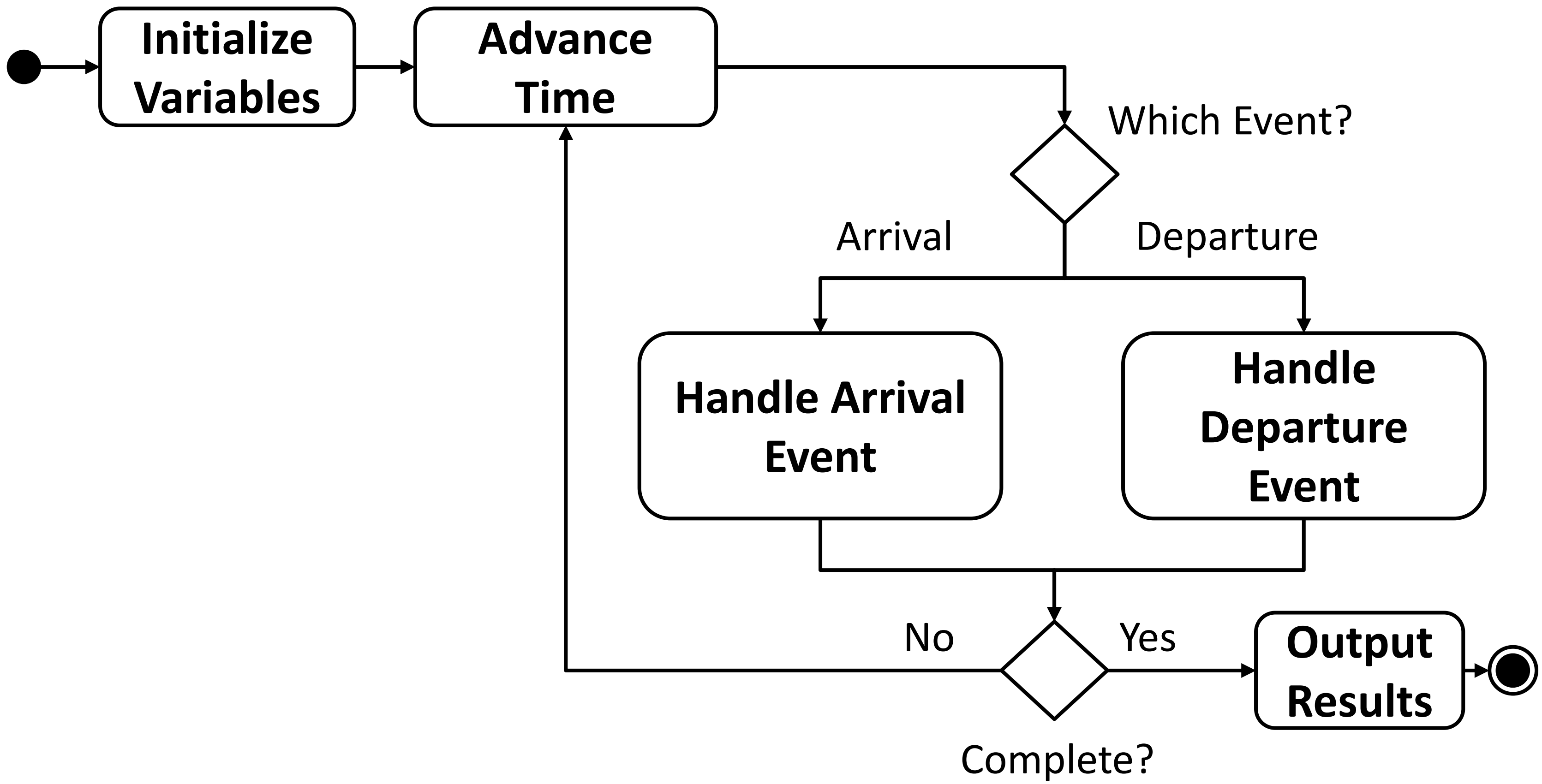
Based on A.M. Law (2013), pp. 6-11.

Event-based Queuing Model

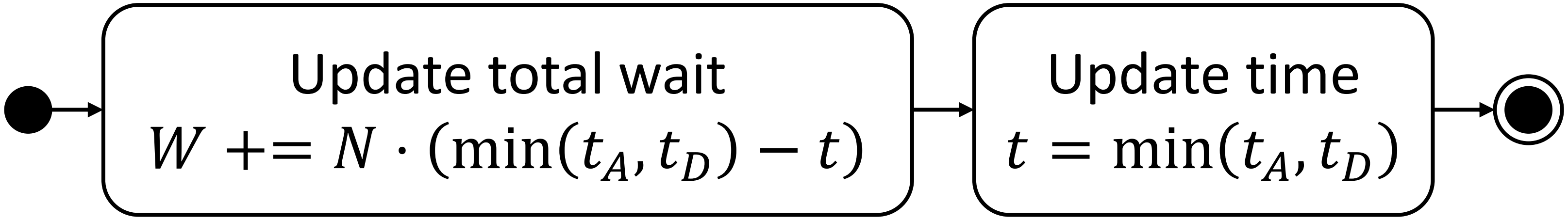


Based on A.M. Law (2013), pp. 12-27.

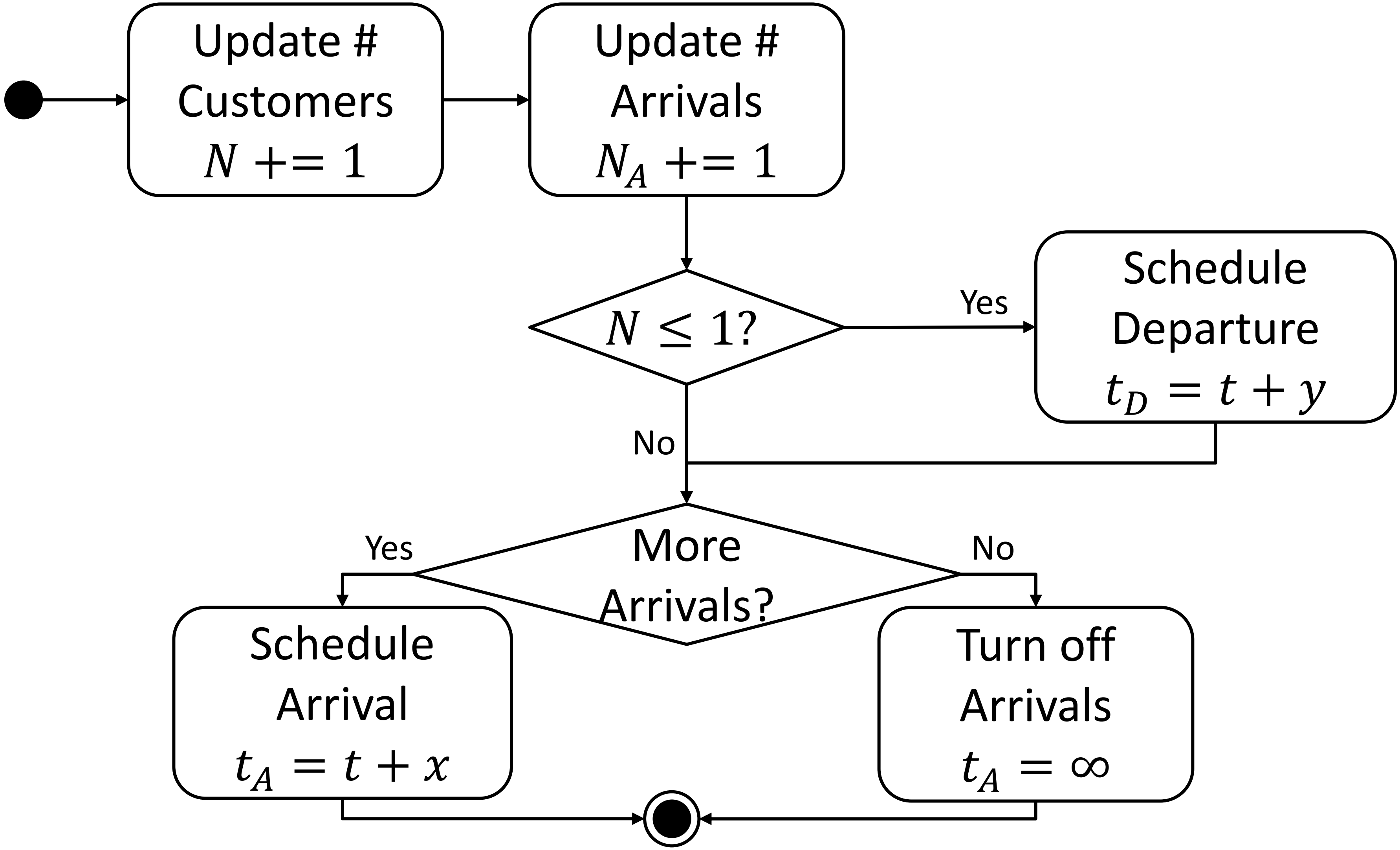
Queuing Activity Diagram



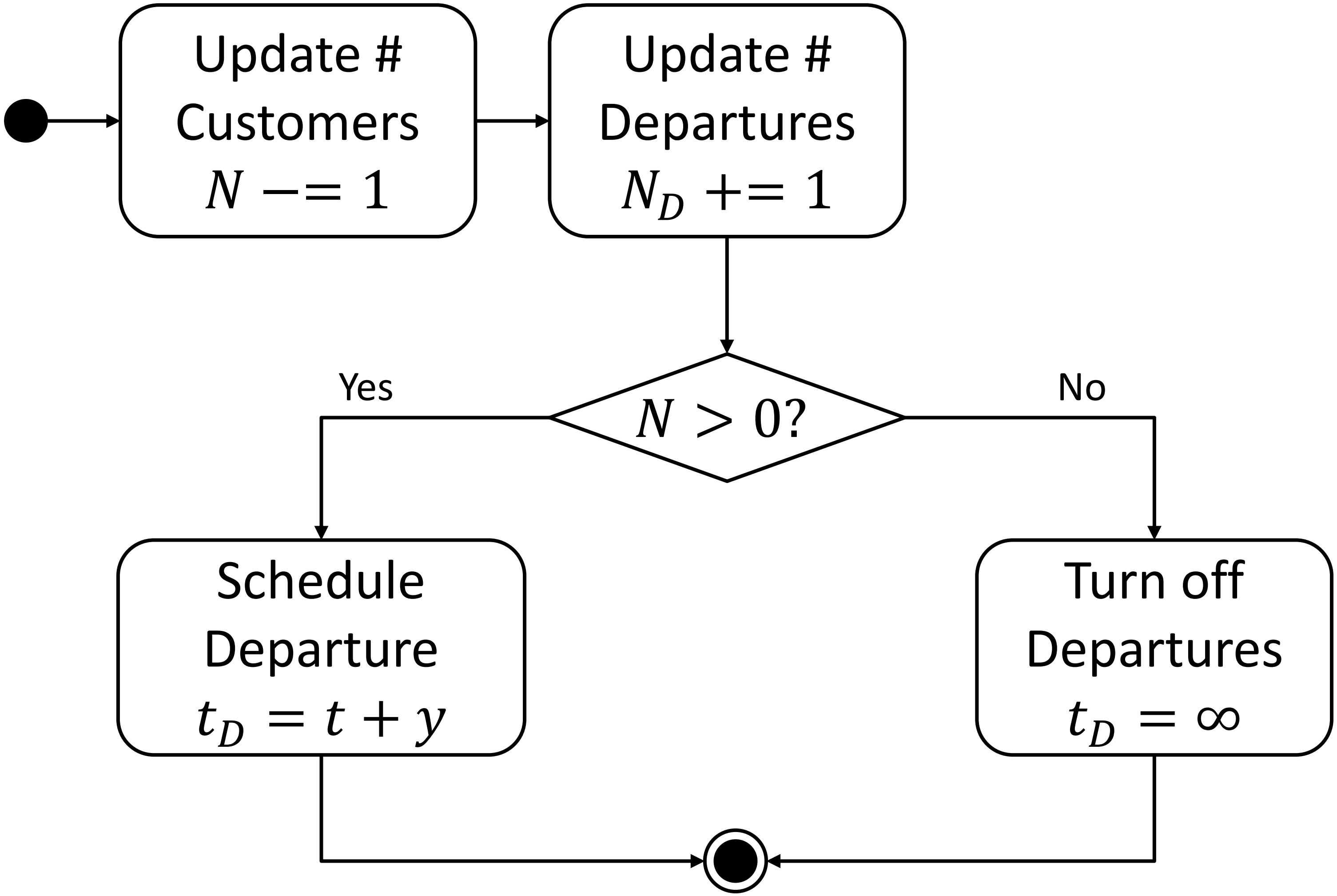
Advance Time



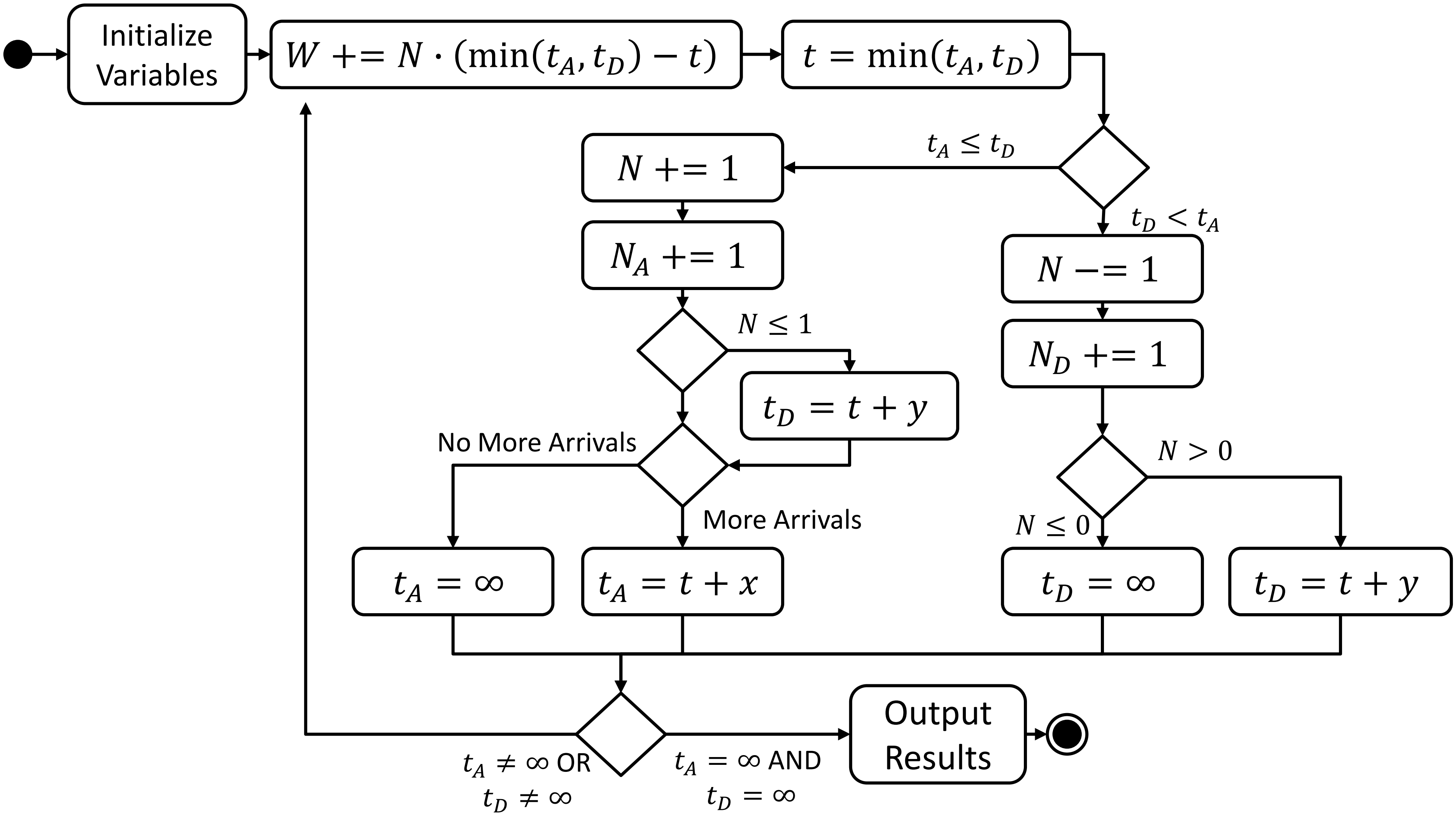
Handle Arrival Event



Handle Departure Event

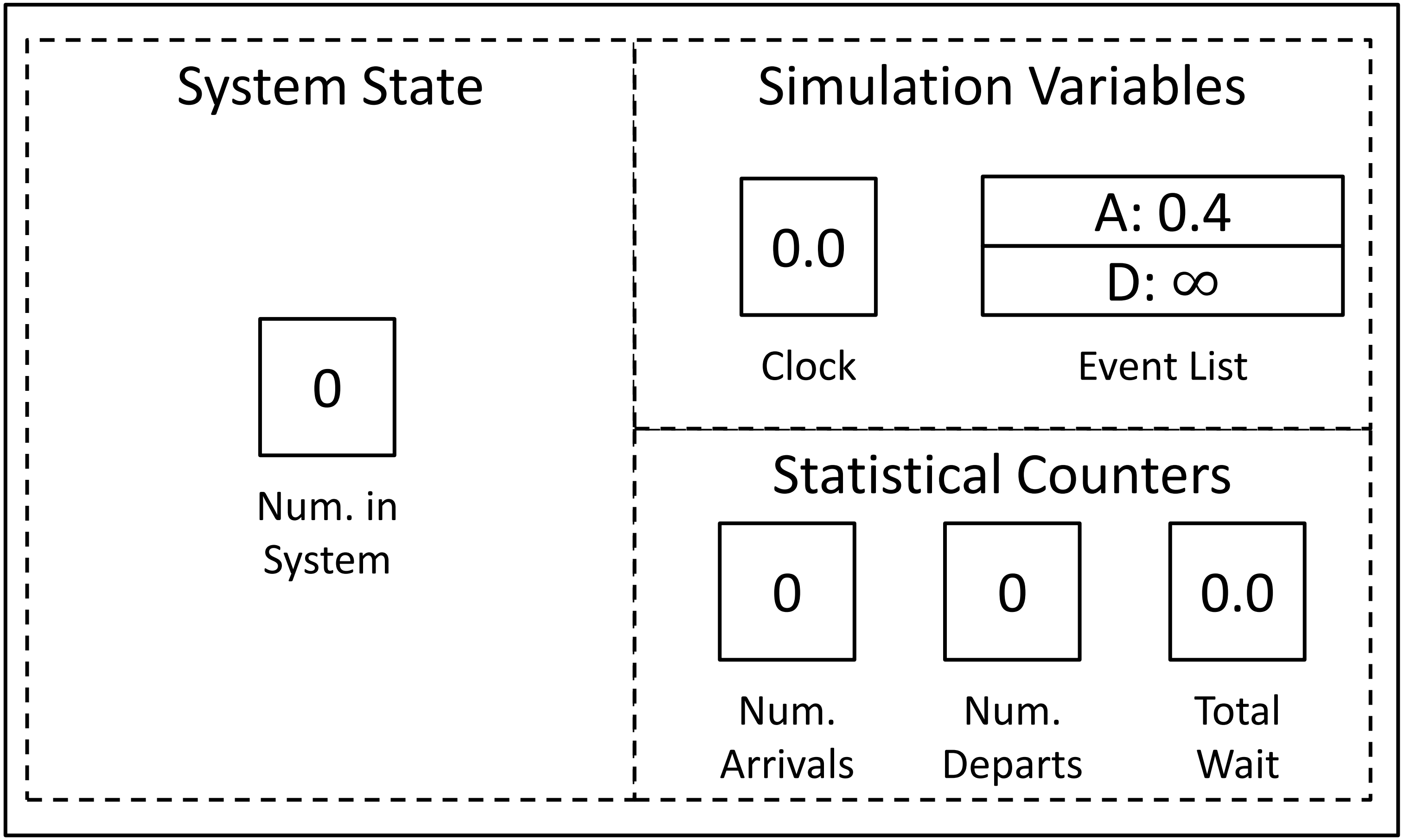


Complete Activity Diagram



Initialize Simulation

Inter-arrival times: 0.4, 1.2, 0.5, 1.7 Service times: 2.0, 0.7, 0.2, 1.1



Based on A.M. Law (2013), pp. 12-27.

Arrival Event @ $t = 0.4$

Inter-arrival times: ~~0.4~~, 1.2, 0.5, 1.7

Service times: 2.0, 0.7, 0.2, 1.1

System State

Num. in System: ~~0~~ 1

Simulation Variables

Clock: ~~0.0~~ 0.4

Event List:
- A: ~~0.4~~ 1.6
- D: ~~∞~~ 2.4

Statistical Counters

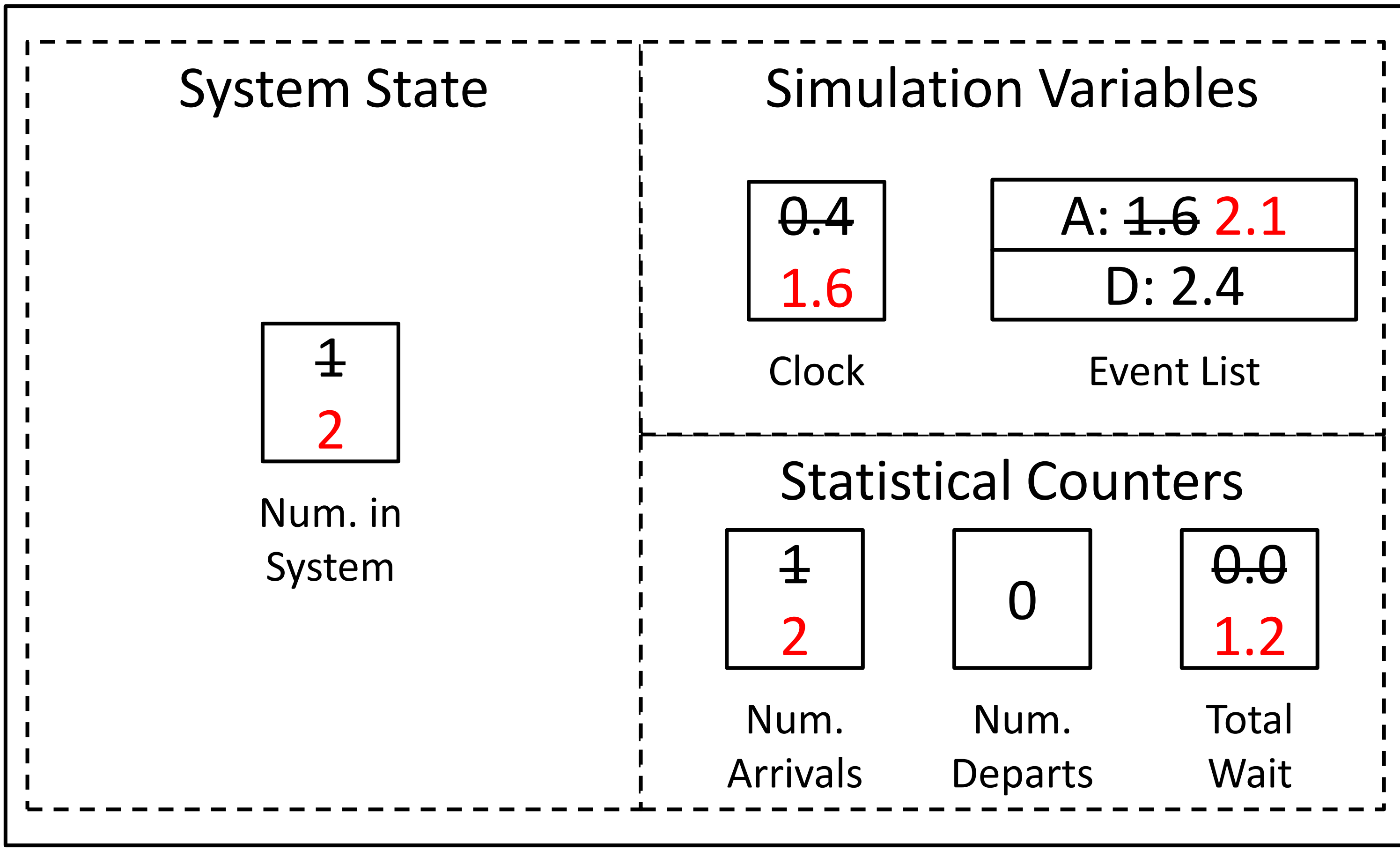
Num. Arrivals	Num. Departs	Total Wait
~~0~~ 1	0	0.0

Based on A.M. Law (2013), pp. 12-27.

Arrival Event @ $t = 1.6$

Inter-arrival times: ~~0.4, 1.2~~, 0.5, 1.7

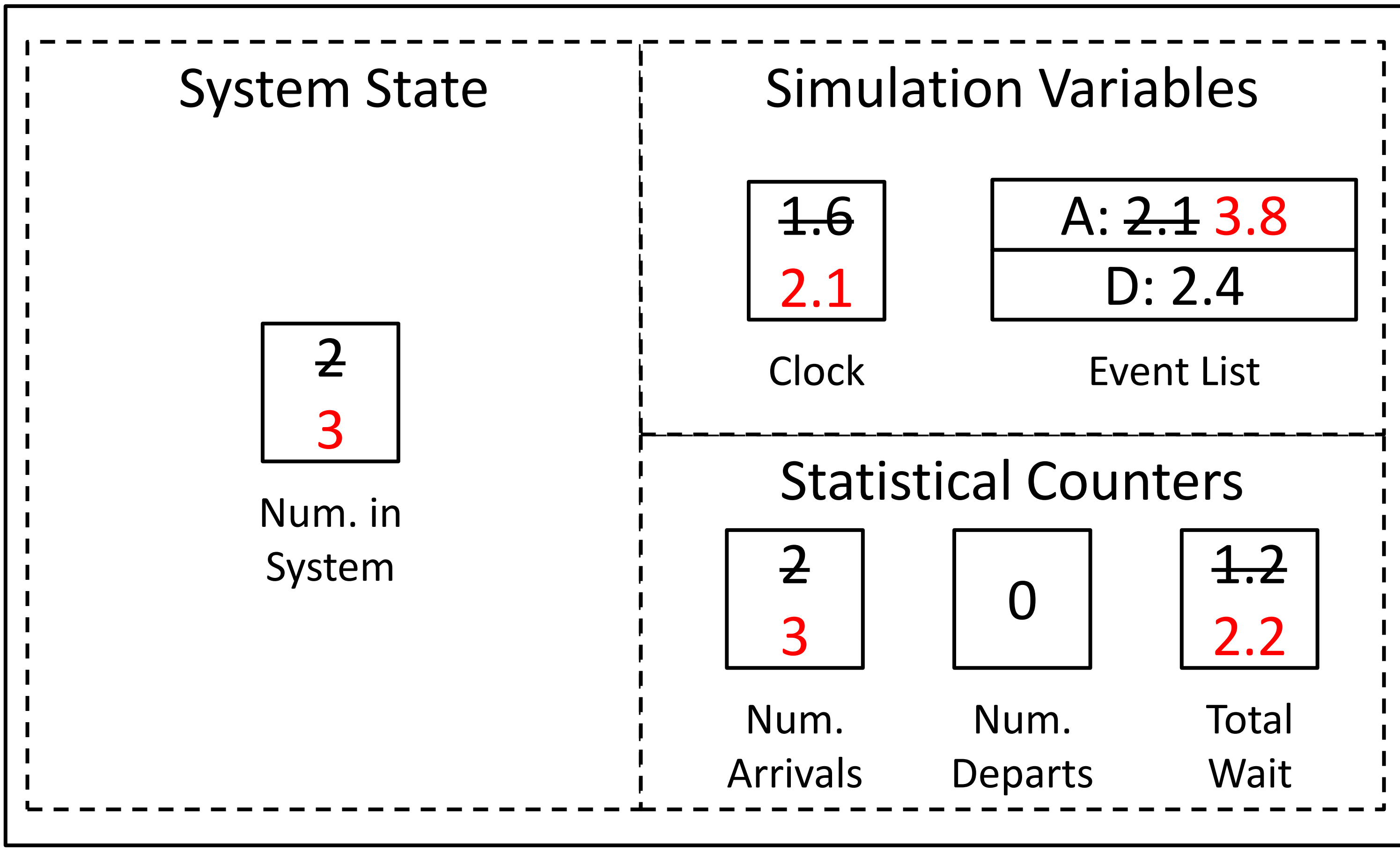
Service times: ~~2.0~~, 0.7, 0.2, 1.1



Based on A.M. Law (2013), pp. 12-27.

Arrival Event @ $t = 2.1$

Inter-arrival times: ~~0.4, 1.2, 0.5~~, 1.7   Service times: ~~2.0~~, 0.7, 0.2, 1.1



Based on A.M. Law (2013), pp. 12-27.

Departure Event @ $t = 2.4$

Inter-arrival times~~: 0.4, 1.2, 0.5, 1.7~~ Service times: ~~2.0~~, 0.7, 0.2, 1.1

System State

Num. in System: ~~3~~ 2

Simulation Variables

Clock: ~~2.1~~ 2.4

Event List:
- A: 3.8
- D: ~~2.4~~ 3.1

Statistical Counters

- Num. Arrivals: 3
- Num. Departs: ~~0~~ 1
- Total Wait: ~~2.2~~ 3.1

Based on A.M. Law (2013), pp. 12-27.

Departure Event @ $t = 3.1$

Inter-arrival times: ~~0.4, 1.2, 0.5, 1.7~~

Service times: ~~2.0, 0.7~~, 0.2, 1.1

System State

~~2~~ 1

Num. in System

Simulation Variables

Clock: ~~2.4~~ 3.1

Event List:
- A: 3.8
- D: ~~3.1~~ 3.3

Statistical Counters

Num. Arrivals	Num. Departs	Total Wait
3	~~1~~ 2	~~3.1~~ 4.5

Based on A.M. Law (2013), pp. 12-27.

Departure Event @ $t = 3.3$

Inter-arrival times: ~~0.4, 1.2, 0.5, 1.7~~

Service times: ~~2.0, 0.7, 0.2~~, 1.1

System State

Num. in System: ~~1~~ 0

Simulation Variables

Clock: ~~3.1~~ 3.3

Event List:
- A: 3.8
- D: ~~3.3~~ ∞

Statistical Counters

- Num. Arrivals: 3
- Num. Departs: ~~2~~ 3
- Total Wait: ~~4.5~~ 4.7

Based on A.M. Law (2013), pp. 12-27.

Arrival Event @ $t = 3.8$

Inter-arrival times: ~~0.4, 1.2, 0.5, 1.7~~

Service times: ~~2.0, 0.7, 0.2, 1.1~~

System State

Num. in System: ~~0~~ 1

Simulation Variables

Clock: ~~3.3~~ 3.8

Event List:
- A: ~~3.8~~ ∞
- D: ~~∞~~ 4.9

Statistical Counters

Num. Arrivals	Num. Departs	Total Wait
~~3~~ 4	3	4.7

Based on A.M. Law (2013), pp. 12-27.

Departure Event @ $t = 4.9$

Inter-arrival times: ~~0.4, 1.2, 0.5, 1.7~~ Service times: ~~2.0, 0.7, 0.2, 1.1~~

System State

Num. in System: ~~1~~ 0

Simulation Variables

Clock: ~~3.8~~ 4.9

Event List:
- A: ∞
- D: ~~4.9~~ ∞

Statistical Counters

Num. Arrivals	Num. Departs	Total Wait
4	~~3~~ 4	~~4.7~~ 5.8

Based on A.M. Law (2013), pp. 12-27.

Event-based Simulation

x : 0.4, 1.2, 0.5, 1.7

y : 2.0, 0.7, 0.2, 1.1

t: Clock Time
t_A: Next Arrival Time
t_D: Next Departure Time
N: Number Customers in System

N_A: Number Arrivals (Cumulative)
N_D: Number Departures (Cumulative)
W: Total Wait Time (Cumulative)

i	t	t_A	t_D	N	N_A	N_D	W
1	0.0	0.4	∞	0	0	0	0
2	0.4	0.4+1.2= 1.6	0.4+2.0= 2.4	0+1= 1	0+1= 1	0	0
3	1.6	1.6+0.5= 2.1	2.4	1+1= 2	1+1= 2	0	0+1*1.2= 1.2
4	2.1	2.1+1.7= 3.8	2.4	2+1= 3	2+1= 3	0	1.2+2*0.5= 2.2
5	2.4	3.8	2.4+0.7= 3.1	3-1= 2	3	0+1= 1	2.2+3*0.3= 3.1

Event-based Simulation

x : 0.4, 1.2, 0.5, 1.7

y : 2.0, 0.7, 0.2, 1.1

t: Clock Time

t_A: Next Arrival Time

t_D: Next Departure Time

N: Number Customers in System

N_A: Number Arrivals (Cumulative)

N_D: Number Departures (Cumulative)

W: Total Wait Time (Cumulative)

i	t	t_A	t_D	N	N_A	N_D	W
5	2.4	3.8	2.4+0.7= 3.1	3-1= 2	3	0+1= 1	2.2+3*0.3= 3.1
6	3.1	3.8	3.1+0.2= 3.3	2-1= 1	3	1+1= 2	3.1+2*0.7= 4.5
7	3.3	3.8	∞	1-1= 0	3	2+1= 3	4.5+1*0.2= 4.7
8	3.8	∞	3.8+1.1= 4.9	0+1= 1	3+1= 4	3	4.7
9	4.9	∞	∞	1-1= 0	4	3+1= 4	4.7+1*1.1= 5.8

Customer- vs. Event-based Sim

i	x	t_{enter}	L_q	t_{served}	W_q	y	t_{exit}	W
1	0.4	0.4	0	0.4	0.0	2.0	2.4	2.0
2	1.2	1.6	1	2.4	0.8	0.7	3.1	1.5
3	0.5	2.1	2	3.1	1.0	0.2	3.3	1.2
4	1.7	3.8	0	3.8	0.0	1.1	4.9	1.1

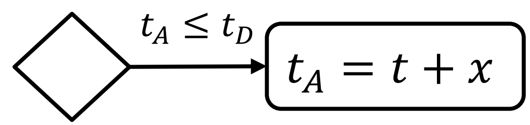
i	t	t_A	t_D	N	N_A	N_D	W
1	0.0	0.4	∞	0	0	0	0
2	0.4	1.6	2.4	1	1	0	0
3	1.6	2.1	2.4	2	2	0	1.2
4	2.1	3.8	2.4	3	3	0	2.2
5	2.4	3.8	3.1	2	3	1	3.1
6	3.1	3.8	3.3	1	3	2	4.5
7	3.3	3.8	∞	0	3	3	4.7
8	3.8	∞	4.9	1	4	3	4.7
9	4.9	∞	∞	0	4	4	5.8

Event-based Sim (Excel)

	A	B	C	D	E	F	G	H
1	Event	t	t_A	t_D	N	N_A	N_D	W
2	0	0	0.33	9999.00	0	0	0	0.00
3	1	=MIN(C2:D2)		0.80	1	1	0	0.00

$$t_e = \min(t_A, t_D)$$

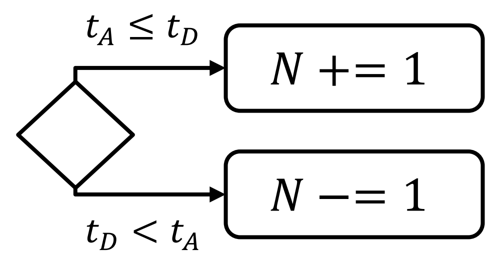
	A	B	C	D	E	F	G	H
1	Event	t	t_A	t_D	N	N_A	N_D	W
2	0	0	0.77	9999.00	0	0	0	0.00
3	1	0.77	=IF(C2<=D2,B3-1.5*LN(1-RAND()),C2)				0	0.00



	A	B	C	D	E	F	G	H	I	
1	Event	t	t_A	t_D	N	N_A	N_D	W		W_b
2	0	0	0.04	9999.00	0	0	0	0.00		
3	1	0.04	0.88	=IF(OR(AND(C2<=D2,E2+1<=1),AND(D2<C2,E2-1>0)),B3-0.75*LN(1-RAND()),IF(AND(D2<						
4	2	0.88	7.96	C2,E2-1<=0),9999,D2))						

t_D is complicated!!
(See next slide)

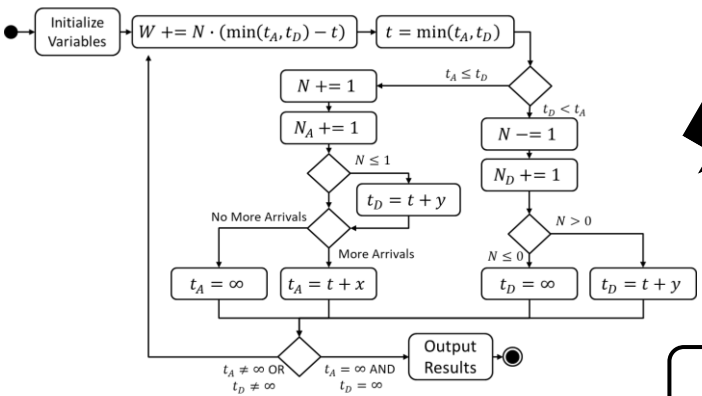
	A	B	C	D	E	F	G	H
1	Event	t	t_A	t_D	N	N_A	N_D	W
2	0	0	2.88	9999.00	0	0	0	0.00
3	1	2.88	3.00	4.63	=IF(C2<=D2,E2+1,E2-1)			0.00
4	2	3.00	4.62	4.63	2	2	0	0.12

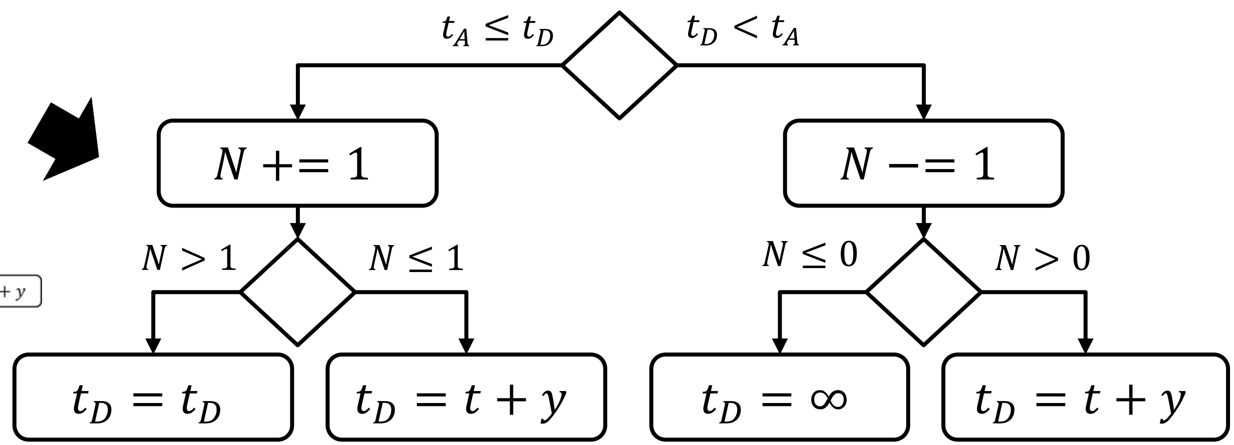


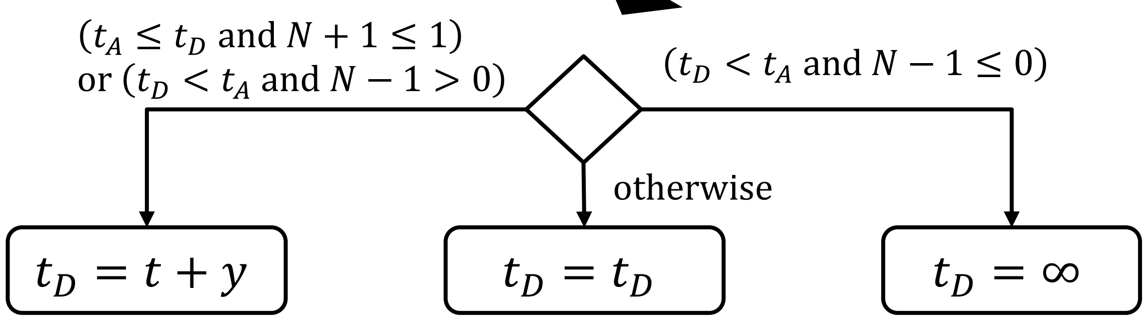
	A	B	C	D	E	F	G	H	I
1	Event	t	t_A	t_D	N	N_A	N_D	W	
2	0	0	2.56	9999.00	0	0	0	0.00	
3	1	2.56	4.36	3.48	1	1	0	=H2+E2*(B3-B2)	
4	2	3.48	4.36	9999.00	0	1	1	0.93	

$$W \mathrel{+}= N \cdot (\min(t_A, t_D) - t)$$

Departure Time Updates



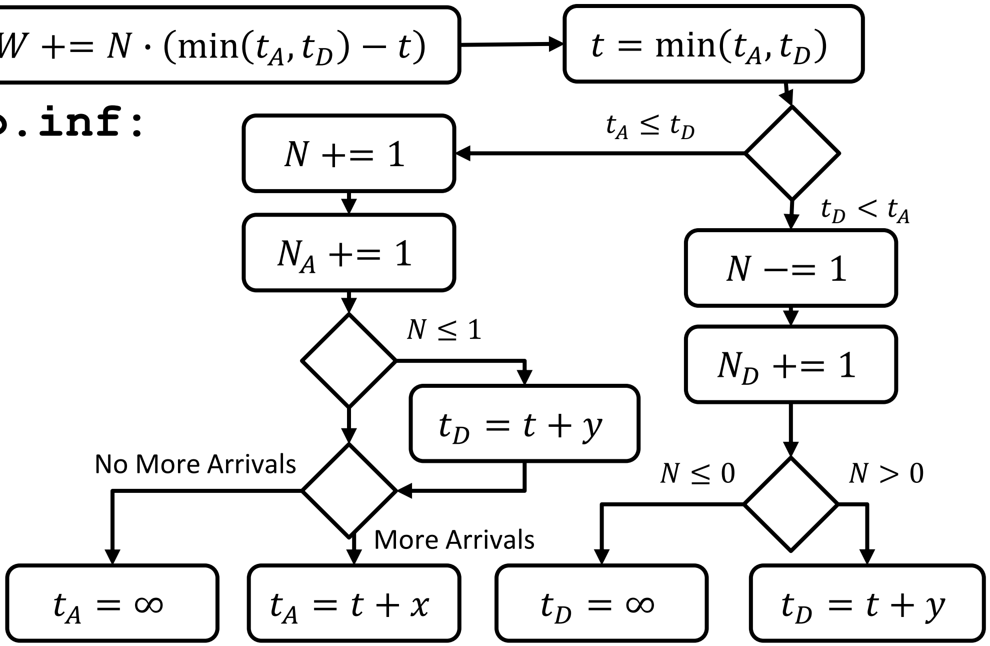




	A	B	C	D	E	F	G	H	I	J	K
1	Event	t	t_A	t_D	N	N_A	N_D	W		W_bar	
2	0	0	0.04	9999.00	0	0	0	0.00		1.84	
3	1	0.04	0.88	=IF(OR(AND(C2<=D2,E2+1<=1),AND(D2<C2,E2-1>0)),B3-0.75*LN(1-RAND()),IF(AND(D2<							
4	2	0.88	7.96	C2,E2-1<=0),9999,D2))							

Event-based Sim (Python)

```
while t_A < np.inf or t_D < np.inf:
  W += N*(min(t_A, t_D) - t)
  t = min(t_A, t_D)
  if t_A <= t_D:
    N += 1
    N_A += 1
    if N <= 1:
      t_D = t + generate_y()
    t_A = t + generate_x() if t < 1000 else np.inf
  else:
    N -= 1
    N_D += 1
    t_D = t + generate_y() if N > 0 else np.inf
```



Discrete Event Simulation

Strengths

- Generate random variables exactly when needed
  - Function of time/state
- Scales well to many types of events
- Only need state from previous time step

Limitations

- Difficult to calculate time-average results (need counter variables)
- State updates can be complex for atomic functions (Excel)