General Principles & Concepts in Discrete Event Simulation

Grishma Sharma
KJSCE

GENERAL PRINCIPLES

Introduction

- This chapter develops a common framework for the Modeling of complex systems using discrete-event simulation.

- It covers the basic building blocks of all discrete-event simulation models: entities and attributes, activities and events.

- In discrete-event simulation, a system is modeled in terms of its state at each point in time; the entities that pass through the system and the entities that represent system resources; and the activities and events that cause system state to change.

- It deals exclusively with dynamic, stochastic system (i.e. involving time and containing random elements) which changes in a discrete manner.

Concepts in Discrete-Event Simulation

- **System:** A collection of entities (e.g., people and machines)
- **Model:** An abstract representation of a system, usually containing structural, logical, or mathematical relationships which describe a system in terms of state, entities and their attributes, sets, processes, events, activities, and delays.
- **System state:** A collection of variables that contain all the information necessary to describe the system at any time.
- **Entity:** Any object or component in the system which requires explicit representation in the model (e.g., a server, a customer, a machine).
- **Attributes:** The properties of a given entity (e.g., the priority of a customer, the routing of a job through a job shop).

- **List:** A collection of (permanently or temporarily) associated entities ordered in some logical fashion (such as all customers currently in a waiting line, ordered by first come, first served, or by priority).

- **Event:** An instantaneous occurrence that changes the state of a system as an arrival of a new customer.

- **Event notice:** A record of an event to occur at the current or some future time, along with any associated data necessary to execute the event; at a minimum, the record includes the event type and the event time.

- **Event list:** A list of event notices for future events, ordered by time of occurrence; also known as the future event list (FEL).

- **Activity:** A duration of time of specified length (e.g., a service time or arrival time), which is known when it begins (although it may be defined in terms of a statistical distribution).

- **Delay:** A duration of time of unspecified indefinite length, which is not known until it ends (e.g., a customer's delay in a last-in, first-out waiting line which, when it begins, depends on future arrivals).

- **Clock:** A variable representing simulated time. The future event list is ranked by the event time recorded in the event notice.

- An activity typically represents a service time, an inter arrival time, or any other processing time whose duration has been characterized and defined by the modeler.

An activity's duration may be specified in a number of ways:

1. **Deterministic**-for example, always exactly 5 minutes.

2. **Statistical**-for example, as a random draw from among 2,5,7 with equal probabilities.

3. A function depending on system variables and/or entity attributes.

- The duration of an activity is computable from its specification at the instant it begins.

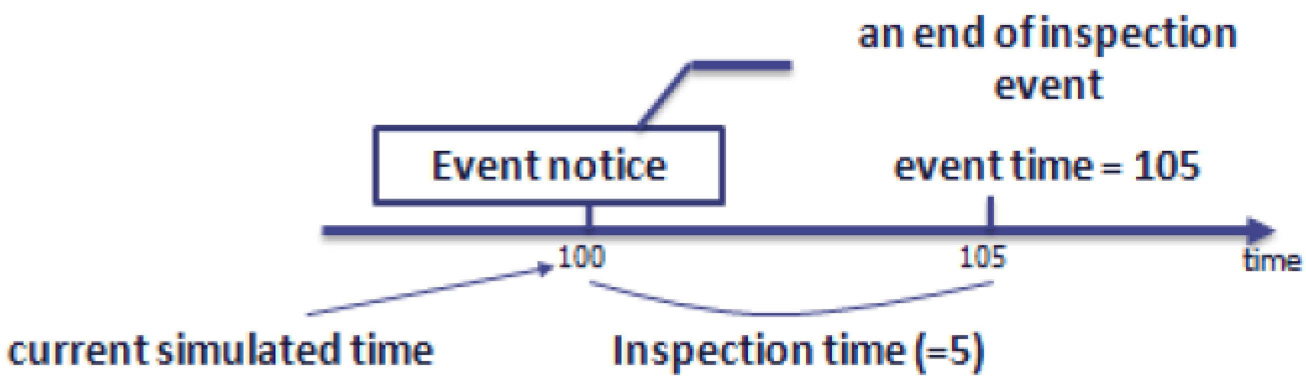
- A delay's duration is not specified by the modeler ahead of time, but rather is determined by system conditions.

- A **delay** is sometimes called a ***conditional wait***, while an **activity** is called unconditional wait.

- The completion of an activity is an event, often called ***primary event.***

- The completion of a delay is sometimes called a ***conditional or secondary event.***

- To keep track of activities and their expected completion time, at the simulated instant that activity duration begins,
- An ***event notice*** is created having an event time equal to the activity's completion time.



EXAMPLE :- The Able Baker Carhop Problem

This example illustrates the simulation procedure when there is more than one service channel. Consider a drive-in restaurant where carhops take orders and bring food to the car. Cars arrive in the manner given in table. There are two carhops-Able and Baker. Able is better able to do the job and works a bit faster than Baker. The distribution of their service times are given.

A discrete- event model has the following components:

- **System State LQ(t)**, the number of cars waiting to be served at time t
- **LA(t)**, 0 or 1 to indicate Able being idle or busy at time t
- **LB (t)**, 0 or 1 to indicate Baker being idle or busy at time t

- **Entities** Neither the customers (i.e., cars) nor the servers need to be explicitly represented, except in terms of the state variables, unless certain customer averages are desired

- **Events** Arrival event, Service completion by Able, Service completion by Baker.

- **Activities** Inter arrival time, Service time by Able, Service time by Baker.

- **Delay** A customer's wait in queue until Able or Baker becomes free.

- The definition of the model components provides a static description of the model.

- A description of the dynamic relationships and interactions between the components is also needed.

- A discrete-event simulation: the modeling over time of a system all of whose state changes occur at discrete points in time-those points when an event occurs.

- A discrete-event simulation proceeds by producing a sequence of system snapshots (or system images) which represent the evolution of the system through time.

Clock	*System State*	*Entities and Attributes*	*Set 1*	*Set 2*	...	*Future Event List, FEL*	*Cumulative Statistics and Counters*
t	$(x, y, z, \ldots)$					$(3, t_1)$– Type 3 event to occur at time t_1	
						$(1, t_2)$– Type 1 event to occur at time t_2	
	.					. .	
	.					. .	
	.					. .	

Prototype system snapshot at simulation time t

The Event-Scheduling/Time-Advance Algorithm

- The mechanism for advancing simulation time and guaranteeing that all events occur in correct chronological order is based on the future event list (FEL).
- Future Event List (FEL) : contain all event notices for events that have been scheduled to occur at a future time.
- Events are arranged chronologically; that is, the event times satisfy.

t < t1 <= t2 <= t3 <= ….,<= tn

- t is the value of CLOCK, the current value of simulated time. The event dated with time t1 is called the **imminent event**; that is, it is the next event will occur.

- Scheduling a future event means that at the instant an activity begins, its duration is computed or drawn as a sample from a statistical distribution and the end-activity event, together with its event time, is placed on the future event list.

- The sequence of actions which a simulator must perform to advance the clock system snapshot is called the event ***scheduling/time-advance algorithm.***

Old system snapshot at time t

CLOCK	*System State*	...	*Future Event List*	...
t	(5, 1, 6)		$(3, t_1)$ – Type 3 event to occur at time t_1 $(1, t_2)$ – Type 1 event to occur at time t_2 $(1, t_3)$ – Type 1 event to occur at time t_3 ⋮ $(2, t_n)$ – Type 2 event to occur at time t_n	

Event-scheduling/time-advance algorithm

Step 1. Remove the event notice for the imminent event (event 3, time t_1) from FEL

Step 2. Advance CLOCK to imminent event time (i.e., advance CLOCK from t to t_1).

Step 3. Execute imminent event: update system state, change entity attributes, and set membership as needed.

Step 4. Generate future events (if necessary) and place their event notices on FEL ranked by event time. (*Example:* Event 4 to occur at time t^*, where $t_2 < t^* < t_3$.)

Step 5. Update cumulative statistics and counters.

New system snapshot at time t_1

CLOCK	*System State*	...	*Future Event List*	...
t_1	(5, 1, 5)		$(1, t_2)$ – Type 1 event to occur at time t_2	
			$(4, t^*)$ – Type 4 event to occur at time t^*	
			$(1, t_3)$ – Type 1 event to occur at time t_3	
			. . .	
			. . .	
			. . .	
			$(2, t_n)$ – Type 2 event to occur at time t_n	

List processing: the management of a list.

- **The removal of the imminent event:** As the imminent event is usually at the top of the list, its removal is as efficient as possible.
- **The addition of a new event to the list, and occasionally removal of some event (called cancellation of an event):** Addition of a new event (and cancellation of an old event) requires a search of the list.
- The **efficiency** of this search depends on the logical organization of the list and on how the search is conducted.

- When event 4 (say, an arrival event) with event time t* is generated at step 4, one possible way to determine its correct position on the FEL is to conduct a top-down search:
- If $t^* < t2$, place event 4 at the top of the FEL.
- If $t2 < t^* < t3$, place event 4 second on the list.
- If t3, $< t^* < t4$, place event 4 third on the list.
- If $tn < t^*$, event 4 last on the list.
- Another way is to conduct a bottom-up search.
- The system snapshot at time 0 is defined by the initial conditions and the generation of the so-called exogenous events.

- The method of generating an external arrival stream, called bootstrapping.

- Every simulation must have a stopping event, here called E, which defines how long the simulation will run. There are generally two ways to stop a simulation:

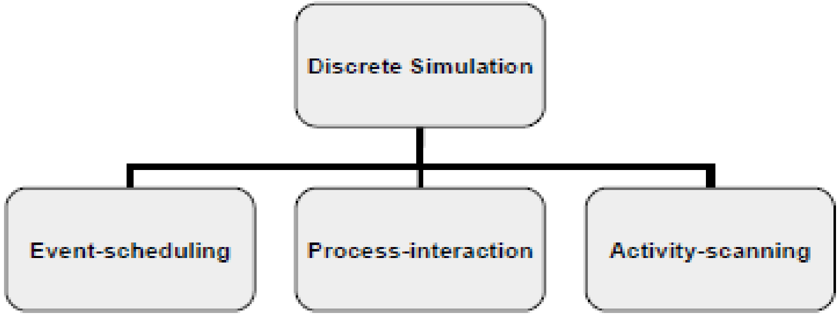
1. At time 0, schedule a stop simulation event at a specified future time TE. Thus, before simulating, it is known that the simulation will run over the time interval [0, TE].

   Example: Simulate a job shop for TE = 40 hours.

2. Run length TE is determined by the simulation itself. Generally, TE is the time of occurrence of some specified event E.

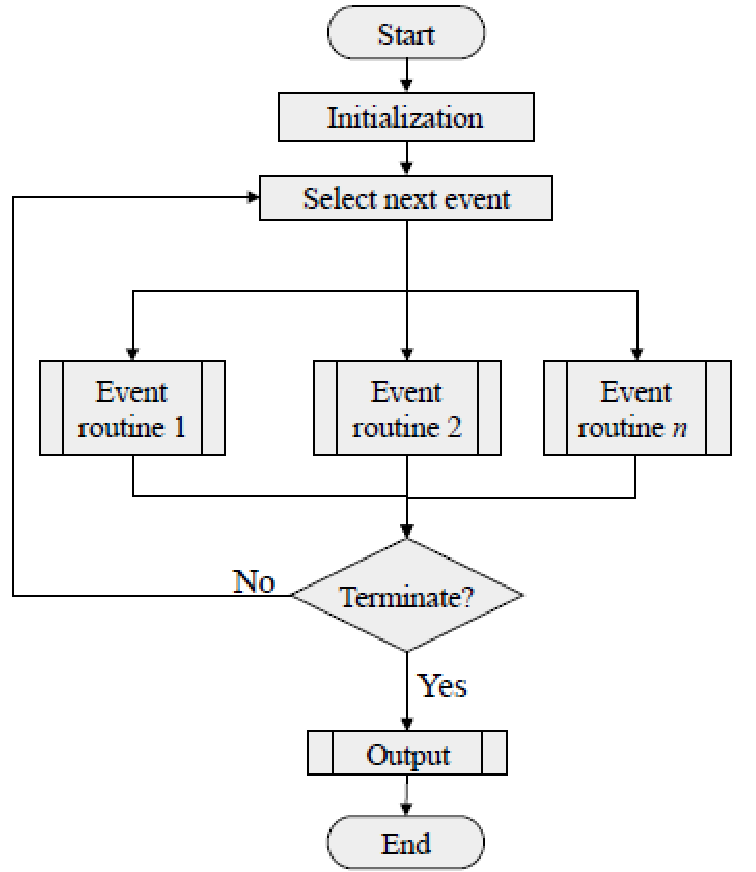
Examples: TE is the time of the 100th service completion at a certain service center. TE is the time of breakdown of a complex system.

- World view
  - A world view is an orientation for the model developer
  - Simulation packages typically support some world views
  - Here, only world views for discrete simulations



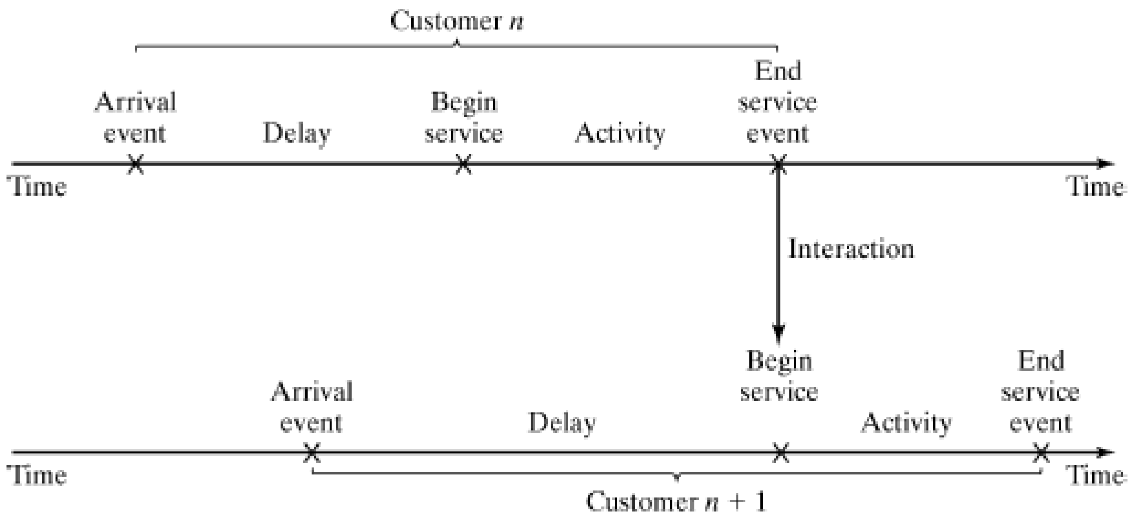
World Views

- Event-scheduling
  - Focus on events
  - Identify the entities and their attributes
  - Identify the attributes of the system
  - Define what causes a change in system state
  - Write a routine to execute for each event
  - Variable time advance



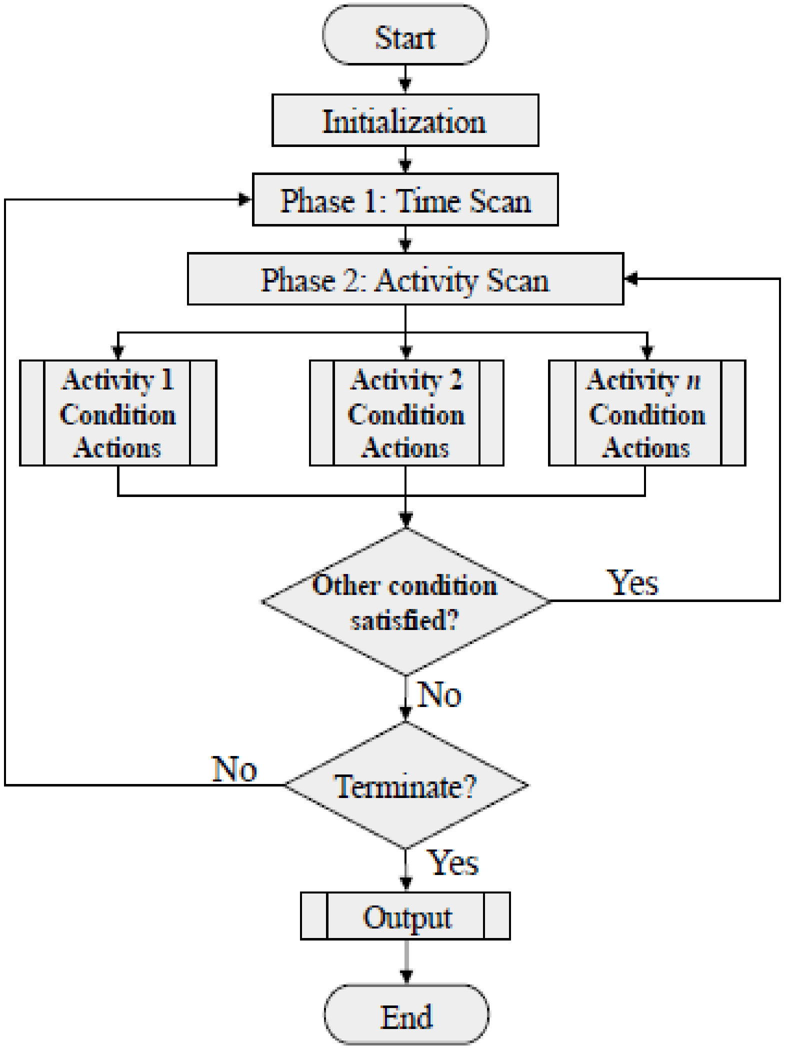
World Views

- Process-interaction
  - Modeler thinks in terms of processes
  - A process is the lifecycle of one entity, which consists of various events and activities
  - Simulation model is defined in terms of entities or objects and their life cycle as they flow through the system, demanding resources and queueing to wait for resources
  - Some activities might require the use of one or more resources whose capacities are limited
  - Processes interact, e.g., one process has to wait in a queue because the resource it needs is busy with another process
  - A process is a time-sequenced list of events, activities and delays, including demands for resources, that define the life cycle of one entity as it moves through a system
  - Variable time advance



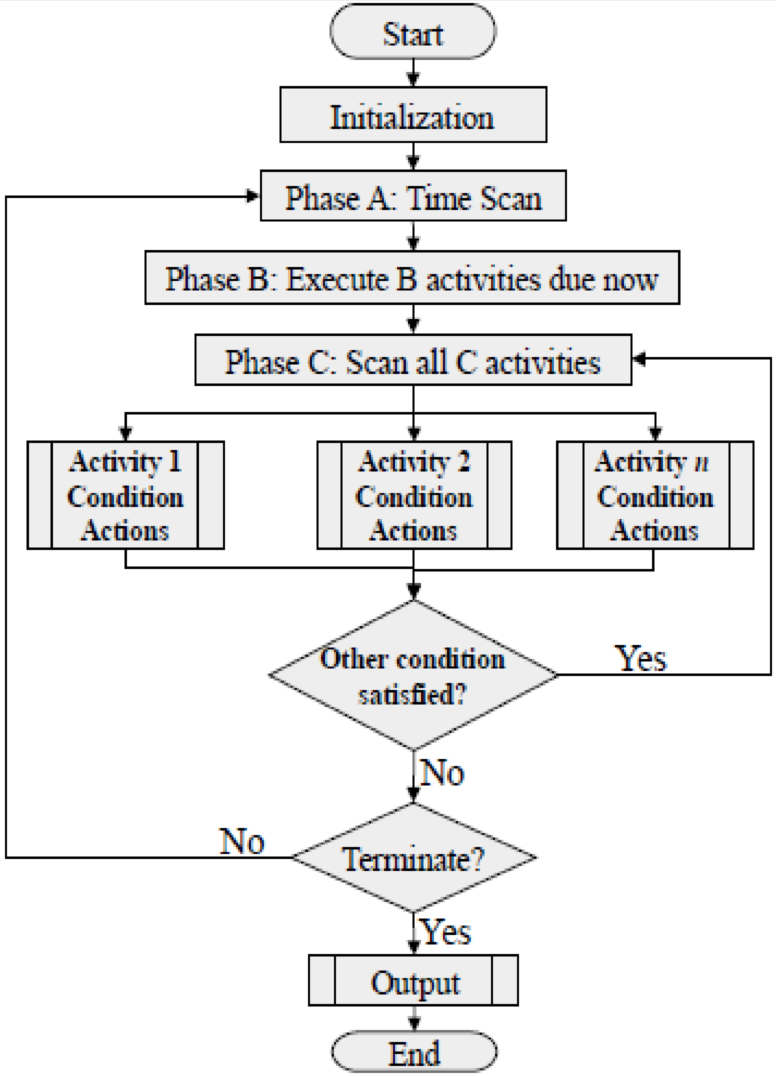
World Views

- Activity-scanning
  - Modeler concentrates on activities of a model and those conditions that allow an activity to begin
  - At each clock advance, the conditions for each activity are checked, and, if the conditions are true, then the corresponding activity begins
  - Fix time advance
  - Disadvantage: The repeated scanning to discover whether an activity can begin results in slow runtime
  - Improvement: Three-phase approach
    - Combination of event scheduling with activity scanning

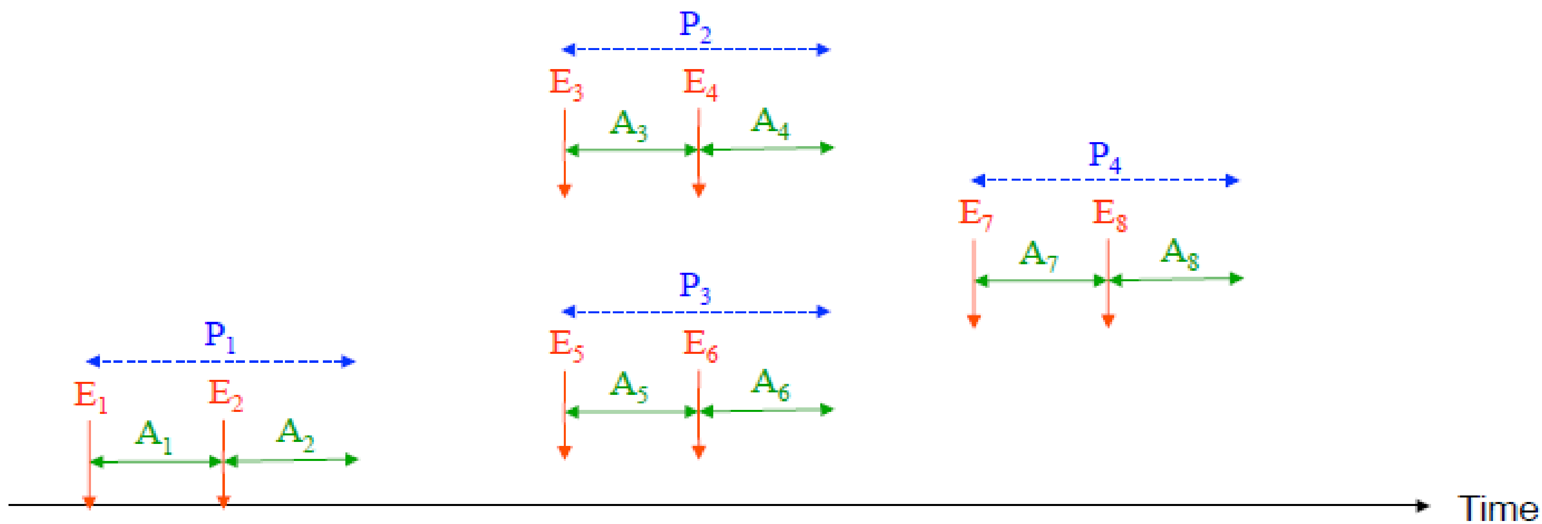


World Views

- Three-phase approach
  - Events are activities of duration zero time units
  - Two types of activities
    - **B activities:** activities bound to occur; all primary events and unconditional activities
    - **C activities:** activities or events that are conditional upon certain conditions being true
  - The B-type activites can be scheduled ahead of time, just as in the event-scheduling approach
    - Variable time advance
    - FEL contains only B-type events
  - Scanning to learn whether any C-type activities can begin or C-type events occur happen only at the end of each time advance, after all B-type events have completed



World Views



Manual Simulation Using Event Scheduling

- In an event-scheduling simulation, a simulation table is used to record the successive system snapshots as time advances.
- Let us consider the example of a grocery shop which has only one checkout counter. **(Single-Channel Queue)**
- The system consists of those customers in the waiting line plus the one (if any) checking out.

The model has the following components:

1. System state (LQ (t), LS (t)),

- where LQ (t) is the number of customers in the waiting line, and LS (t) is the number being served (0 or 1) at time t.

2. Entities: The server and customers are not explicitly modeled, except in terms of the state variables above.

3. Events

- Arrival (A)
- Departure (D)
- Stopping event (E), scheduled to occur at time 60.

4. Event notices

- (A, t). Representing an arrival event to occur at future time t
- (D, t), representing a customer departure at future time t
- (E, 60), representing the simulation-stop event at future time 60

5. Activities

- Inter arrival time
- Service time

6. Delay

- Customer time spent in waiting line.
- In this model, the FEL will always contain either two or three event notices.

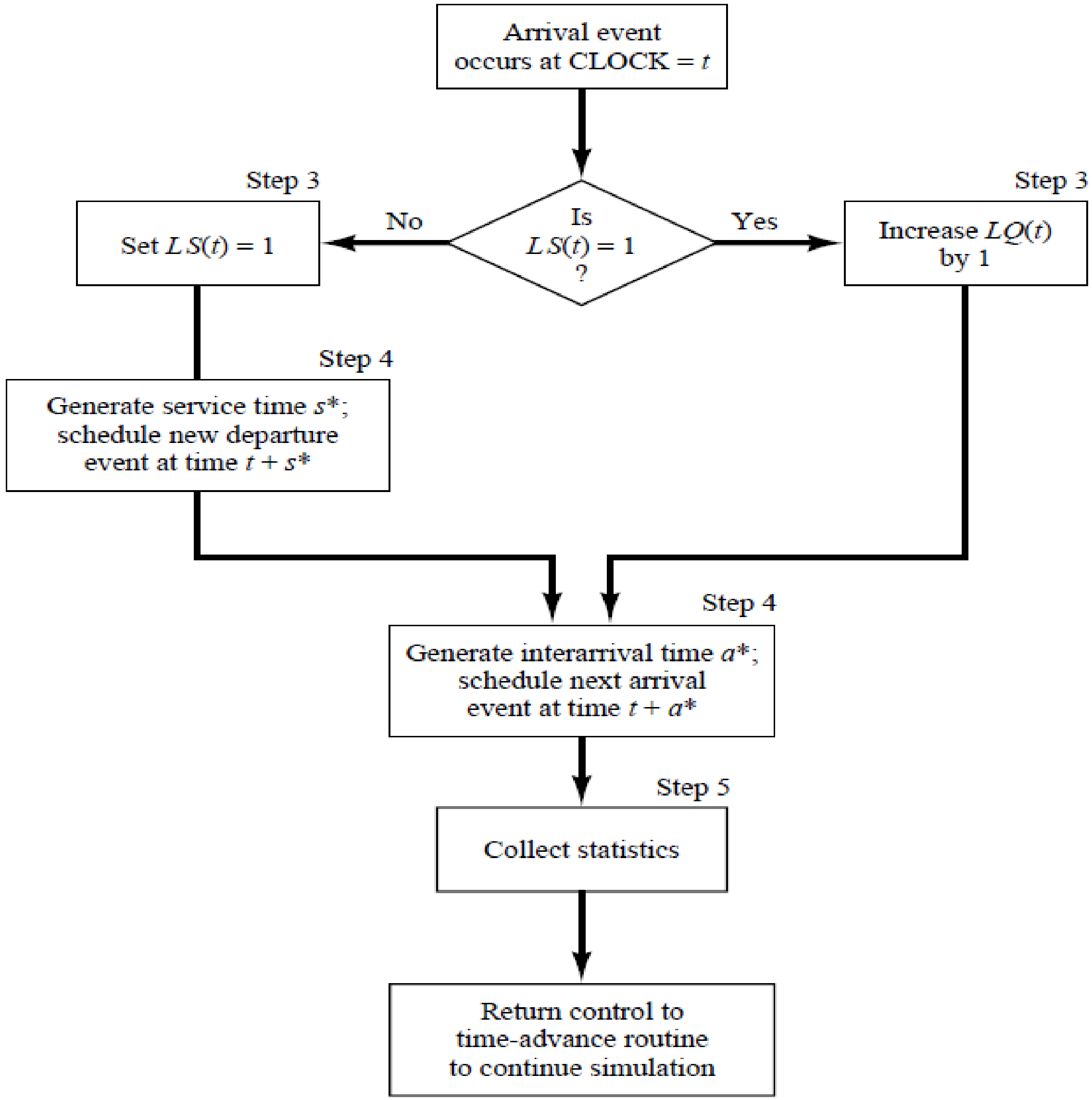


Figure 3.5 Execution of the arrival event.

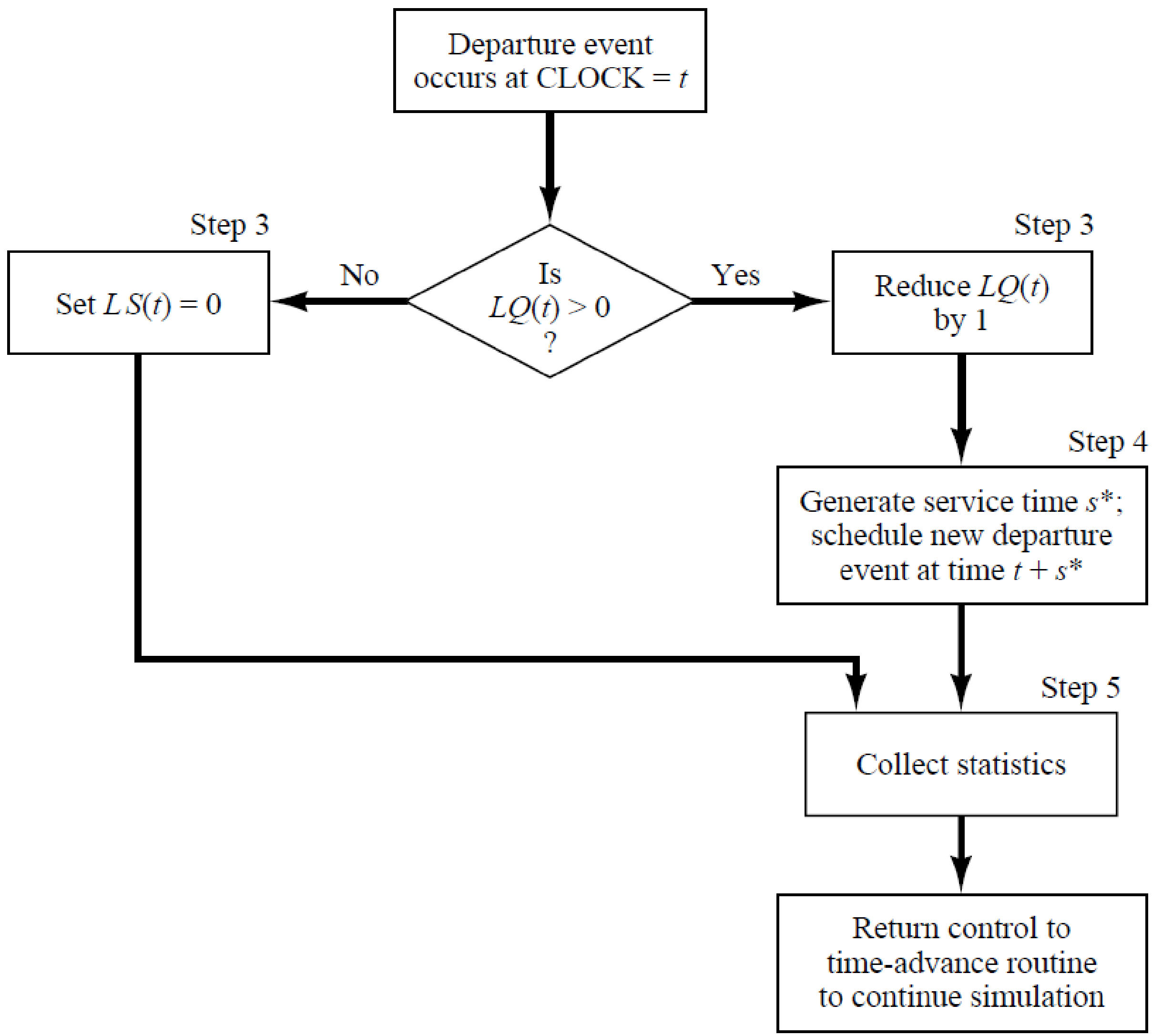


Figure 3.6 Execution of the departure event.

- The inter arrival times and service times will be

Interarrival Times	8	6	1	8	3	8	...
Service Times	4	1	4	3	2	4	...

Initial conditions

- the system snapshot at time zero (CLOCK = 0)
- LQ(0) = 0, LS(0) = 1
- both a departure event and arrival event on the FEL.
- The simulation is scheduled to stop at time 60.
- **Server utilization: total server busy time (B) / total time (TE).**
- **Maximum queue length MQ**
- a* : the generated inter arrival time
- s* : the generated service times

Clock	System State: $LQ(t)$	$LS(t)$	Future Event List	Comment	Cumulative Statistics: B	MQ
0	0	1	$(D,4)$ $(A,8)$ $(E,60)$	First A occurs $(a^* = 8)$ Schedule next A $(s^* = 4)$ Schedule first D	0	0
4	0	0	$(A,8)$ $(E,60)$	First D occurs: $(D,4)$	4	0
8	0	1	$(D,9)$ $(A,14)$ $(E,60)$	Second A occurs: $(A,8)$ $(a^* = 6)$ Schedule next A $(s^* = 1)$ Schedule next D	4	0
9	0	0	$(A,14)$ $(E,60)$	Second D occurs: $(D,9)$	5	0
14	0	1	$(A,15)$ $(D,18)$ $(E,60)$	Third A occurs: $(A,14)$ $(s^* = 4)$ Schedule next D	5	0
15	1	1	$(D,18)$ $(A,23)$ $(E,60)$	Fourth A occurs: $(A,15)$ (Customer delayed)	6	1
18	0	1	$(D,21)$ $(A,23)$ $(E,60)$	Third D occurs: $(D,18)$ $(s^* = 3)$ Schedule next D	9	1
21	0	0	$(A,23)$ $(E,60)$	Fourth D occurs: $(D,21)$	12	1

The simulation Table for check out counter - covers the time interval [0, 21]

- As soon as the system snapshot at time CLOCK = 0 is complete, the simulation begins.
- At time 0, the imminent event is (D, 4).
- The CLOCK is advanced to time 4, and (D, 4) is removed from the FEL.
- Since *LS(t*) = 1 for 0 <= t <= 4 (i.e., the server was busy for 4 minutes), the cumulative busy time is Increased from B = 0 to B = 4.
- By the event logic in Figure 1(B), set LS (4) = 0 (the server becomes idle).
- The FEL is left with only two future events, (A, 8) and (E, 0).
- The simulation CLOCK is next advanced to time 8 and an arrival event is executed.

Suppose the system analyst desires to estimate

- **Mean response time:** the average length of time a customer spends in the system
- **Mean proportion** of customers who spend 4 or more minutes in the system
- **Entities (Ci, t):** representing customer Ci who arrived at time t
- **Event notices:**
- **(A, t, Ci)**, the arrival of customer Ci at future time t
- **(D, t, Cj)**, the departure of customer Cj at future time t
- Set : **“CHECKOUTLINE”** the set of all customers currently at the checkout counter (being served or waiting to be served), ordered by time of arrival
- A customer entity with arrival time as an attribute is added in order to estimate mean response time.

Three new cumulative statistics will be collected :

- **S** : the sum of customer response times for all customers who have departed by the current time
- **F** : the total number of customers who spend 4 or more minutes at the checkout counter
- **ND**: the total number of departures up to the current simulation time.
- These three cumulative statistics will be updated whenever the departure event occurs.

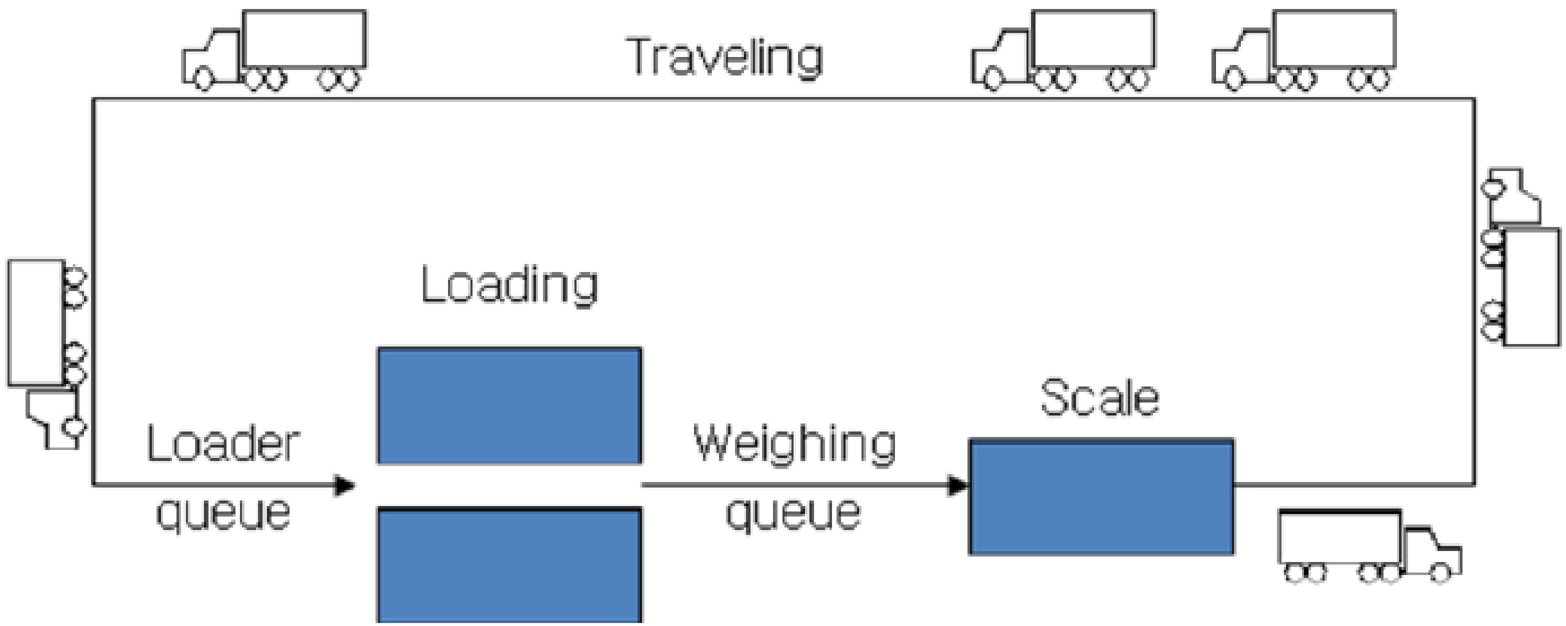
The response time for customer is computed by

- Response time = CLOCK TIME - attribute “time of arrival”
- For a simulation run length of 21 minutes
- The average response time was S/ND = 15/4 = 3.75 minutes
- The observed proportion of customers who spent 4 or more minutes in the system was F/ND = 0.75.

Clock	System State $LQ(t)$	$LS(t)$	List "CHECKOUT LINE"	Future Event List	Cumulative Statistics S	N_D	F
0	0	1	$(C1, 0)$	$(D, 4, C1)$ $(A, 8, C2)$ $(E, 60)$	0	0	0
4	0	0		$(A, 8, C2)$ $(E, 60)$	4	1	1
8	0	1	$(C2, 8)$	$(D, 9, C2)$ $(A, 14, C3)$ $(E, 60)$	4	1	1
9	0	0		$(A, 14, C3)$ $(E, 60)$	5	2	1
14	0	1	$(C3, 14)$	$(A, 15, C4)$ $(D, 18, C3)$ $(E, 60)$	5	2	1
15	1	1	$(C3, 14)$ $(C4, 15)$	$(D, 18, C3)$ $(A, 23, C5)$ $(E, 60)$	5	2	1
18	0	1	$(C4, 15)$	$(D, 21, C4)$ $(A, 23, C5)$ $(E, 60)$	9	3	2
21	0	0		$(A, 23, C5)$ $(E, 60)$	15	4	3

Example - (The Dump Truck Problem)

Six dump trucks are used to haul coal from the entrance of a small mine to the railroad. Each truck is loaded by one of two loaders. After loading, a truck immediately moves to scale, to be weighted as soon as possible. Both the loaders and the scale have a first come, first serve waiting line(or queue) for trucks. The time taken to travel from loader to scale is considered negligible. After being weighted, a truck begins a travel time and then afterward returns to the loader queue.


Traveling
Loading
Loader
queue
Weighing
queue
Scale

Distribution of Loading for the Dump Truck

Loading time	Probability	Cumulative probability	Random-Digit Assignment
5	0.30	0.30	1-3
10	0.50	0.80	4-8
15	0.20	1.00	9-0

Distribution of Weighing Time for the Dump Truck

Weighing time	Probability	Cumulative probability	Random-Digit Assignment
12	0.70	0.70	1-7
16	0.30	1.00	8-0

Distribution of Travel Time for the Dump Truck

Travel time	Probability	Cumulative probability	Random-Digit Assignment
40	0.40	0.40	1-4
60	0.30	0.70	5-7
80	0.20	0.90	8-9
100	0.10	1.00	0

The activity times are taken from the following list as needed:

Loading Time	10	5	5	10	15	10	10
Weighing Time	12	12	12	16	12	16	
Travel Time	60	100	40	40	80		

System state [LQ(t), L(t), WQ(t), W(t)]

- LQ(t) = number of trucks in loader queue
- L(t) = number of trucks (0, 1, or 2) being loaded
- WQ(t) = number of trucks in weigh queue
- W(t) = number of trucks (0 or 1) being weighed, all at simulation time t

Event notices :

- (ALQ, t, DTi), dump truck i arrives at loader queue (ALQ) at time t
- (EL, t, DTi), dump truck i ends loading (EL) at time t
- (EW, t, DTi), dump truck i ends weighing (EW) at time t

- **Entities :** The six dump trucks (DT 1, … , DT 6)
- **Lists :**
  1. Loader queue : all trucks waiting to begin loading, ordered on a first come, first served basis
  2. Weigh queue : all trucks waiting to be weighed, ordered on a first come, first served basis

- **Activities :** Loading time, weighing time, and travel time
- **Delays :** Delay at loader queue, and delay at scale
- It has been assumed that five of the trucks are at the loaders and one is at the scale at time 0.

	System State				Lists			Cumulative Statistics	
Clock t	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue	Future Event List	BL	BS
0	3	2	0	1	DT4 DT5 DT6		(EL, 5, DT3) (EL, 10, DT2) (EW, 12, DT1)	0	0
5	2	2	1	1	DT5 DT6	DT3	(EL, 10, DT2) (EL, 5 + 5, DT4) (EW, 12, DT1)	10	5

	System State				Lists			Cumulative Statistics	
Clock t	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue	Future Event List	BL	BS
10	1	2	2	1	DT6	DT3 DT2	(EL, 10, DT4) (EW, 12, DT1) (EL, 10 + 10, DT5)	20	10
10	0	2	3	1		DT3 DT2 DT4	(EW, 12, DT1) (EL, 20, DT5) (EL, 10 + 15, DT6)	20	10
12	0	2	2	1		DT2 DT4	(EL, 20, DT5) (EW,12+12,DT3) (EL, 25, DT6) (ALQ,12+60,DT1)	24	12

	System State				Lists			Cumulative Statistics	
Clock t	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue	Future Event List	BL	BS
20	0	1	3	1		DT2 DT4 DT5	(EW, 24, DT3) (EL, 25, DT6) (ALQ, 72, DT1)	40	20
24	0	1	2	1		DT4 DT5	(EL, 25, DT6) (EW, 24+12, DT2) (ALQ, 72, DT1) (ALQ, 24+100, DT3)	44	24
25	0	0	3	1		DT4 DT5 DT6	(EW, 36, DT2) (ALQ, 72, DT1) (ALQ, 124, DT3)	45	25
36	0	0	2	1		DT5 DT6	(EW, 36+16. DT4) (ALQ, 72, DT1) (ALQ, 36+40, DT2) (ALQ, 124, DT3	45	52

	System State				Lists			Cumulative Statistics	
Clock t	LQ(t)	L(t)	WQ(t)	W(t)	Loader Queue	Weigh Queue	Future Event List	BL	BS
52	0	0	1	1		DT6	(EW, 52+12, DT5) (ALQ, 72, DT1) (ALQ, 76, DT2) (ALQ, 52+40, DT4) (ALQ, 124, DT3)	45	52
64	0	0	0	1			(ALQ, 72, DT1) (ALQ, 76, DT2) (EW, 64+16, DT6) (ALQ, 92, DT4) (ALQ, 124, DT3) (ALQ, 64+80, DT5)	45	64
72	0	1	0	1			(ALQ, 76, DT2) (EW, 80, DT6) (EL, 72+10, DT1) (ALQ, 92, DT4) (ALQ, 124, DT3) (ALQ, 144, DT5	45	72
76	0	2	0	1			(EW, 80, DT6) (EL, 82, DT1) (EL, 76+10, DT2) (ALQ, 92, DT4) (ALQ, 124, DT3) (ALQ, 144, DT5	49	76

This logic for the occurrence of the end-loading event

- **When an end-loading (EL) event** occurs, say for truck j at time t , other events may be triggered.
- If the scale is idle [W (t) =0], truck j begins weighing and an end-weighing event (EW) is scheduled on the FEL.
- Otherwise, truck **j** joins the weigh queue.
- If at this time there is another truck waiting for a loader, it will be removed from the loader queue and will begin loading by the scheduling of an end-loading event (EL) on the FEL.

In order to estimate the loader and scale utilizations, two cumulative statistics are maintained:

- BL = total busy time of both loaders from time 0 to time t
- BS = total busy time of the scale from time 0 to time t

- The utilizations are estimated as follows:
- **Average loader utilization** $= \frac{49/2}{76} = 0.32$
- **Average scale utilization** $= \frac{76}{76} = 1.00$

Using the activity scanning approach

Activity	Condition
Loading time	Truck is at front of loader queue, and at least one loader is idle.
Weighing time	Truck is at front of weigh queue and weigh scale is idle.
Travel time	Truck has just completed weighing.

Using the process-interaction approach