LAB

7 BASIC MODELING CONCEPTS

Imagination is more important than knowledge.

—Albert Einstein

In this chapter we continue to describe other basic features and modeling concepts of ProModel. In Section L7.1 we show an application with multiple locations and multiple entity types. Section L7.2 describes modeling of multiple parallel locations. Section L7.3 shows various routing rules. In Section L7.4 we introduce the concept of variables. Section L7.5 introduces the inspection process, tracking of defects, and rework. In Section L7.6 we show how to assemble nonidentical entities and produce assemblies. In Section L7.7 we show the process of making temporary entities through the process of loading and subsequent unloading. Section L7.8 describes how entities can be accumulated before processing. Section L7.9 shows the splitting of one entity into multiple entities. In Section L7.10 we introduce various decision statements with appropriate examples. Finally, in Section L7.11 we show you how to model a system that shuts down periodically.

L7.1 Multiple Locations, Multiple Entity Types

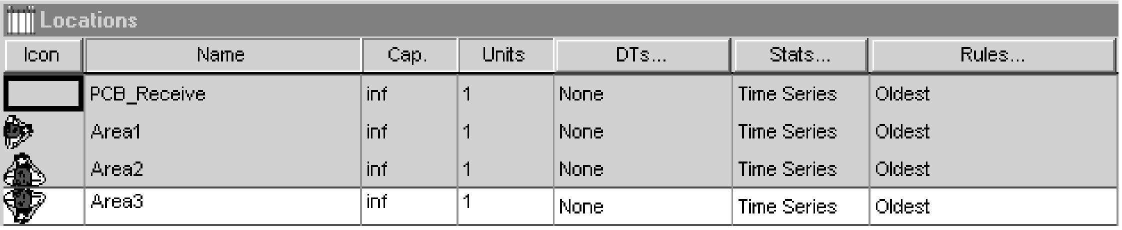
Problem Statement

In one department at **Pomona Electronics,** three different printed circuit boards are assembled. Each board is routed through three assembly areas. The routing order is different for each of the boards. Further, the time to assemble a board depends on the board type and the operation. The simulation model is intended to determine the time to complete 500 boards of each type. The assembly time for each board is exponentially distributed with the mean times shown in Table L7.1.

Define three locations (Area1, Area2, and Area3) where assembly work is done and another location, PCB_Receive, where all the printed circuit boards

FIGURE L7.1

The three processing locations and the receiving dock.



Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	PCB_Receive	inf	1	None	Time Series	Oldest
	Area1	inf	1	None	Time Series	Oldest
	Area2	inf	1	None	Time Series	Oldest
	Area3	inf	1	None	Time Series	Oldest

FIGURE L7.2

Layout of Pomona Electronics.

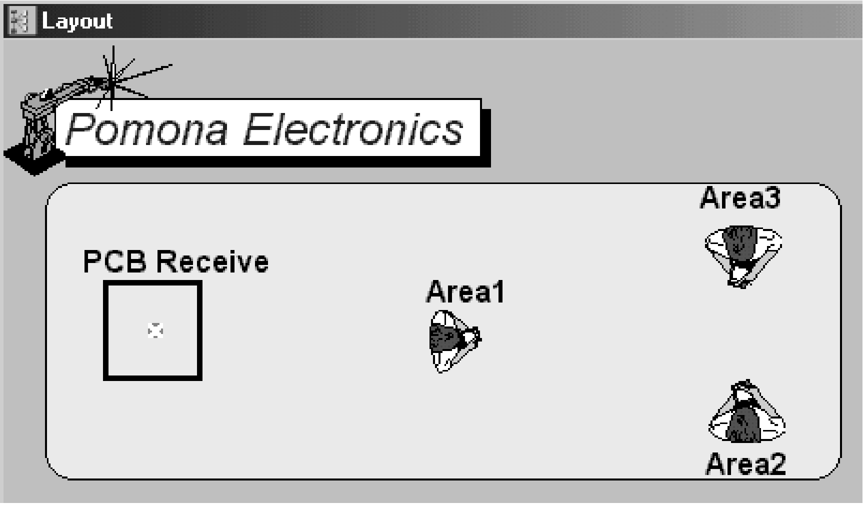


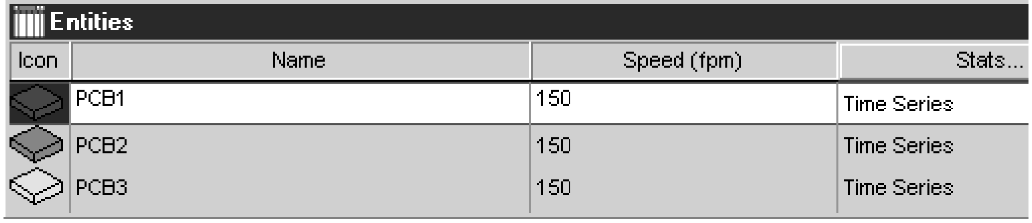
TABLE L7.1 Assembly Times for the Printed Circuit Boards

Printed Circuit Board 1		Printed Circuit Board 2		Printed Circuit Board 3	
Area	*Mean Time*	*Area*	*Mean Time*	*Area*	*Mean Time*
1	10	2	5	3	12
2	12	1	6	2	14
3	15	3	8	1	15

arrive (Figure L7.1). Assume each of the assembly areas has infinite capacity. The layout of Pomona Electronics is shown in Figure L7.2. Note that we used Background Graphics → Behind Grid, from the Build menu, to add the Pomona Electronics logo on the simulation model layout. Add the robot graphics (or something appropriate). Define three entities as PCB1, PCB2, and PCB3 (Figure L7.3).

FIGURE L7.3

The three types of circuit boards.

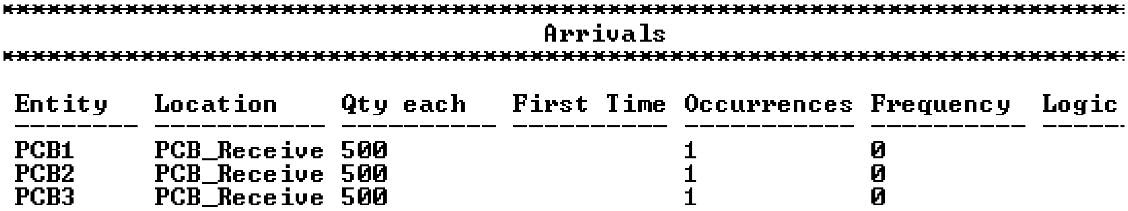


Entities

Icon	Name	Speed (fpm)	Stats...
	PCB1	150	Time Series
	PCB2	150	Time Series
	PCB3	150	Time Series

FIGURE L7.4

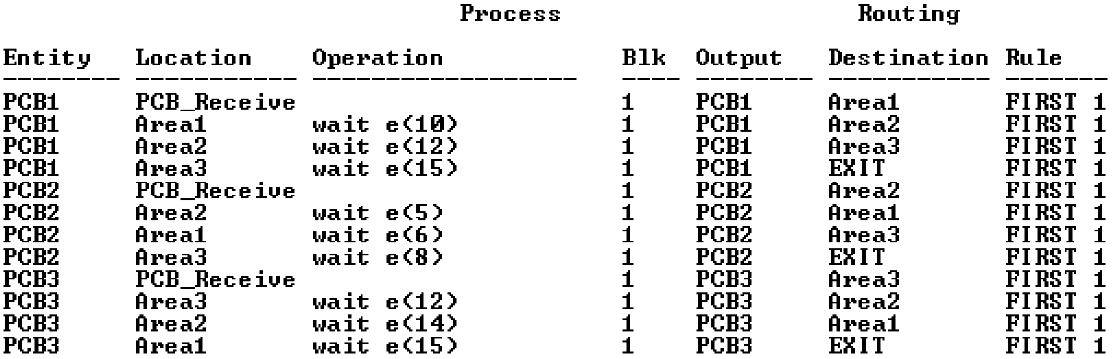
The arrival process for all circuit boards.



```
**************************************************************************
                                Arrivals
**************************************************************************

Entity   Location     Qty each   First Time Occurrences Frequency  Logic
-------- ------------ ---------- ---------- ----------- ---------- -----
PCB1     PCB_Receive 500                   1           0
PCB2     PCB_Receive 500                   1           0
PCB3     PCB_Receive 500                   1           0
```

FIGURE L7.5

Processes and routings for Pomona Electronics.



```
                             Process                    Routing

Entity   Location     Operation            Blk  Output   Destination Rule
-------- ------------ -------------------- ---- -------- ----------- -------
PCB1     PCB_Receive                       1    PCB1     Area1       FIRST 1
PCB1     Area1        wait e(10)           1    PCB1     Area2       FIRST 1
PCB1     Area2        wait e(12)           1    PCB1     Area3       FIRST 1
PCB1     Area3        wait e(15)           1    PCB1     EXIT        FIRST 1
PCB2     PCB_Receive                       1    PCB2     Area2       FIRST 1
PCB2     Area2        wait e(5)            1    PCB2     Area1       FIRST 1
PCB2     Area1        wait e(6)            1    PCB2     Area3       FIRST 1
PCB2     Area3        wait e(8)            1    PCB2     EXIT        FIRST 1
PCB3     PCB_Receive                       1    PCB3     Area3       FIRST 1
PCB3     Area3        wait e(12)           1    PCB3     Area2       FIRST 1
PCB3     Area2        wait e(14)           1    PCB3     Area1       FIRST 1
PCB3     Area1        wait e(15)           1    PCB3     EXIT        FIRST 1
```

Define the arrival process (Figure L7.4). Assume all 1500 boards are in stock when the assembly operations begin. The process and routing tables are developed as in Figure L7.5.

Run the simulation model. Note that the whole batch of 1500 printed circuit boards (500 of each) takes a total of 2 hours and 27 minutes to be processed.

L7.2 Multiple Parallel Identical Locations

Location: Capacity versus Units

Each location in a model has associated with it some finite capacity. The *capacity* of a location refers to the maximum number of entities that the location can hold at any time. The reserved word `INF` or `INFINITE` sets the capacity to the maximum allowable value. The default capacity for a location is one unless a counter, conveyor, or queue is the first graphic assigned to it. The default capacity in that case is `INFINITE`.

The *number of units* of a location refers to the number of identical, interchangeable, and parallel locations referenced as a single location for routing and processing. A multiunit location eliminates the need to create multiple locations and multiple processes for locations that do the same thing. While routings to and from each unit are the same, each unit may have unique downtimes.

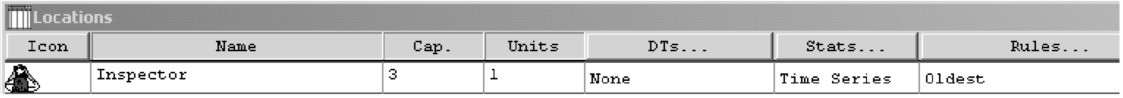
A multiunit location is created by entering a number greater than one as the number of units for a normal location. A corresponding number of locations will appear below the original location with a numeric extension appended to each copied location name designating the unit number of that location. The original location record becomes the prototype for the unit records. Each unit will inherit the prototype's characteristics unless the individual unit's characteristics are changed. Downtimes, graphic symbols, and, in the case of multicapacity locations, capacities may be assigned to each unit. As the number of units of a location is changed, the individual unit locations are automatically created or destroyed accordingly.

The following three situations describe the advantages and disadvantages of modeling multicapacity versus multiunit locations.

- *Multicapacity locations*. Locations are modeled as a single unit with multicapacity (Figure L7.6). All elements of the location perform identical operations. When one element of the location is unavailable due to downtime, all elements are unavailable. Only clock-based downtimes are allowed.
- *Multiunit locations*. Locations are modeled as single capacity but multiple units (Figure L7.7). These are locations consisting of two or more parallel and interchangeable processing units. Each unit shares the same sources of input and the same destinations for output, but each may have independent operating characteristics. This method provides more flexibility in the assignment of downtimes and in selecting an individual unit to process a particular entity.
- *Multiple, single-capacity locations*. Locations are modeled as individual and single capacity (Figure L7.8). Usually noninterchangeable locations are modeled as such. By modeling as individual locations, we gain the flexibility of modeling separate downtimes for each element. In addition, we now have complete flexibility to determine which element will process a particular entity.

FIGURE L7.6

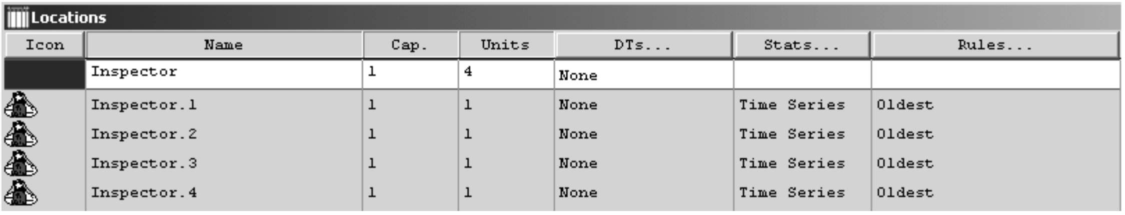
Single unit of multicapacity location.



Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Inspector	3	1	None	Time Series	Oldest

FIGURE L7.7

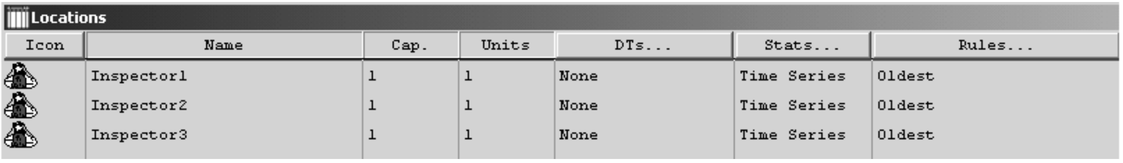
Multiple units of single-capacity locations.



Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Inspector	1	4	None		
	Inspector.1	1	1	None	Time Series	Oldest
	Inspector.2	1	1	None	Time Series	Oldest
	Inspector.3	1	1	None	Time Series	Oldest
	Inspector.4	1	1	None	Time Series	Oldest

FIGURE L7.8

Multiple single-capacity locations.



Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Inspector1	1	1	None	Time Series	Oldest
	Inspector2	1	1	None	Time Series	Oldest
	Inspector3	1	1	None	Time Series	Oldest

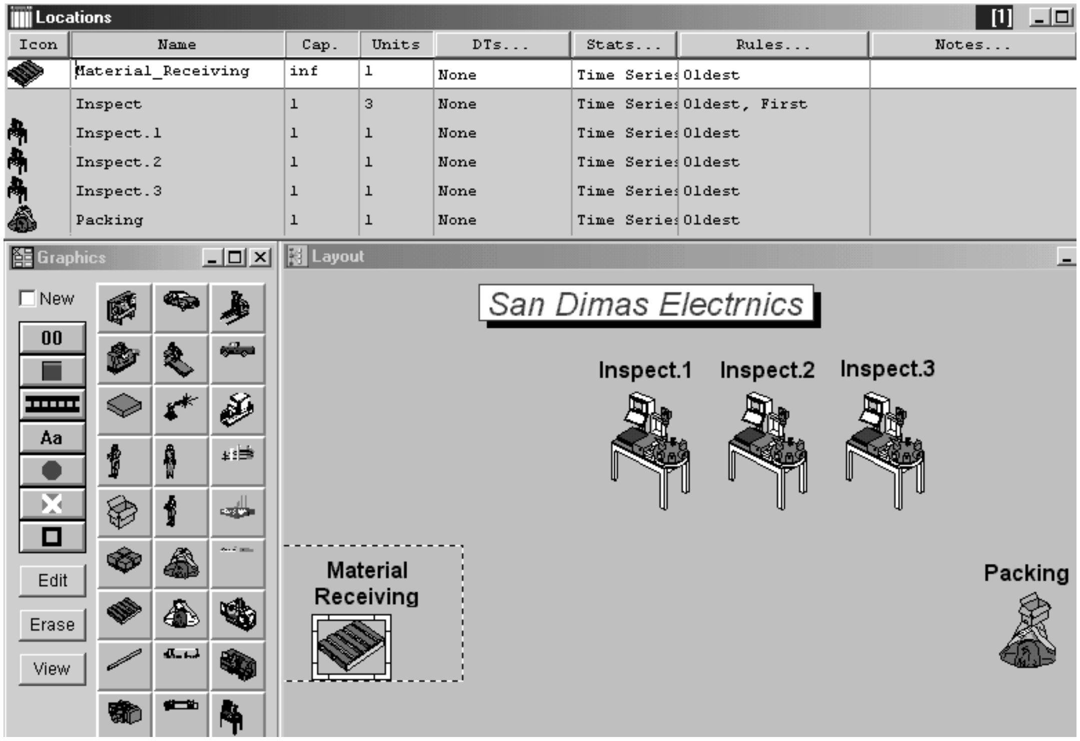
Problem Statement

At *San Dimas Electronics,* jobs arrive at three identical inspection machines according to an exponential distribution with a mean interarrival time of 12 minutes. The first available machine is selected. Processing on any of the parallel machines is normally distributed with a mean of 10 minutes and a standard deviation of 3 minutes. Upon completion, all jobs are sent to a fourth machine, where they queue up for date stamping and packing for shipment; this takes five minutes normally distributed with a standard deviation of two minutes. Completed jobs then leave the system. Run the simulation for one month (20 days, eight hours each). Calculate the average utilization of the four machines. Also, how many jobs are processed by each of the four machines?

Define a location called Inspect. Change its units to 3. Three identical parallel locations—that is, Inspect.1, Inspect.2, and Inspect.3—are thus created. Also, define a location for all the raw material to arrive (Material_Receiving). Change the capacity of this location to infinite. Define a location for Packing (Figure L7.9). Select Background Graphics from the Build menu. Make up a label “San Dimas Electronics.” Add a rectangular border. Change the font and color appropriately.

FIGURE L7.9

The locations and the layout of San Dimas Electronics.

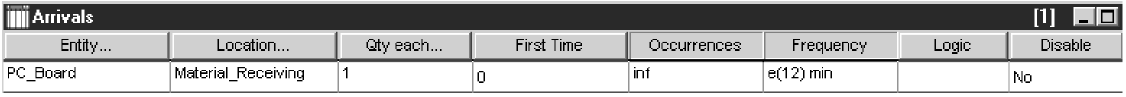


Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Material_Receiving	inf	1	None	Time Series	Oldest	
	Inspect	1	3	None	Time Series	Oldest, First	
	Inspect.1	1	1	None	Time Series	Oldest	
	Inspect.2	1	1	None	Time Series	Oldest	
	Inspect.3	1	1	None	Time Series	Oldest	
	Packing	1	1	None	Time Series	Oldest	

FIGURE L7.10

Arrivals of PCB at San Dimas Electronics.



Arrivals

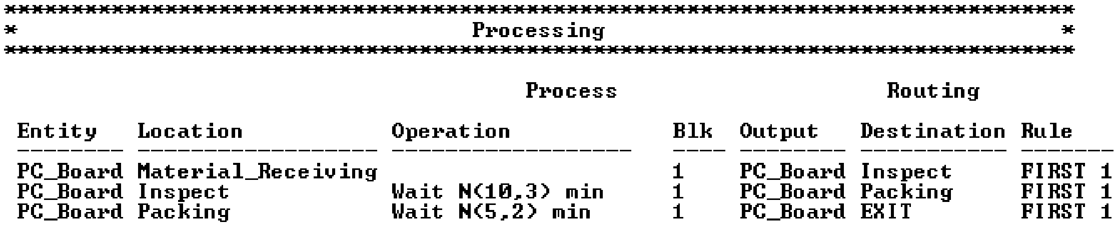
Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
PC_Board	Material_Receiving	1	0	inf	e(12) min		No

Define an entity called PCB. Define the frequency of arrival of the entity PCB as exponential with a mean interarrival time of 12 minutes (Figure L7.10). Define the process and routing at San Dimas Electronics as shown in Figure L7.11.

In the Simulation menu select Options. Enter 160 in the Run Hours box. Run the simulation model. The average utilization and the number of jobs processed at the four locations are given in Table L7.2.

FIGURE L7.11

Process and routing tables at San Dimas Electronics.



```
********************************************************************************
*                                  Processing                                  *
********************************************************************************

                                       Process                    Routing

 Entity   Location            Operation            Blk  Output   Destination Rule
 -------- ------------------- ------------------- ---- -------- ----------- -------
 PC_Board Material_Receiving                       1    PC_Board Inspect     FIRST 1
 PC_Board Inspect             Wait N(10,3) min     1    PC_Board Packing     FIRST 1
 PC_Board Packing             Wait N(5,2) min      1    PC_Board EXIT        FIRST 1
```

TABLE L7.2 Summary of Results of the Simulation Run

	Average Utilization	*Number of Jobs Processed*
Inspector1	49.5	437
Inspector2	29.7	246
Inspector3	15.9	117
Packing	41.5	798

L7.3 Routing Rules

Routing rules are used in selecting among competing downstream processes for routing an entity. Sometimes customers prefer one downstream resource to another. The selection among several competing resources may follow one of these rules:

- FIRST *available:* Select the first location that has available capacity.
- MOST *available:* Select the location that has the most available capacity.
- *By* TURN: Rotate the selection among two or more available locations.
- RANDOM: Select randomly among two or more available locations.
- *If* JOIN *request:* Select the location that satisfies a JOIN request (see the JOIN statement).
- *If* LOAD *request:* Select the location that satisfies a LOAD request (see the LOAD statement).
- *If* SEND *request:* Select the location that satisfies a SEND request (see the SEND statement).
- *Longest:* Select the location that has been unoccupied the longest.
- *Unoccupied until* FULL: Continue to select the same location until it is full.
- *If* EMPTY: Select the location only when empty and continue to select until it is full.

- *Probabilistic:* Select based on the probability specified (such as .75).
- *User condition:* Select the location that satisfies the Boolean condition specified by the user (such as AT2>5). Conditions may include any numeric expression except for location attributes, resource-specific functions, and downtime-specific functions.
- CONTINUE: Continue at the same location for additional operations. CONTINUE is allowed only for blocks with a single routing.
- *As* ALTERNATE *to:* Select as an alternate if available and if none of the above rules can be satisfied.
- *As* BACKUP: Select as a backup if the location of the first preference is down.
- DEPENDENT: Select only if the immediately preceding routing was processed.

If only one routing is defined for a routing block, use the first available, join, load, send, if empty, or continue rule. The most available, by turn, random, longest unoccupied, until full, probabilistic, and user condition rules are generally only useful when a block has multiple routings.

Problem Statement

Amar, Akbar, and Anthony are three tellers in the local branch of **Bank of India.** Figure L7.12 shows the layout of the bank. Assume that customers arrive at the bank according to a uniform distribution (mean of five minutes and half-width of four minutes). All the tellers service the customers according to another uniform

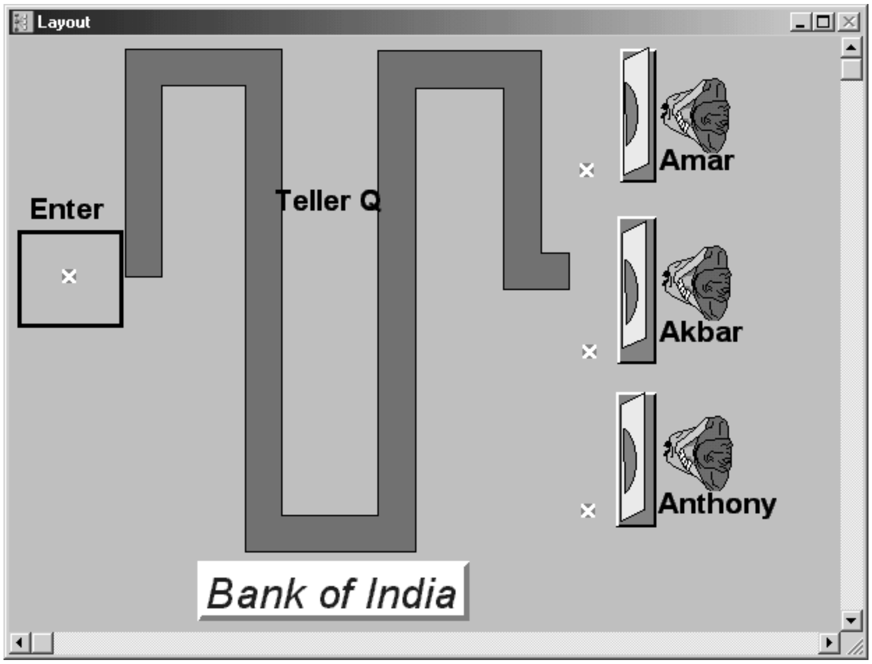
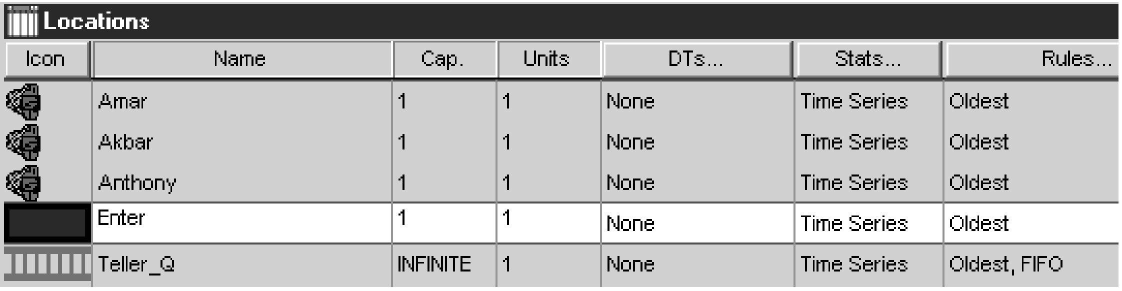


FIGURE L7.12
Layout of the Bank of India.

Figure L7.13

Locations at the Bank of India.

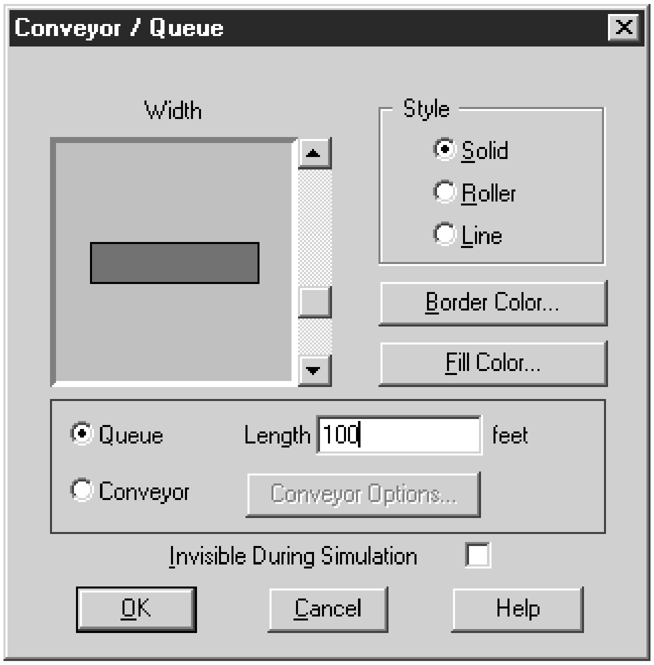


Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Amar	1	1	None	Time Series	Oldest
	Akbar	1	1	None	Time Series	Oldest
	Anthony	1	1	None	Time Series	Oldest
	Enter	1	1	None	Time Series	Oldest
	Teller_Q	INFINITE	1	None	Time Series	Oldest, FIFO

Figure L7.14

Queue menu.



Conveyor / Queue

Style: ● Solid ○ Roller ○ Line

● Queue Length 100 feet

○ Conveyor

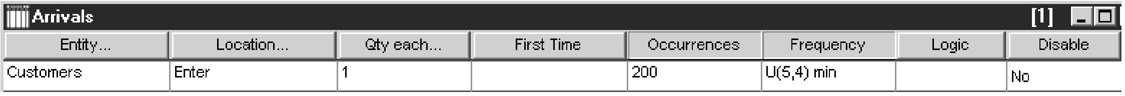
Invisible During Simulation □

distribution (mean of 10 minutes and half-width of 6 minutes). However, the customers prefer Amar to Akbar, and Akbar over Anthony. If the teller of choice is busy, the customers choose the first available teller. Simulate the system for 200 customer service completions. Estimate the teller's utilization (percentage of time busy).

The locations are defined as Akbar, Anthony, Amar, Teller_Q, and Enter as shown in Figure L7.13. The Teller_Q is exactly 100 feet long. Note that we have checked the queue option in the Conveyor/Queue menu (Figure L7.14). The

FIGURE L7.15

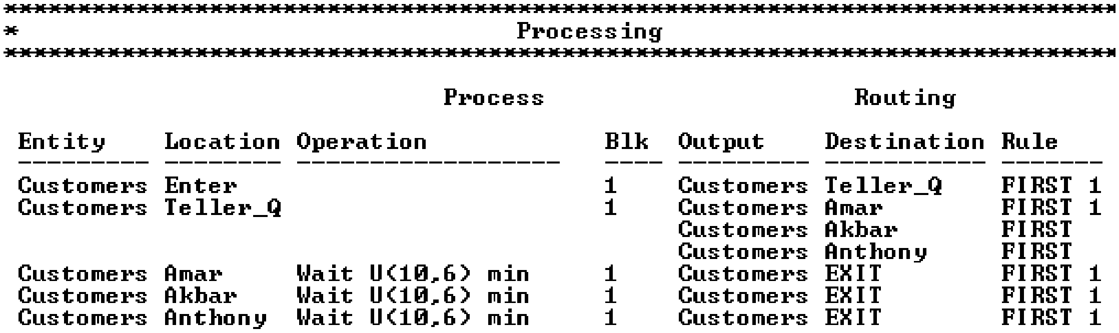
Customer arrival at the Bank of India.



Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Customers	Enter	1		200	U(5,4) min		No

FIGURE L7.16

Process and routing tables at the Bank of India.



```
***************************************************************************
*                                 Processing
***************************************************************************

                               Process                       Routing

 Entity    Location Operation           Blk  Output    Destination Rule
 --------- -------- ------------------  ---- --------- ----------- -------
 Customers Enter                        1    Customers Teller_Q    FIRST 1
 Customers Teller_Q                     1    Customers Amar        FIRST 1
                                             Customers Akbar       FIRST
                                             Customers Anthony     FIRST
 Customers Amar     Wait U(10,6) min    1    Customers EXIT        FIRST 1
 Customers Akbar    Wait U(10,6) min    1    Customers EXIT        FIRST 1
 Customers Anthony  Wait U(10,6) min    1    Customers EXIT        FIRST 1
```

TABLE L7.3 Utilization of Tellers at the Bank of India

	% Utilization	
Tellers	*Selection in Order of Preference*	*Selection by Turn*
Amar	79	63.9
Akbar	64.7	65.1
Anthony	46.9	61.5

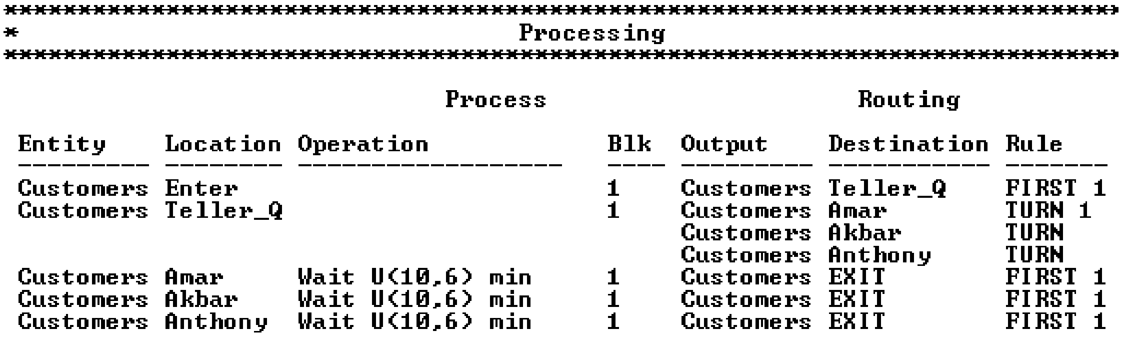
customer arrival process is shown in Figure L7.15. The processes and routings are shown in Figure L7.16. Note that the customers go to the tellers Amar, Akbar, and Anthony in the order they are specified in the routing table.

The results of the simulation model are shown in Table L7.3. Note that Amar, being the favorite teller, is much more busy than Akbar and Anthony.

If the customers were routed to the three tellers in turn (selected in rotation), the process and routing tables would be as in Figure L7.17. Note that By Turn was selected from the Rule menu in the routing table. These results of the simulation model are also shown in Table L7.3. Note that Amar, Akbar, and Anthony are now utilized almost equally.

FIGURE L7.17

Process and routing tables for tellers selected by turn.



```
********************************************************************************
*                                    Processing
********************************************************************************

                                   Process                       Routing

 Entity    Location Operation             Blk  Output    Destination Rule
 --------- -------- ------------------    ---- --------- ----------- -------
 Customers Enter                          1    Customers Teller_Q    FIRST 1
 Customers Teller_Q                       1    Customers Amar        TURN 1
                                               Customers Akbar       TURN
                                               Customers Anthony     TURN
 Customers Amar     Wait U(10,6) min      1    Customers EXIT        FIRST 1
 Customers Akbar    Wait U(10,6) min      1    Customers EXIT        FIRST 1
 Customers Anthony  Wait U(10,6) min      1    Customers EXIT        FIRST 1
```

L7.4 Variables

Variables are placeholders for either real or integer numbers that may change during the simulation. Variables are typically used for making decisions or for gathering data. Variables can be defined to track statistics and monitor other activities during a simulation run. This is useful when the built-in statistics don't capture a particular performance metric of interest. Variables might be defined to track

- The number of customers waiting in multiple queues.
- Customer waiting time during a specific time period.
- Customer time in the bank.
- Work-in-process inventory.
- Production quantity.

In ProModel two types of variables are used—local variables and global variables.

- *Global variables* are accessible from anywhere in the model and at any time. Global variables are defined through the Variables(global) editor in the Build menu. The value of a global variable may be displayed dynamically during the simulation. It can also be changed interactively. Global variables can be referenced anywhere a numeric expression is valid.
- *Local variables* are temporary variables that are used for quick convenience when a variable is needed only within a particular operation (in the Process table), move logic (in the Routing table), logic (in the Arrivals, Resources, or Subroutine tables), the initialization or termination logic (in the General Information dialog box), and so forth. Local variables are available only within the logic in which they are declared

and are not defined in the Variables edit table. They are created for each entity, downtime occurrence, or the like executing a particular section of logic. A new local variable is created for each entity that encounters an `INT` or `REAL` statement. It exists only while the entity processes the logic that declared the local variable. Local variables may be passed to subroutines as parameters and are available to macros.

A local variable must be declared before it is used. To declare a local variable, use the following syntax:

```
INT or REAL <name1>{= expression}, <name2>{= expression}
```

Examples:

```
INT HourOfDay, WIP
REAL const1 = 2.5, const2 = 5.0
INT Init_Inventory = 170
```

In Section L7.11 we show you how to use a local variable in your simulation model logic.

Problem Statement—Tracking Work in Process and Production

In the **Poly Casting Inc.** machine shop, raw castings arrive in batches of four every hour. From the raw material store they are sent to the mill, where they undergo a milling operation that takes an average of three minutes with a standard deviation of one minute (normally distributed). The milled castings go to the grinder, where they are ground for a duration that is uniformly distributed (minimum four minutes and maximum six minutes) or U(5,1). After grinding, the ground pieces go to the finished parts store. Run the simulation for 100 hours. Track the work-in-process inventory and the production quantity.

The complete simulation model layout is shown in Figure L7.18. The locations are defined as Receiving_Dock, Mill, Grinder, and Finish_Parts_Store (Figure L7.19). Castings (entity) are defined to arrive in batches of four (Qty each) every 60 minutes (Frequency) as shown in Figure L7.20. The processes and routings are shown in Figure L7.21.

Define a variable in your model to track the work-in-process inventory (`WIP`) of parts in the machine shop. Also define another variable to track the production (`PROD_QTY`) of finished parts (Figure L7.22). Note that both of these are integer type variables.

In the process table, add the following operation statement in the Receiving location (Figure L7.21).

```
WIP = WIP + 1
```

Add the following operation statements in the outgoing Finish_Parts_Store location (Figure L7.21):

```
WIP = WIP - 1
PROD_QTY = PROD_QTY + 1
```

Alternatively, these three statements could be written as `INC WIP`, `DEC WIP`, and `INC PROD_QTY`.

FIGURE L7.18

Layout of Poly Casting Inc.

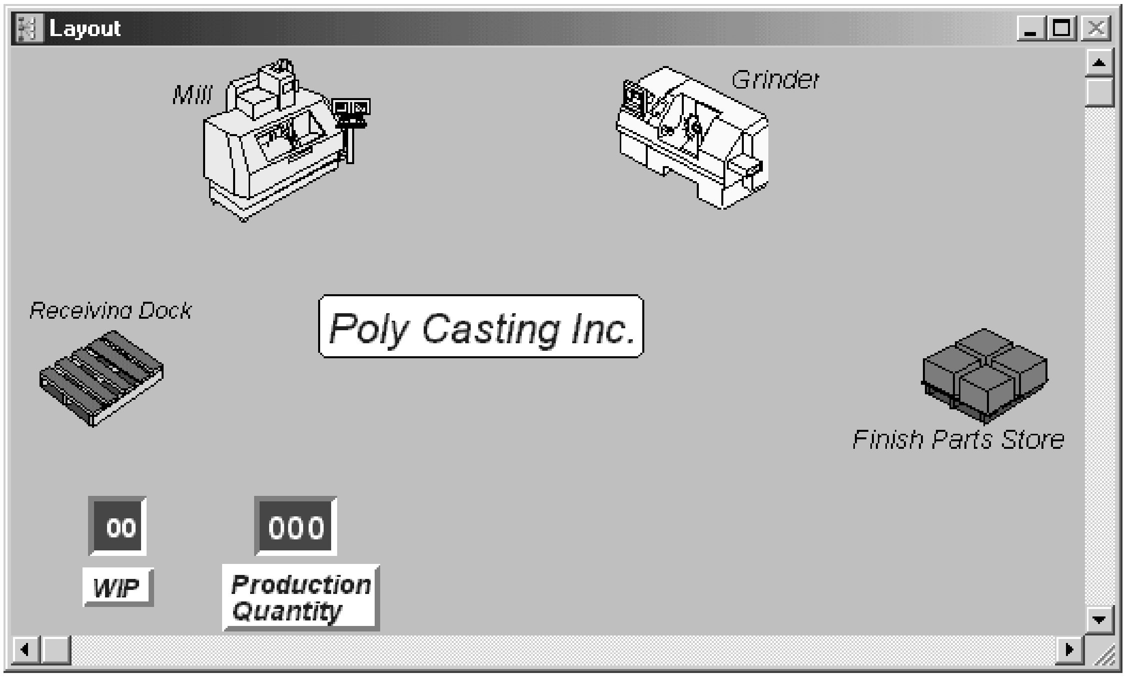
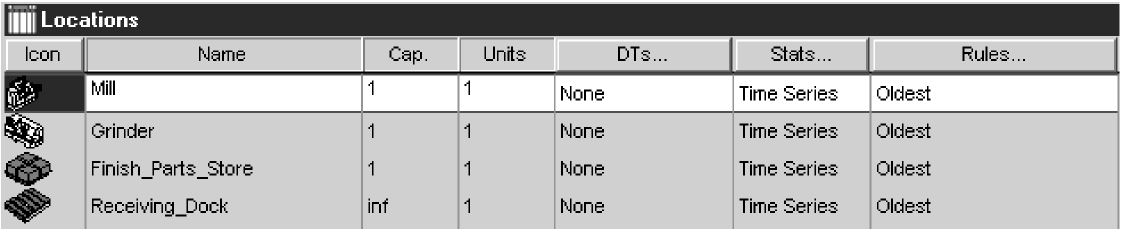


FIGURE L7.19

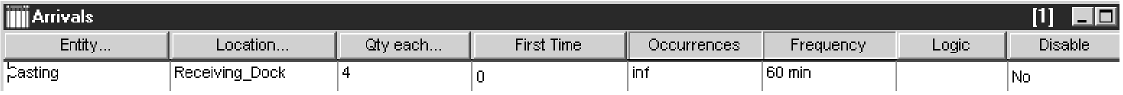
Locations at Poly Casting Inc.


Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Mill	1	1	None	Time Series	Oldest
	Grinder	1	1	None	Time Series	Oldest
	Finish_Parts_Store	1	1	None	Time Series	Oldest
	Receiving_Dock	inf	1	None	Time Series	Oldest

FIGURE L7.20

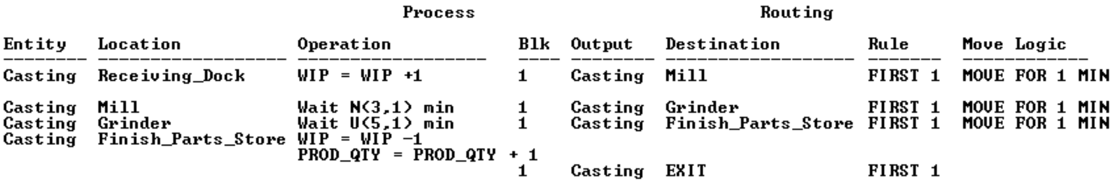
Arrival of castings at Poly Casting Inc.


Arrivals [1]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Casting	Receiving_Dock	4	0	inf	60 min		No

FIGURE L7.21

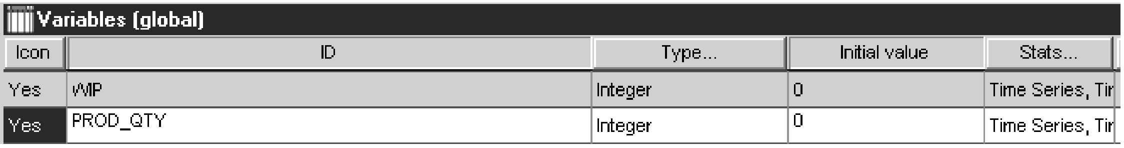
Processes and routings for the Poly Casting Inc. model.



```
                                    Process                               Routing

Entity   Location            Operation           Blk  Output    Destination          Rule     Move Logic
-------- ------------------- ------------------- ---- --------- -------------------- -------  ------------
Casting  Receiving_Dock      WIP = WIP +1        1    Casting   Mill                 FIRST 1  MOVE FOR 1 MIN

Casting  Mill                Wait N(3,1) min     1    Casting   Grinder              FIRST 1  MOVE FOR 1 MIN
Casting  Grinder             Wait U(5,1) min     1    Casting   Finish_Parts_Store   FIRST 1  MOVE FOR 1 MIN
Casting  Finish_Parts_Store  WIP = WIP -1
                             PROD_QTY = PROD_QTY + 1
                                                 1    Casting   EXIT                 FIRST 1
```

FIGURE L7.22

Variables for the Poly Casting Inc. model.



Variables (global)

Icon	ID	Type...	Initial value	Stats...
Yes	WIP	Integer	0	Time Series, Tir
Yes	PROD_QTY	Integer	0	Time Series, Tir

L7.5 Uncertainty in Routing—Track Defects and Rework

Sometimes an entity needs to be routed to a destination from a list of available locations on a probabilistic basis. The location to which a specific entity is routed remains uncertain. A certain percentage of entities will be routed to a specific destination. The total of these percentages, however, should be 100 percent.

After processing is completed, a workpiece is frequently inspected and checked for correctness. If found to be good, the workpiece moves on to the next operation. If found to be defective, it is either scrapped or reworked. For rework, the workpiece is sent to one of the earlier operations. The inspection process is carried out either on all workpieces or on a sample of workpieces.

Problem Statement

Poly Casting Inc. in Section L7.4 decides to add an inspection station at the end of the machine shop, after the grinding operation. After inspection, 30 percent of the widgets are sent back to the mill for rework, 10 percent are sent back to the grinder for rework, and 5 percent are scrapped. The balance, 55 percent, pass inspection and go on to the finished parts store. The inspection takes a time that is triangularly distributed with a minimum of 4, mode of 5, and maximum of 6 minutes. The process times for rework are the same as those for new jobs. Track the amount of rework at the mill and the grinder. Also track the amount of scrapped parts and finished production. Run the simulation for 100 hours.

The locations are defined as mill, grinder, inspection, finish parts store, receiving dock, scrap parts, mill rework, and grind rework as shown in Figure L7.23.

FIGURE L7.23

Simulation model layout for Poly Castings Inc. with inspection.

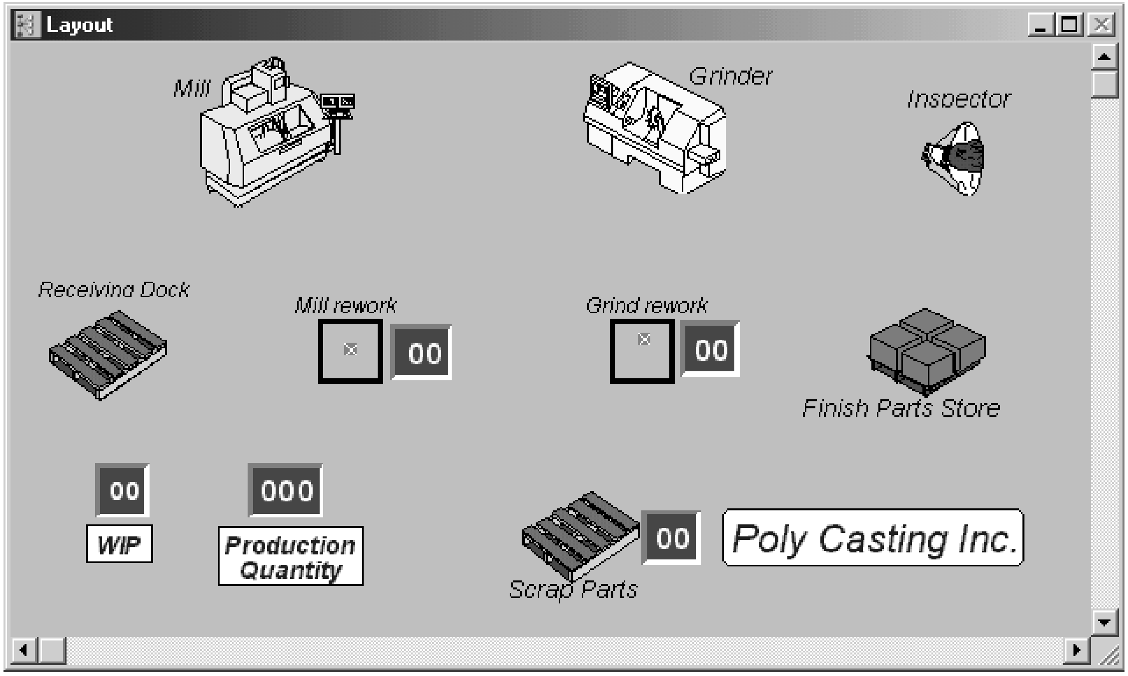
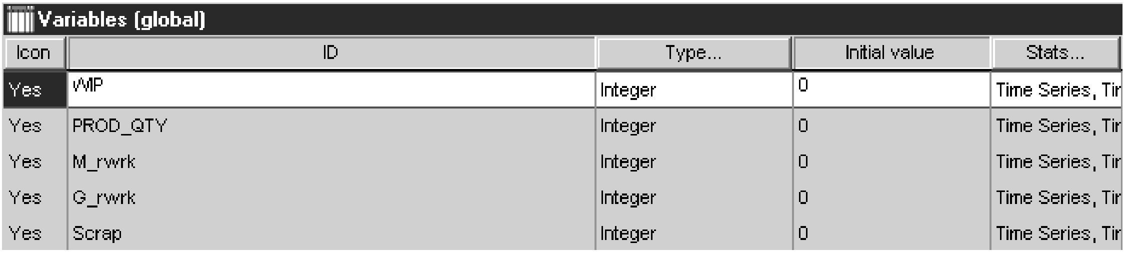


FIGURE L7.24

Variables for the Poly Castings Inc. with inspection model.



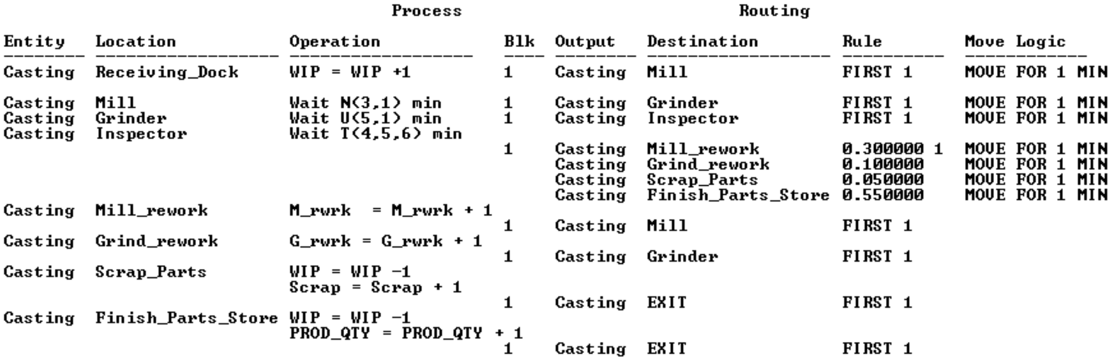
Variables (global)

Icon	ID	Type...	Initial value	Stats...
Yes	WIP	Integer	0	Time Series, Tir
Yes	PROD_QTY	Integer	0	Time Series, Tir
Yes	M_rwrk	Integer	0	Time Series, Tir
Yes	G_rwrk	Integer	0	Time Series, Tir
Yes	Scrap	Integer	0	Time Series, Tir

The last four locations are defined with infinite capacity. The arrivals of castings are defined in batches of four every hour. Next we define five variables (Figure L7.24) to track work in process, production quantity, mill rework, grind rework, and scrap quantity. The processes and routings are defined as in Figure L7.25.

FIGURE L7.25

Processes and routings for the Poly Castings Inc. with inspection model.



```
                                        Process                              Routing

Entity   Location           Operation          Blk  Output   Destination        Rule       Move Logic
-------- ------------------ ------------------ ---- -------- ------------------ ---------- ------------
Casting  Receiving_Dock     WIP = WIP +1       1    Casting  Mill               FIRST 1    MOVE FOR 1 MIN

Casting  Mill               Wait N(3,1) min    1    Casting  Grinder            FIRST 1    MOVE FOR 1 MIN
Casting  Grinder            Wait U(5,1) min    1    Casting  Inspector          FIRST 1    MOVE FOR 1 MIN
Casting  Inspector          Wait T(4,5,6) min
                                               1    Casting  Mill_rework        0.300000 1 MOVE FOR 1 MIN
                                                    Casting  Grind_rework       0.100000   MOVE FOR 1 MIN
                                                    Casting  Scrap_Parts        0.050000   MOVE FOR 1 MIN
                                                    Casting  Finish_Parts_Store 0.550000   MOVE FOR 1 MIN
Casting  Mill_rework        M_rwrk  = M_rwrk + 1
                                               1    Casting  Mill               FIRST 1
Casting  Grind_rework       G_rwrk = G_rwrk + 1
                                               1    Casting  Grinder            FIRST 1
Casting  Scrap_Parts        WIP = WIP -1
                            Scrap = Scrap + 1
                                               1    Casting  EXIT               FIRST 1
Casting  Finish_Parts_Store WIP = WIP -1
                            PROD_QTY = PROD_QTY + 1
                                               1    Casting  EXIT               FIRST 1
```

L7.6 Batching Multiple Entities of Similar Type

L7.6.1 Temporary Batching—`GROUP/UNGROUP`

Frequently we encounter a situation where widgets are batched and processed together. After processing is over, the workpieces are unbatched again. For example, an autoclave or an oven is fired with a batch of jobs. After heating, curing, or bonding is done, the batch of jobs is separated and the individual jobs go their way. The individual pieces retain their properties during the batching and processing activities.

For such temporary batching, use the `GROUP` statement. For unbatching, use the `UNGROUP` statement. One may group entities by individual entity type by defining a process record for the type to group, or group them irrespective of entity type by defining an `ALL` process record. ProModel maintains all of the characteristics and properties of the individual entities of the grouped entities and allows them to remain with the individual entities after an `UNGROUP` command. Note that the capacity of the location where Grouping occurs must be at least as large as the group size.

Problem Statement

El Segundo Composites receive orders for aerospace parts that go through cutting, lay-up, and bonding operations. Cutting and lay-up take uniform (20,5) minutes and uniform (30,10) minutes. The bonding is done in an autoclave in batches of five parts and takes uniform (100,10) minutes. After bonding, the parts go to the shipment clerk individually. The shipment clerk takes normal (20,5) minutes to get each part ready for shipment. The orders are received on average once every 60 minutes, exponentially distributed. The time to transport these parts from one machine to another takes on average 15 minutes. Figure out the amount of WIP in the shop. Simulate for six months or 1000 working hours.

The locations are defined as Ply_Cutting, LayUp, Oven, Order_Q, Ship_Clerk, and Ship_Q. The layout of El Segundo Composites is shown in Figure L7.26. The processes and the routings are shown in Figure L7.27. A variable WIP is defined to keep track of the amount of work in process in the shop. The WIP value history over the simulated 1000 hours is shown in Figure L7.28. Note that the average WIP in the shop has been around 7, although on rare occasions the WIP has been as much as 23.

FIGURE L7.26

Layout of El Segundo Composites.

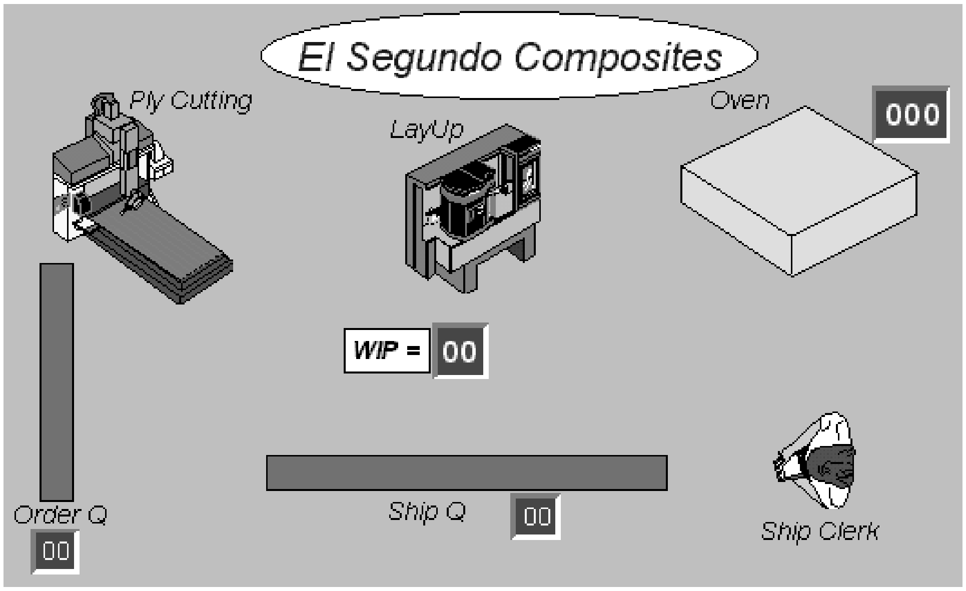
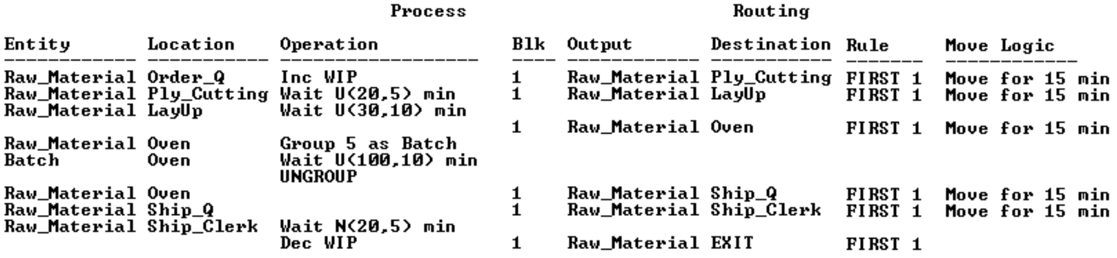


FIGURE L7.27

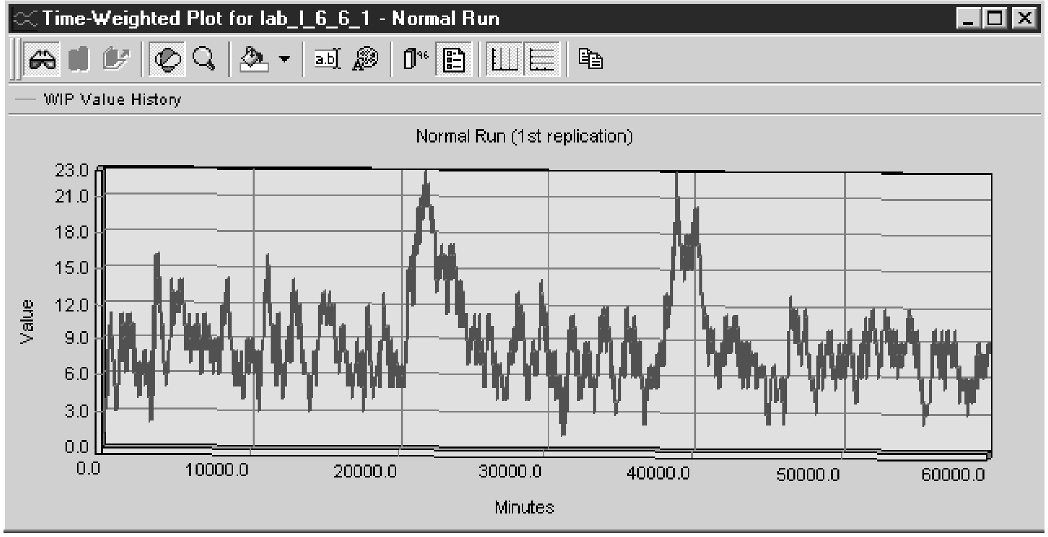
Process and routing tables for El Segundo Composites.



```
                                   Process                          Routing

Entity       Location    Operation                Blk  Output       Destination  Rule     Move Logic
------------ ----------- ------------------       ---- ------------ -----------  -------  ------------
Raw_Material Order_Q     Inc WIP                  1    Raw_Material Ply_Cutting  FIRST 1  Move for 15 min
Raw_Material Ply_Cutting Wait U(20,5) min         1    Raw_Material LayUp        FIRST 1  Move for 15 min
Raw_Material LayUp       Wait U(30,10) min
                                                  1    Raw_Material Oven         FIRST 1  Move for 15 min
Raw_Material Oven        Group 5 as Batch
Batch        Oven        Wait U(100,10) min
                         UNGROUP
Raw_Material Oven                                 1    Raw_Material Ship_Q       FIRST 1  Move for 15 min
Raw_Material Ship_Q                               1    Raw_Material Ship_Clerk   FIRST 1  Move for 15 min
Raw_Material Ship_Clerk  Wait N(20,5) min
                         Dec WIP                  1    Raw_Material EXIT         FIRST 1
```

FIGURE L7.28

Work in process value history.



L7.6.2 Permanent Batching—`COMBINE`

In some situations, multiple entities of the same type or different types are batched together permanently. After batching, a single batch entity is formed, alternatively with a different name. In such permanent batching, the characteristics of the individual entities are lost; that is, the individual entities cannot be ungrouped later.

When defining the location, the capacity of the location where you use the `COMBINE` statement should be at least as large as the combined quantity.

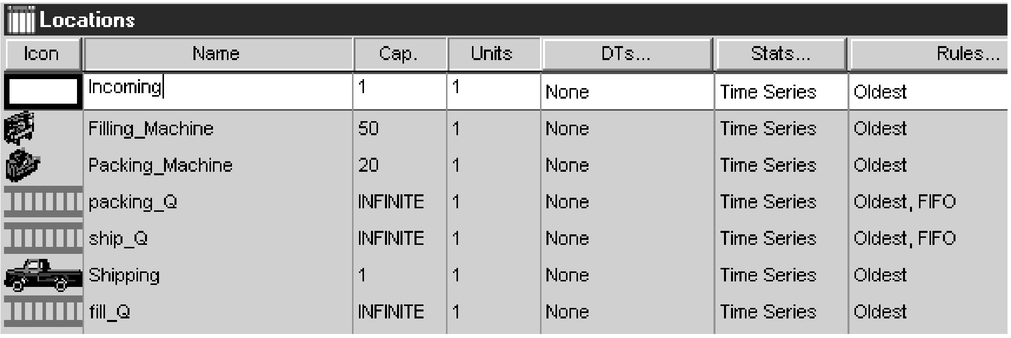
Problem Statement

At the Garden Reach plant of the **Calcutta Tea Company,** the filling machine fills empty cans with 50 bags of the best Darjeeling tea at the rate of one can every 1±0.5 seconds uniformly distributed. The tea bags arrive to the packing line with a mean interarrival time of one second exponentially distributed. The filled cans go to a packing machine where 20 cans are combined into one box. The packing operation takes uniform (20±10) seconds. The boxes are shipped to the dealers. This facility runs 24 hours a day. Simulate for one day.

The various locations at the Calcutta Tea Company plant are shown in Figure L7.29. Three entities (Teabag, Can, and Box) are defined next. Teabags are defined to arrive with exponential interarrival time with a mean of one second. The processes and routing logic are shown in Figure L7.30. The layout of the Calcutta Tea Company plant is shown in Figure L7.31.

FIGURE L7.29

Locations at the Calcutta Tea Company.

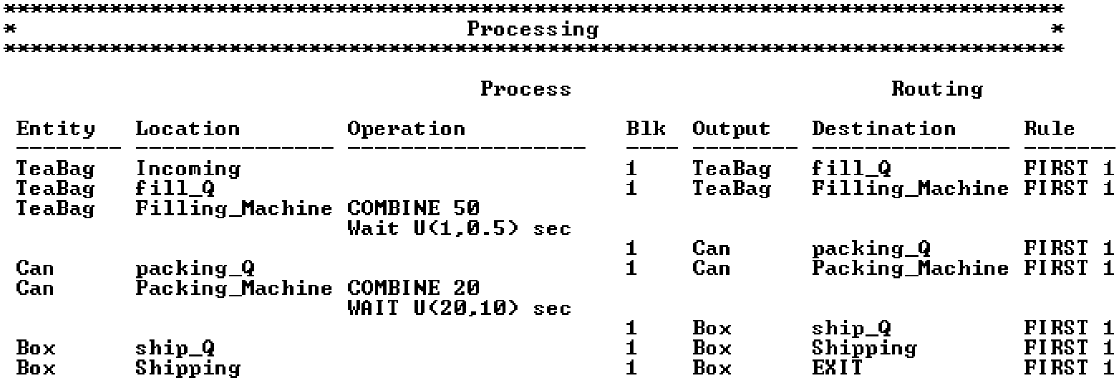


Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	Incoming	1	1	None	Time Series	Oldest
	Filling_Machine	50	1	None	Time Series	Oldest
	Packing_Machine	20	1	None	Time Series	Oldest
	packing_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	ship_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	Shipping	1	1	None	Time Series	Oldest
	fill_Q	INFINITE	1	None	Time Series	Oldest

FIGURE L7.30

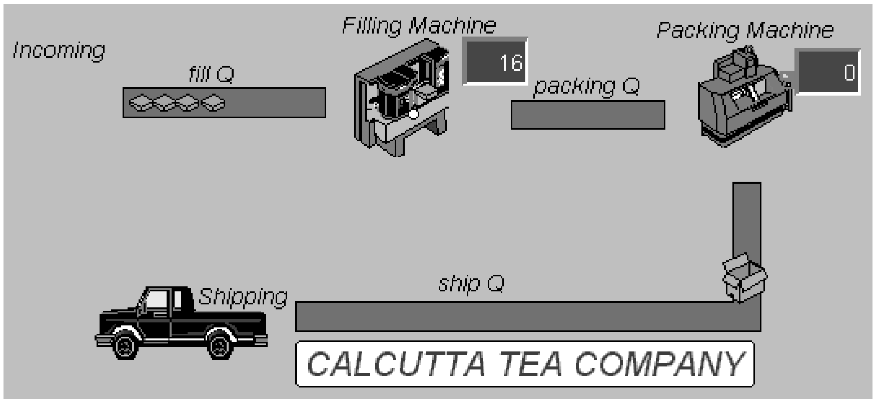
Process and routing tables at the Calcutta Tea Company.



```
********************************************************************************
*                                  Processing                                  *
********************************************************************************

                                   Process                        Routing

 Entity   Location         Operation            Blk  Output   Destination      Rule
 -------- ---------------- ------------------   ---- -------- ---------------- -------
 TeaBag   Incoming                              1    TeaBag   fill_Q           FIRST 1
 TeaBag   fill_Q                                1    TeaBag   Filling_Machine  FIRST 1
 TeaBag   Filling_Machine  COMBINE 50
                           Wait U(1,0.5) sec
                                                1    Can      packing_Q        FIRST 1
 Can      packing_Q                             1    Can      Packing_Machine  FIRST 1
 Can      Packing_Machine  COMBINE 20
                           WAIT U(20,10) sec
                                                1    Box      ship_Q           FIRST 1
 Box      ship_Q                                1    Box      Shipping         FIRST 1
 Box      Shipping                              1    Box      EXIT             FIRST 1
```

FIGURE L7.31

Layout of the Calcutta Tea Company.



L7.7 Attaching One or More Entities to Another Entity

L7.7.1 Permanent Attachment—JOIN

Sometimes one or more entities are attached permanently to another entity, as in an assembly operation. The assembly process is a permanent bonding: the assembled entities lose their separate identities and properties. The individual entities that are attached cannot be separated again.

The *join* process is used to permanently assemble two or more individual entities together. For every JOIN statement, there must be a corresponding If Join Request rule. Joining is a two-step process:

1. Use the JOIN statement at the designated assembly location.
2. Use the join routing rule for all joining entities.

One of the joining entities, designated as the "base" entity, issues the JOIN statement. All other joining entities must travel to the assembly location on an If Join Request routing rule.

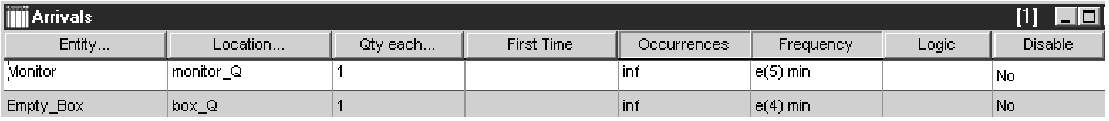
Problem Statement

At **Shipping Boxes Unlimited** computer monitors arrive at Monitor_Q at the rate of one every 15 minutes (exponential) and are moved to the packing table. Boxes arrive at Box_Q, at an average rate of one every 15 minutes (exponential) and are also moved to the packing table. At the packing table, monitors are packed into boxes. The packing operation takes normal (5,1) minutes. Packed boxes are sent to the inspector (Inspect_Q). The inspector checks the contents of the box and tallies with the packing slip. Inspection takes normal (4,2) minutes. After inspection, the boxes are loaded into trucks at the shipping dock (Shipping_Q). The loading takes uniform (5,1) minutes. Simulate for 100 hours. Track the number of monitors shipped and the WIP of monitors in the system.

The locations at Shipping Boxes Unlimited are defined as Monitor_Q, Box_Q, Shipping_Q, Inspect_Q, Shipping_Dock, Packing_Table, and Inspector. All the queue locations are defined with a capacity of infinity. Three entities (Monitor, Empty Box, and Full_Box) are defined next. The arrivals of monitors and empty boxes are shown in Figure L7.32. The processes and routings are shown in Figure L7.33. A snapshot of the simulation model is captured in

FIGURE L7.32

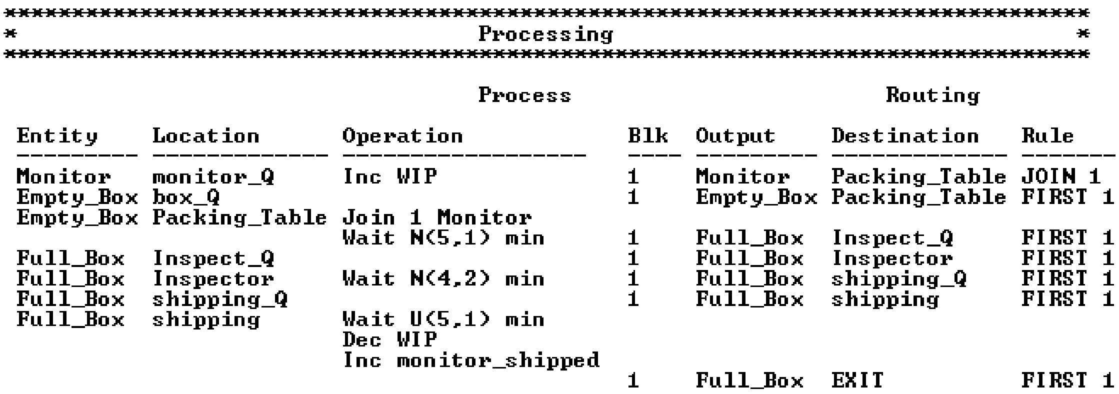
Arrival of monitors and empty boxes at Shipping Boxes Unlimited.



Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Monitor	monitor_Q	1		inf	e(5) min		No
Empty_Box	box_Q	1		inf	e(4) min		No

FIGURE L7.33

Processes and routings for Shipping Boxes Unlimited.



```
*********************************************************************************
*                                  Processing                                   *
*********************************************************************************

                                          Process                        Routing

 Entity    Location      Operation            Blk  Output    Destination   Rule
 --------- ------------- ------------------   ---- --------- ------------- -------
 Monitor   monitor_Q     Inc WIP              1    Monitor   Packing_Table JOIN 1
 Empty_Box box_Q                              1    Empty_Box Packing_Table FIRST 1
 Empty_Box Packing_Table Join 1 Monitor
                         Wait N(5,1) min      1    Full_Box  Inspect_Q     FIRST 1
 Full_Box  Inspect_Q                          1    Full_Box  Inspector     FIRST 1
 Full_Box  Inspector     Wait N(4,2) min      1    Full_Box  shipping_Q    FIRST 1
 Full_Box  shipping_Q                         1    Full_Box  shipping      FIRST 1
 Full_Box  shipping      Wait U(5,1) min
                         Dec WIP
                         Inc monitor_shipped
                                              1    Full_Box  EXIT          FIRST 1
```

FIGURE L7.34

A snapshot of the simulation model for Shipping Boxes Unlimited.

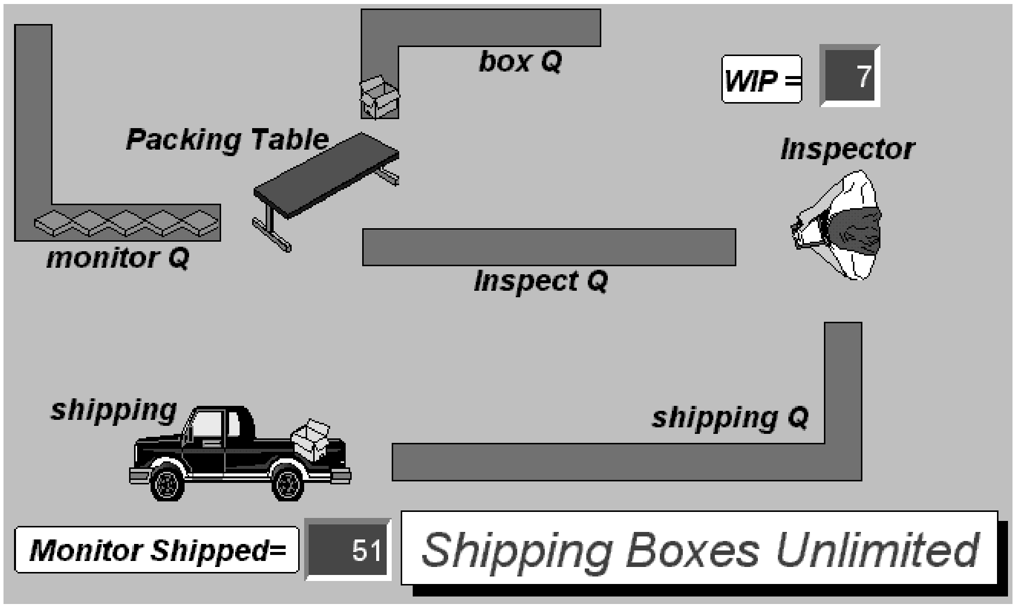
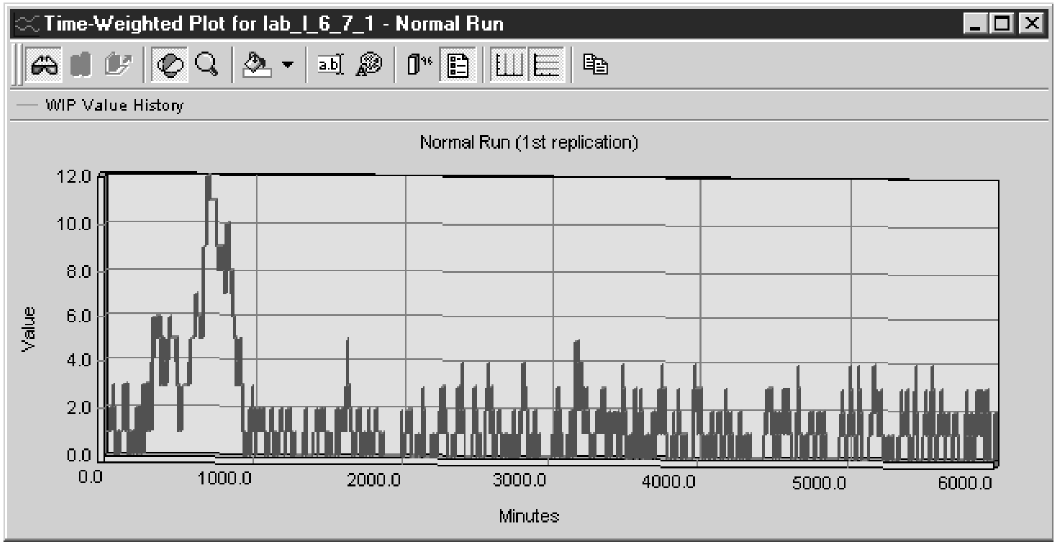


Figure L7.34. The plot of the work-in-process inventory for the 100 hours of simulation run is shown in Figure L7.35. Note that the work-in-process inventory rises to as much as 12 in the beginning. However, after achieving steady state, the WIP inventory stays mostly within the range of 0 to 3.

FIGURE L7.35

Time-weighted plot of the WIP inventory at Shipping Boxes Unlimited.



L7.7.2 Temporary Attachment—LOAD/UNLOAD

Sometimes one or more entities are attached to another entity temporarily, processed or moved together, and detached later in the system. In ProModel, we use the LOAD and UNLOAD statements to achieve that goal. The properties of the individual attached entities are preserved. This of course requires more memory space and hence should be used only when necessary.

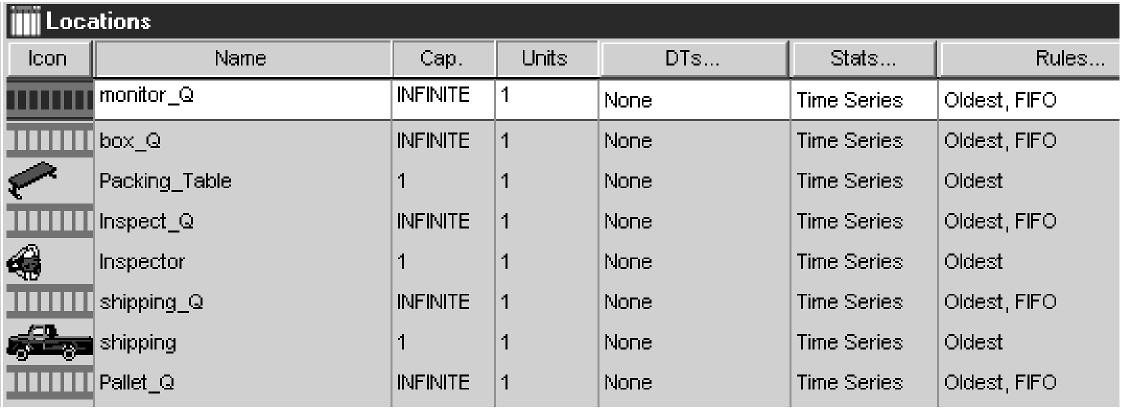
Problem Statement

For the **Shipping Boxes Unlimited** problem in Section L7.7.1, assume the inspector places (loads) a packed box on an empty pallet. The loading takes anywhere from two to four minutes, uniformly distributed. The loaded pallet is sent to the shipping dock and waits in the shipping queue. At the shipping dock, the packed boxes are unloaded from the pallet. The unloading time is also uniformly distributed; U(3,1) min. The boxes go onto a waiting truck. The empty pallet is returned, via the pallet queue, to the inspector. One pallet is used and recirculated in the system. Simulate for 100 hours. Track the number of monitors shipped and the WIP of monitors.

The locations defined in this model are Monitor_Q, Box_Q, Inspect_Q, Shipping_Q, Pallet_Q, Packing_Table, Inspector, and Shipping_Dock (Figure L7.36). All queue locations have infinite capacity. The layout of *Shipping Boxes Unlimited* is shown in Figure L7.37. Five entities are defined: Monitor, Box, Empty_Box, Empty_Pallet, and Full_Pallet as shown in Figure L7.38. The

FIGURE L7.36

The locations at Shipping Boxes Unlimited.

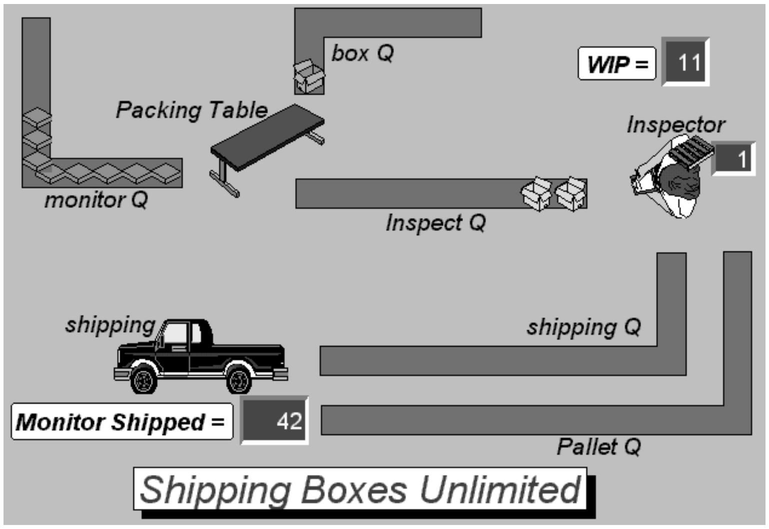


Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...
	monitor_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	box_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	Packing_Table	1	1	None	Time Series	Oldest
	Inspect_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	Inspector	1	1	None	Time Series	Oldest
	shipping_Q	INFINITE	1	None	Time Series	Oldest, FIFO
	shipping	1	1	None	Time Series	Oldest
	Pallet_Q	INFINITE	1	None	Time Series	Oldest, FIFO

FIGURE L7.37

Boxes loaded on pallets at Shipping Boxes Unlimited.



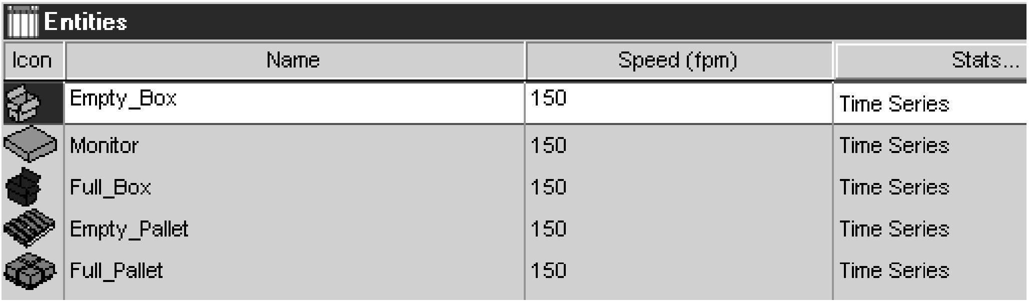
arrivals of monitors, empty boxes, and empty pallets are shown in Figure L7.39. The processes and routings are shown in Figure L7.40. Note that comments can be inserted in a line of code as follows (Figure L7.40):

```
/* inspection time */
```

The plot of the work-in-process inventory for the 100 hours of simulation run is presented in Figure L7.41. Note that after the initial transient period (≅1200 minutes), the work-in-process inventory drops and stays mostly in a range of 0–2.

FIGURE L7.38

Entities at Shipping Boxes Unlimited.

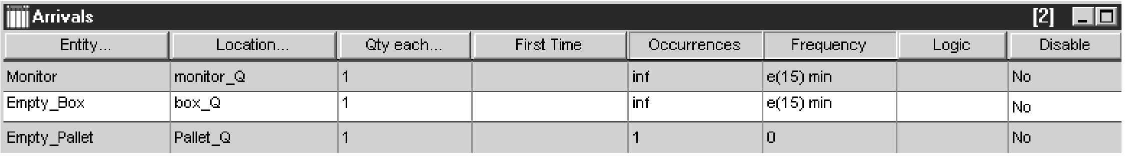


Entities

Icon	Name	Speed (fpm)	Stats...
	Empty_Box	150	Time Series
	Monitor	150	Time Series
	Full_Box	150	Time Series
	Empty_Pallet	150	Time Series
	Full_Pallet	150	Time Series

FIGURE L7.39

Arrival of monitors, empty boxes, and empty pallets at Shipping Boxes Unlimited.

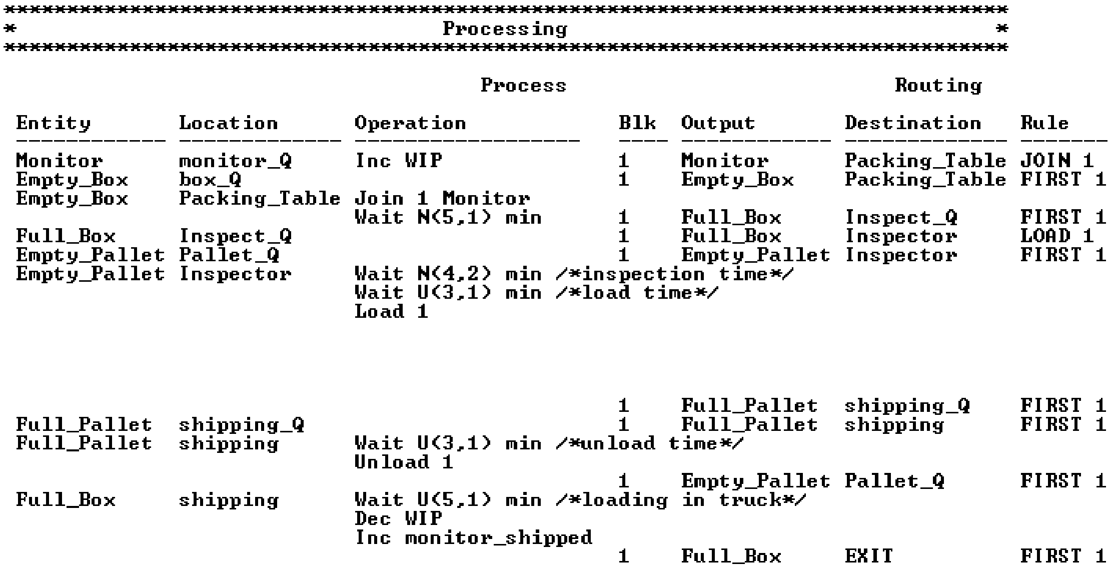


Arrivals [2]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Monitor	monitor_Q	1		inf	e(15) min		No
Empty_Box	box_Q	1		inf	e(15) min		No
Empty_Pallet	Pallet_Q	1		1	0		No

FIGURE L7.40

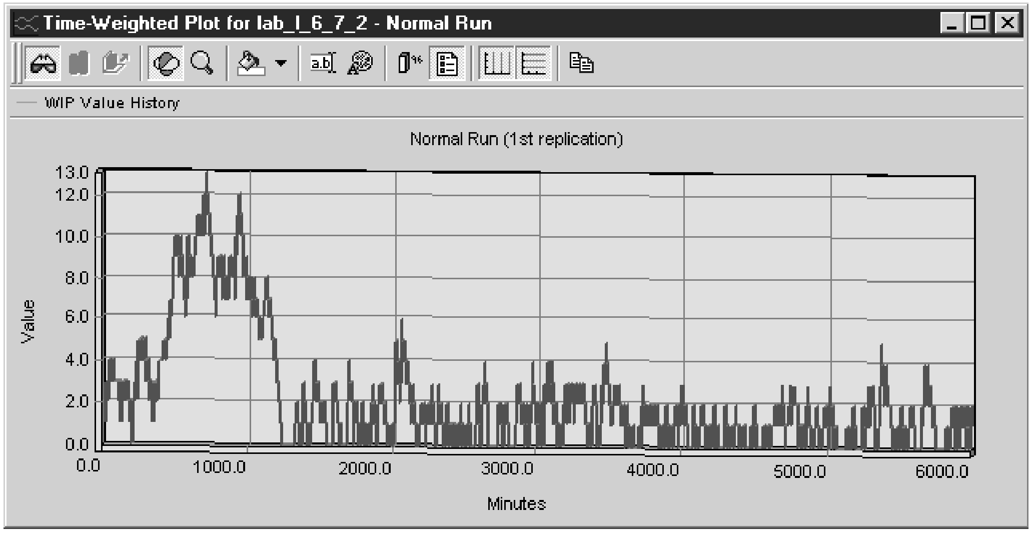
Process and routing tables at Shipping Boxes Unlimited.



```
*******************************************************************************
*                                  Processing                                 *
*******************************************************************************

                                      Process                          Routing

 Entity       Location      Operation          Blk  Output       Destination   Rule
 ------------ ------------- ------------------ ---- ------------ ------------- -------
 Monitor      monitor_Q     Inc WIP            1    Monitor      Packing_Table JOIN 1
 Empty_Box    box_Q                            1    Empty_Box    Packing_Table FIRST 1
 Empty_Box    Packing_Table Join 1 Monitor
                            Wait N(5,1) min    1    Full_Box     Inspect_Q     FIRST 1
 Full_Box     Inspect_Q                        1    Full_Box     Inspector     LOAD 1
 Empty_Pallet Pallet_Q                         1    Empty_Pallet Inspector     FIRST 1
 Empty_Pallet Inspector     Wait N(4,2) min /*inspection time*/
                            Wait U(3,1) min /*load time*/
                            Load 1



                                               1    Full_Pallet  shipping_Q    FIRST 1
 Full_Pallet  shipping_Q                       1    Full_Pallet  shipping      FIRST 1
 Full_Pallet  shipping      Wait U(3,1) min /*unload time*/
                            Unload 1
                                               1    Empty_Pallet Pallet_Q      FIRST 1
 Full_Box     shipping      Wait U(5,1) min /*loading in truck*/
                            Dec WIP
                            Inc monitor_shipped
                                               1    Full_Box     EXIT          FIRST 1
```

FIGURE L7.41

Time-weighted plot of the WIP inventory at Shipping Boxes Unlimited.



L7.8 Accumulation of Entities

Sometimes we need to hold entities at a location until a certain number accumulate. Once this critical limit is reached, the entities are released for further processing downstream. This allows us to model a certain type of batching that involves simply the accumulation of parts and their subsequent release—all at a single location. The capacity of the location at which the accumulation takes place must be at least as large as the accumulation limit or the amount of accumulation.

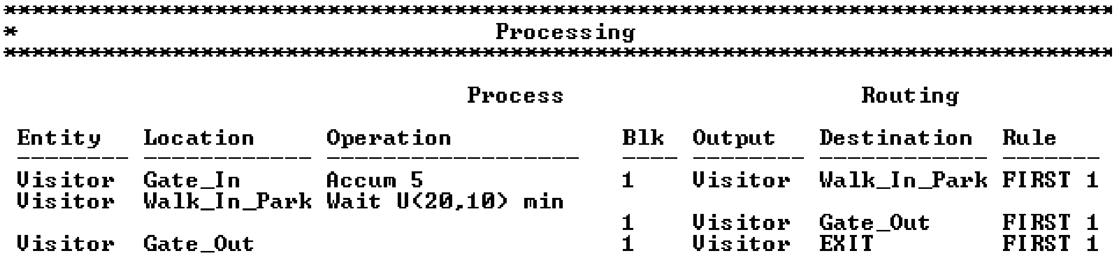
Problem Statement

Visitors arrive at **California Adventure Park** in groups that vary in size from two to four (uniformly distributed). The average time between arrival of two successive groups is five minutes, exponentially distributed. All visitors wait in front of the gate until five visitors have accumulated. At that point the gate opens and allows the visitors to enter the park. On average, a visitor spends 20±10 minutes (uniformly distributed) in the park. Simulate for 1000 hours. Track how many visitors are waiting outside the gate and how many are in the park.

Three locations (Gate_In, Walk_In_Park, and Gate_Out) are defined in this model. Visitor is defined as the entity. The processes and the layout of the adventure park are shown in Figures L7.42 and L7.43.

FIGURE L7.42

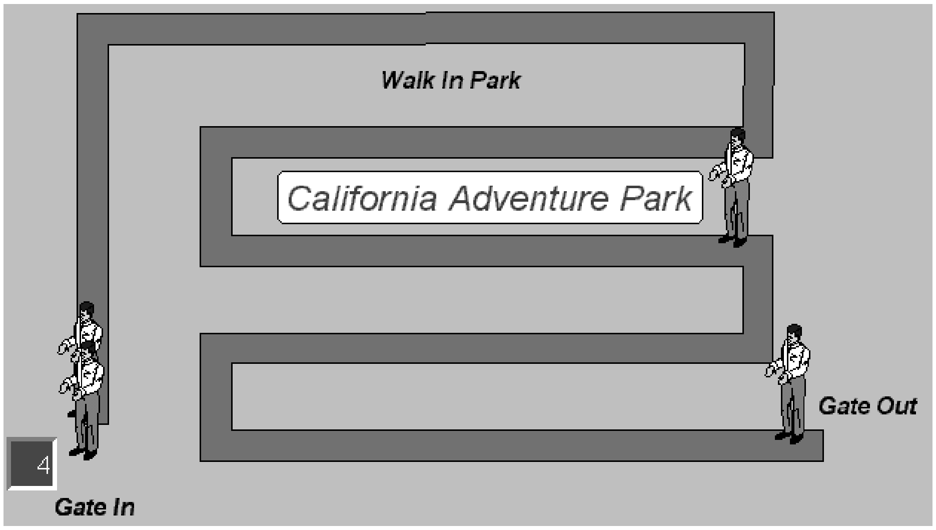
Process and routing tables for California Adventure Park.



```
********************************************************************************
*                                   Processing
********************************************************************************

                                   Process                      Routing

 Entity   Location     Operation            Blk  Output   Destination  Rule
 -------- ------------ -------------------  ---- -------- ------------ -------
 Visitor  Gate_In      Accum 5              1    Visitor  Walk_In_Park FIRST 1
 Visitor  Walk_In_Park Wait U(20,10) min
                                            1    Visitor  Gate_Out     FIRST 1
 Visitor  Gate_Out                          1    Visitor  EXIT         FIRST 1
```

FIGURE L7.43

Layout of California Adventure Park.



L7.9 Splitting of One Entity into Multiple Entities

When a single incoming entity is divided into multiple entities at a location and processed individually, we can use the SPLIT construct. The SPLIT command splits the entity into a specified number of entities, and optionally assigns them a new name. The resulting entities will have the same attribute values as the original entity. Use SPLIT to divide a piece of raw material into components, such as a silicon wafer into silicon chips or six-pack of cola into individual cans.

Problem Statement

The cafeteria at **San Dimas High School** receives 10 cases of milk from a vendor each day before the lunch recess. On receipt, the cases are split open and individual cartons (10 per case) are stored in the refrigerator for distribution to students during lunchtime. The distribution of milk cartons takes triangular(.1,.15,.2) minute per student. The time to split open the cases takes a minimum of 5 minutes and a maximum of 7 minutes (uniform distribution) per case. Moving the cases from receiving to the refrigerator area takes five minutes per case, and moving the cartons from the refrigerator to the distribution area takes 0.2 minute per carton. Students wait in the lunch line to pick up one milk carton each. There are only 100 students at this high school. Students show up for lunch with a mean interarrival time of 1 minute (exponential). On average, how long does a carton stay in the cafeteria before being distributed and consumed? What are the maximum and the minimum times of stay? Simulate for 10 days.

The layout of the San Dimas High School cafeteria is shown in Figure L7.44. Three entities—Milk_Case, Milk_Carton, and Student—are defined. Ten milk cases arrive with a frequency of 480 minutes. One hundred students show up for lunch each day. The arrival of students and milk cases is shown in Figure L7.45. The processing and routing logic is shown in Figure L7.46.

FIGURE L7.44

Layout of the San Dimas High School cafeteria.

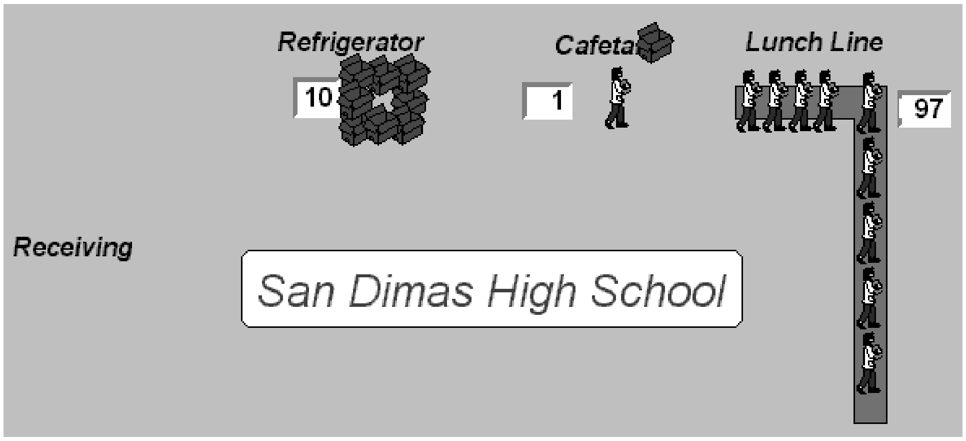
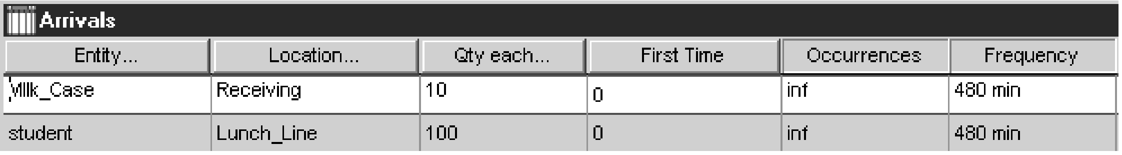


FIGURE L7.45

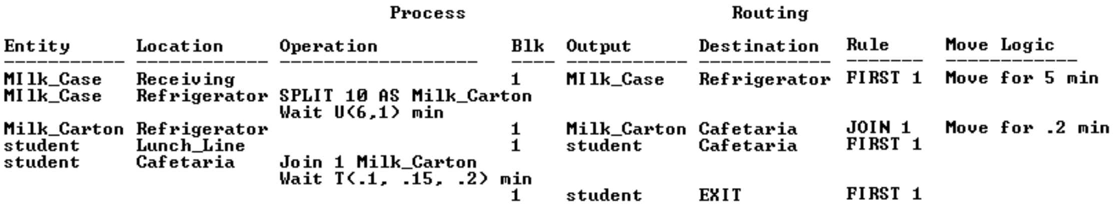
Arrival of milk and students at the San Dimas High School cafeteria.



Entity...	Location...	Qty each...	First Time	Occurrences	Frequency
Milk_Case	Receiving	10	0	inf	480 min
student	Lunch_Line	100	0	inf	480 min

FIGURE L7.46

Process and routing logic at the San Dimas High School cafeteria.



```
                                  Process                          Routing

Entity      Location     Operation           Blk  Output      Destination  Rule     Move Logic
----------- ------------ ------------------- ---- ----------- ------------ -------  -----------
MIlk_Case   Receiving                        1    MIlk_Case   Refrigerator FIRST 1  Move for 5 min
MIlk_Case   Refrigerator SPLIT 10 AS Milk_Carton
                         Wait U(6,1) min
Milk_Carton Refrigerator                     1    Milk_Carton Cafetaria    JOIN 1   Move for .2 min
student     Lunch_Line                       1    student     Cafetaria    FIRST 1
student     Cafetaria    Join 1 Milk_Carton
                         Wait T(.1, .15, .2) min
                                             1    student     EXIT         FIRST 1
```

L7.10 Decision Statements

In ProModel you can make use of several general control statements for decision making. ProModel meets the modern standards of structured program design. Although there is no formal definition, computer scientists agree that a structured program should have modular design and use only the following three types of logical structures: sequences, decisions, and loops.

- *Sequences:* Program statements are executed one after another.
- *Decisions:* One of two blocks of program code is executed based on a test for some condition.
- *Loops:* One or more statements are executed repeatedly as long as a specified condition is true.

All the control statements available in ProModel are shown in Figure L7.47. In this section we introduce you to some of these statements with examples.

L7.10.1 IF-THEN-ELSE Statement

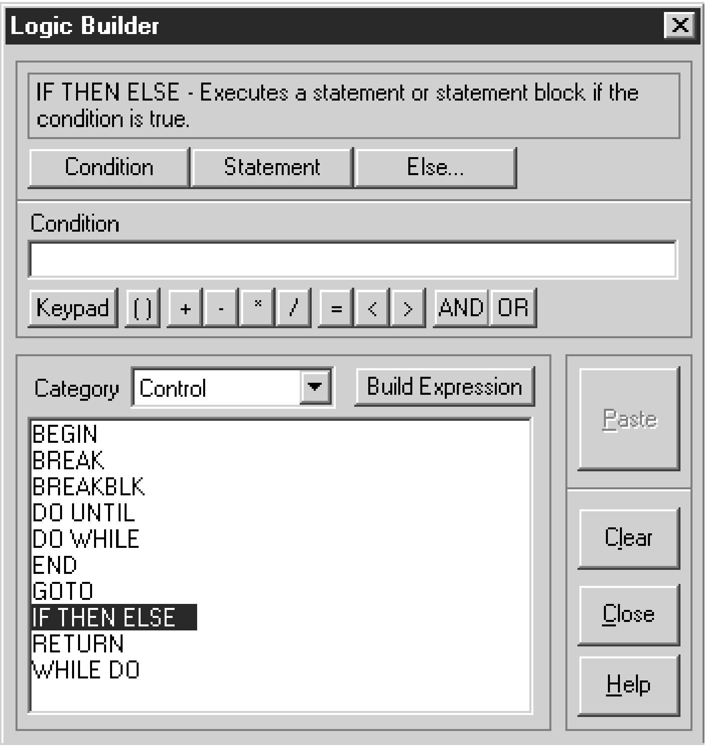
An IF block allows a program to decide on a course of action based on whether a certain condition is true or false. A program block of the form

```
IF condition THEN
     action1
ELSE
     action2
```

causes the program to take action1 if condition is true and action2 if condition is false. Each action consists of one or more ProModel statements. After an action is taken, execution continues with the line after the IF block.

Problem Statement

The **Bombay Restaurant** offers only a drive-in facility. Customers arrive at the rate of six each hour (exponential interarrival time). They place their orders at the first window, drive up to the next window for payment, pick up food from the last window, and then leave. The activity times are given in Table L7.4. The drive-in facility can accommodate 10 cars at most. However, customers typically leave



Logic Builder

IF THEN ELSE - Executes a statement or statement block if the condition is true.

Category: Control

FIGURE L7.47

Control statements available in ProModel.

TABLE L7.4 Process Times at the Bombay Restaurant

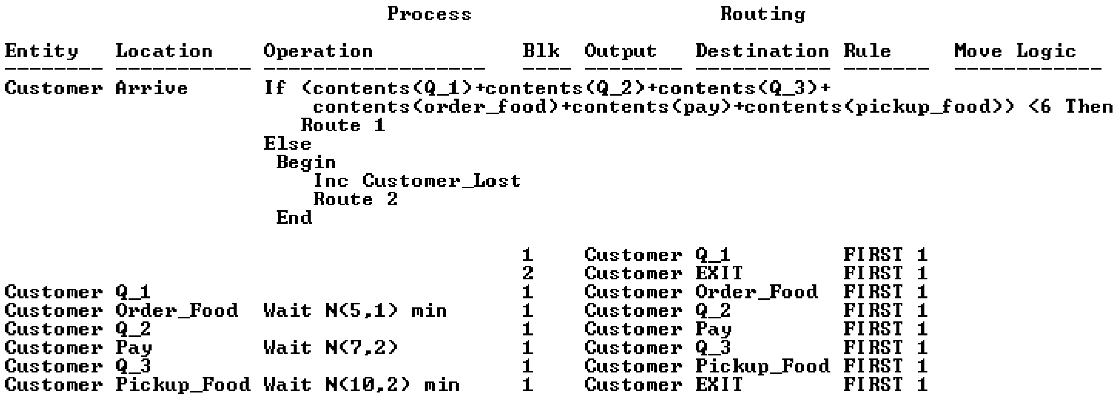
Activity	*Process Time (minutes)*
Order food	Normal (5,1)
Make payment	Normal (7,2)
Pick up food	Normal (10,2)

and go to the Madras Café across the street if six cars are waiting in line when they arrive. Simulate for 100 days (8 hours each day). Estimate the number of customers served each day. Estimate on average how many customers are lost each day to the competition.

An additional location (Arrive) is added to the model. After the customers arrive, they check if there are fewer than six cars at the restaurant. If yes, they join the line and wait; if not, they leave and go across the street to the Madras Café. An `IF-THEN-ELSE` statement is added to the logic window in the processing table (Figure L7.48). A variable (Customer_Lost) is added to the model to keep track of the number of customers lost to the competition. The layout of the Bombay

FIGURE L7.48

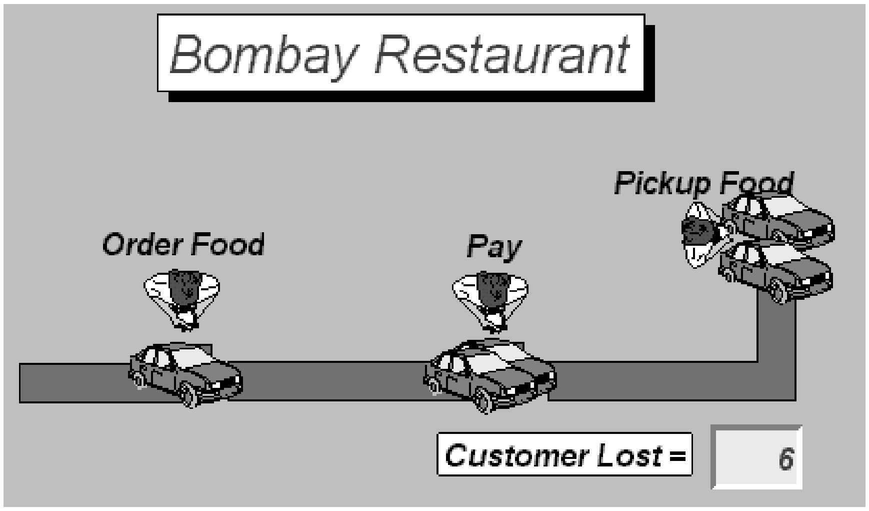
Process and routing logic at the Bombay Restaurant.



```
                                   Process                     Routing

Entity   Location    Operation           Blk  Output   Destination Rule     Move Logic
-------- ----------- ------------------  ---- -------- ----------- -------  -----------
Customer Arrive      If (contents(Q_1)+contents(Q_2)+contents(Q_3)+
                         contents(order_food)+contents(pay)+contents(pickup_food)) <6 Then
                        Route 1
                     Else
                      Begin
                         Inc Customer_Lost
                         Route 2
                      End

                                         1    Customer Q_1         FIRST 1
                                         2    Customer EXIT        FIRST 1
Customer Q_1                             1    Customer Order_Food  FIRST 1
Customer Order_Food  Wait N(5,1) min     1    Customer Q_2         FIRST 1
Customer Q_2                             1    Customer Pay         FIRST 1
Customer Pay         Wait N(7,2)         1    Customer Q_3         FIRST 1
Customer Q_3                             1    Customer Pickup_Food FIRST 1
Customer Pickup_Food Wait N(10,2) min    1    Customer EXIT        FIRST 1
```

FIGURE L7.49

Layout of the Bombay Restaurant.



Restaurant is shown in Figure L7.49. Note that Q_1 and Q_2 are each 100 feet long and Q_3 is 200 feet long.

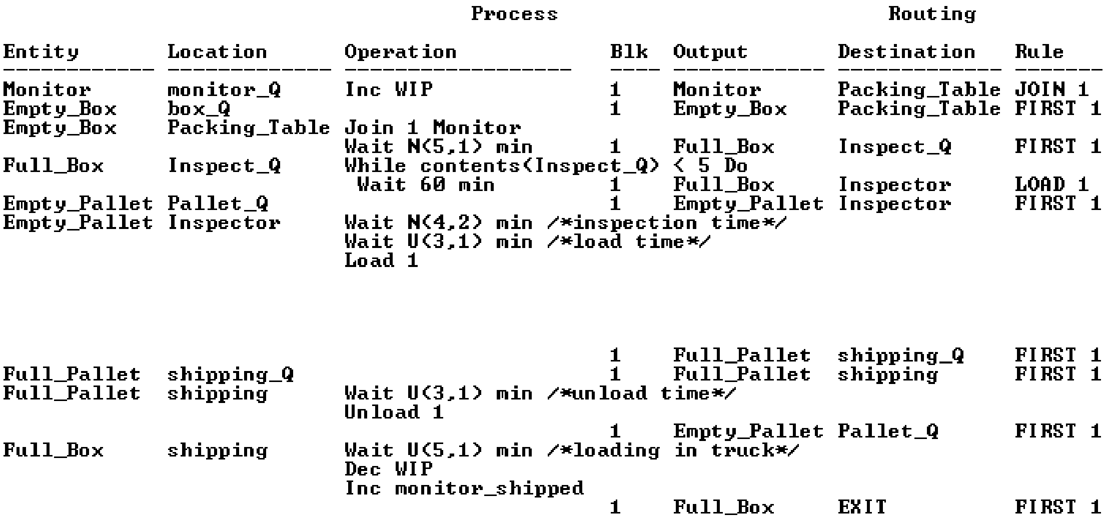
The total number of customers lost is 501 in 100 days. The number of customers served in 100 days is 4791. The average cycle time per customer is 36.6 minutes.

L7.10.2 WHILE-DO Loop

The WHILE-DO block repeats a group of statements continuously while a condition remains true. If the condition is false, the loop is bypassed.

FIGURE L7.50

An example of the WHILE-DO *logic for Shipping Boxes Unlimited.*



```
                                    Process                           Routing

Entity       Location      Operation            Blk  Output        Destination    Rule
------------ ------------- -------------------- ---- ------------- -------------- -------
Monitor      monitor_Q     Inc WIP              1    Monitor       Packing_Table  JOIN 1
Empty_Box    box_Q                              1    Empty_Box     Packing_Table  FIRST 1
Empty_Box    Packing_Table Join 1 Monitor
                           Wait N(5,1) min      1    Full_Box      Inspect_Q      FIRST 1
Full_Box     Inspect_Q     While contents(Inspect_Q) < 5 Do
                            Wait 60 min         1    Full_Box      Inspector      LOAD 1
Empty_Pallet Pallet_Q                           1    Empty_Pallet  Inspector      FIRST 1
Empty_Pallet Inspector     Wait N(4,2) min /*inspection time*/
                           Wait U(3,1) min /*load time*/
                           Load 1



                                                1    Full_Pallet   shipping_Q     FIRST 1
Full_Pallet  shipping_Q                         1    Full_Pallet   shipping       FIRST 1
Full_Pallet  shipping      Wait U(3,1) min /*unload time*/
                           Unload 1
                                                1    Empty_Pallet  Pallet_Q       FIRST 1
Full_Box     shipping      Wait U(5,1) min /*loading in truck*/
                           Dec WIP
                           Inc monitor_shipped
                                                1    Full_Box      EXIT           FIRST 1
```

Problem Statement

The inspector in Section L7.7.2 is also the supervisor of the shop. As such, she inspects only when at least five full boxes are waiting for inspection in the Inspect_Q. A WHILE-DO loop is used to check if the queue has five or more boxes waiting for inspection (Figure L7.50). The loop will be executed every hour. Figure L7.51 shows a time-weighted plot of the contents of the inspection queue. Note how the queue builds up to 5 (or more) before the inspector starts inspecting the full boxes.

L7.10.3 DO-WHILE Loop

The DO-WHILE block repeats a group of statements continuously while a condition remains true. This loop will be executed at least once; use it for processes that will be executed at least once and possibly more.

Problem Statement

For the **Poly Casting Inc.** example in Section L7.4, we would like to assign an upper limit to the work-in-process inventory in the shop. A DO-WHILE loop will be used to check if the level of WIP is equal to or above five; if so, the incoming castings will wait at the receiving dock (Figure L7.52). The loop will be executed once every hour to check if the level of WIP has fallen below five. Figure L7.53 show a time-weighted plot of the value of WIP at Poly Castings Inc. Note that the level of WIP is kept at or below five.

FIGURE L7.51

A plot of the contents of the inspection queue.

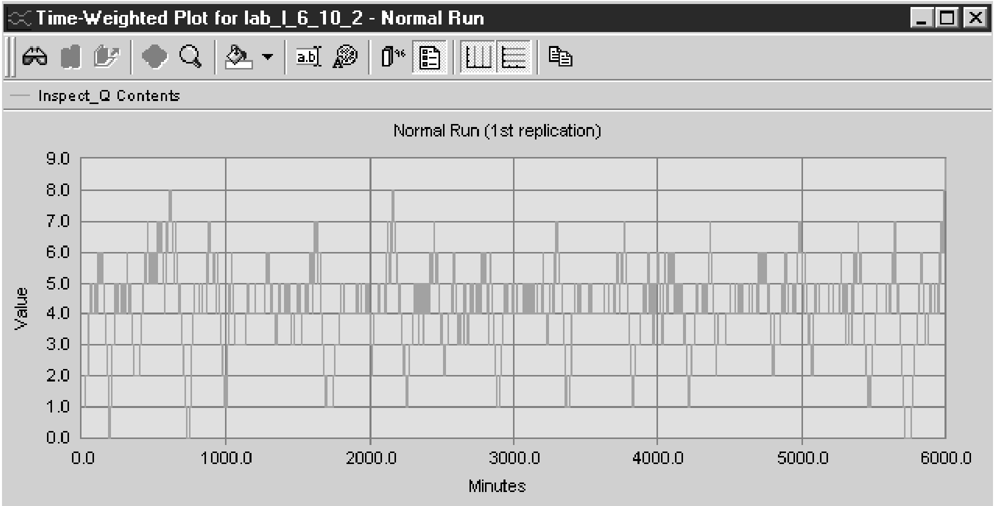
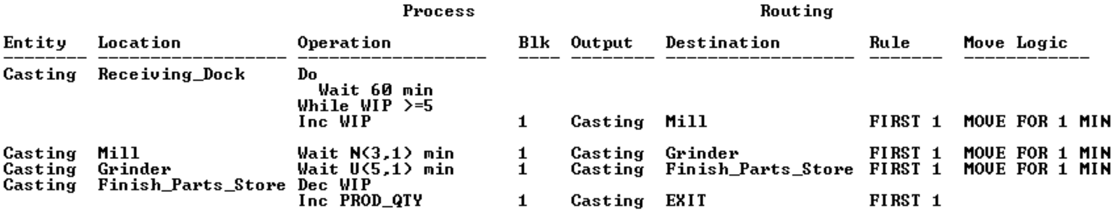


FIGURE L7.52

An example of a DO-WHILE loop.



```
                                    Process                                          Routing

Entity   Location            Operation           Blk  Output    Destination          Rule     Move Logic
-------- ------------------- ------------------- ---- --------- -------------------- -------- ------------
Casting  Receiving_Dock      Do
                               Wait 60 min
                             While WIP >=5
                             Inc WIP             1    Casting   Mill                 FIRST 1  MOVE FOR 1 MIN

Casting  Mill                Wait N(3,1) min     1    Casting   Grinder              FIRST 1  MOVE FOR 1 MIN
Casting  Grinder             Wait U(5,1) min     1    Casting   Finish_Parts_Store   FIRST 1  MOVE FOR 1 MIN
Casting  Finish_Parts_Store  Dec WIP
                             Inc PROD_QTY        1    Casting   EXIT                 FIRST 1
```

L7.10.4 GOTO Statement

The GOTO block jumps to the statement identified by the designated label. A label should follow the normal rules for names. A colon follows a label.

Problem Statement

The **Indian Bank** has acquired the Bank of India in Section L7.3. Anthony has been given a golden handshake (that is, fired). As a result of new training provided to the two remaining tellers (Amar and Akbar) there has been a change in their customer service time. Whenever more than three customers are waiting for service, the service time is reduced to an average of five minutes and a standard deviation of two minutes (normally distributed). The layout of the new Indian Bank is shown in Figure L7.54. A GOTO statement is used to jump to the appropriate wait statement depending on the number of customers waiting for service in the Teller_Q, as shown in Figure L7.55.

FIGURE L7.53

A plot of the value of WIP at Poly Castings Inc.

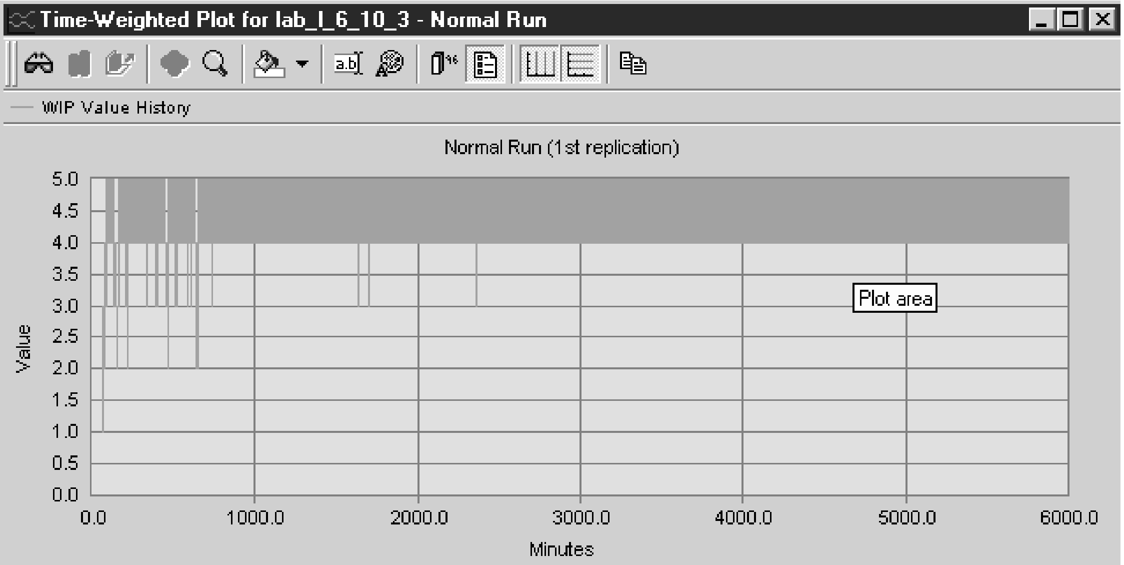


FIGURE L7.54

The layout of the Indian Bank.

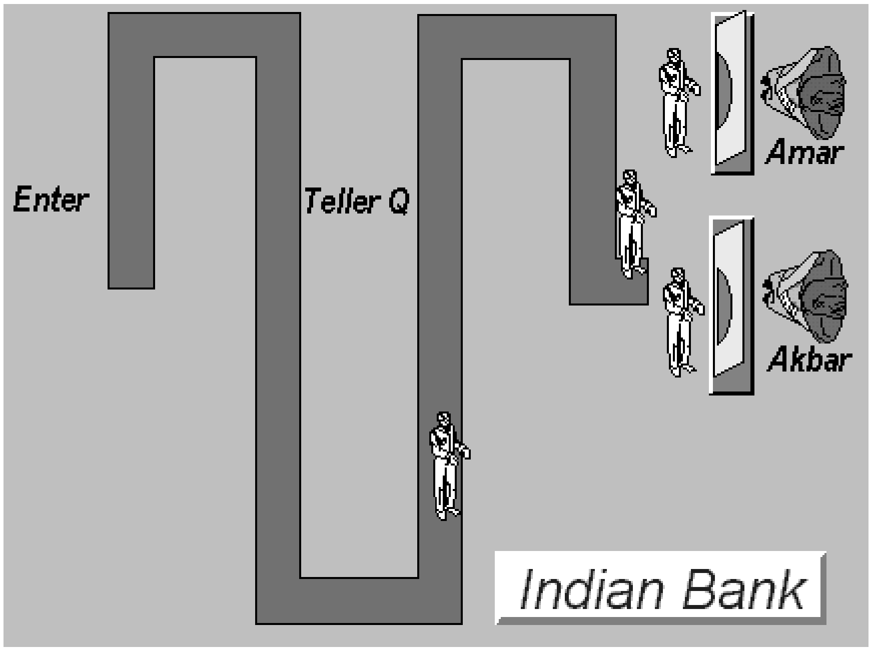
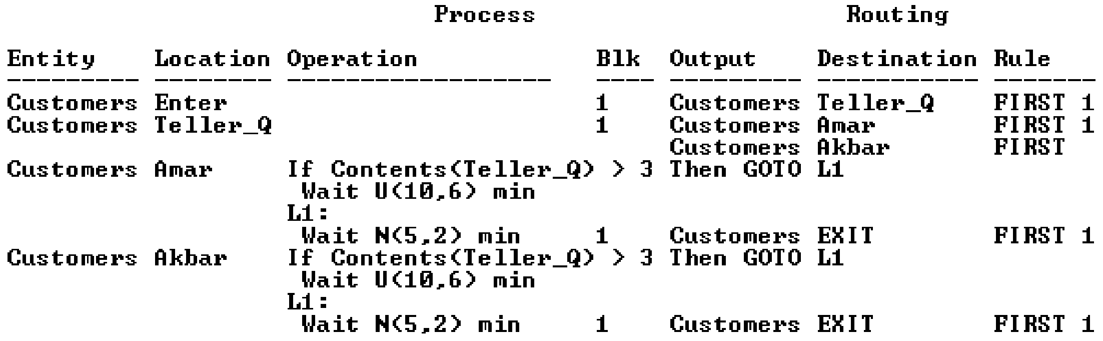


FIGURE L7.55

An example of a GOTO *statement.*



```
                              Process                        Routing

Entity    Location Operation           Blk  Output    Destination Rule
--------- -------- ------------------  ---- --------- ----------- -------
Customers Enter                        1    Customers Teller_Q    FIRST 1
Customers Teller_Q                     1    Customers Amar        FIRST 1
                                            Customers Akbar       FIRST
Customers Amar     If Contents(Teller_Q) > 3 Then GOTO L1
                    Wait U(10,6) min
                   L1:
                    Wait N(5,2) min    1    Customers EXIT        FIRST 1
Customers Akbar    If Contents(Teller_Q) > 3 Then GOTO L1
                    Wait U(10,6) min
                   L1:
                    Wait N(5,2) min    1    Customers EXIT        FIRST 1
```

L7.11 Periodic System Shutdown

Some service systems periodically stop admitting new customers for a given duration to catch up on the backlog of work and reduce the congestion. Some automobile service centers, medical clinics, restaurants, amusement parks, and banks use this strategy. The modulus mathematical operator in ProModel is useful for simulating such periodic shutdowns.

Problem Statement

The **Bank of India** in Section L7.3 opens for business each day for 8 hours (480 minutes). Assume that the customers arrive to the bank according to an exponential distribution (mean of 4 minutes). All the tellers service the customers according to a uniform distribution (mean of 10 minutes and half-width of 6 minutes). Customers are routed to the three tellers in turn (selected in rotation).

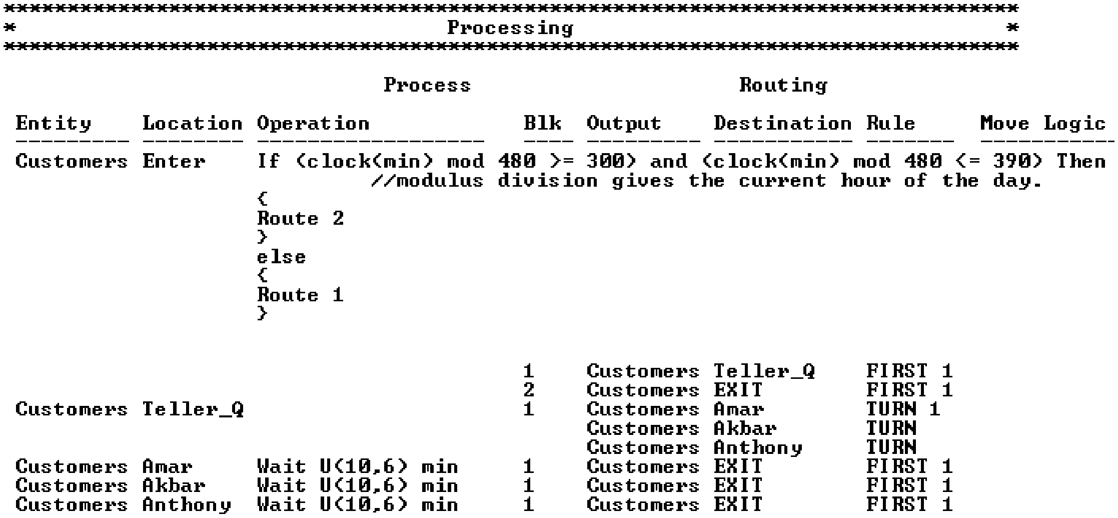
Each day after 300 minutes (5 hours) of operation, the front door of the bank is locked, and any new customers arriving at the bank are turned away. Customers already inside the bank continue to get served. The bank reopens the front door 90 minutes (1.5 hours) later to new customers. Simulate the system for 480 minutes (8 hours). Make a time-series plot of the Teller_Q to show the effect of locking the front door on the bank.

The logic for locking the front door is shown in Figure L7.56. The simulation clock time clock(min) is a cumulative counter of hours elapsed since the start of the simulation run. The current time of any given day can be determined by modulus dividing the current simulation time, clock(min), by 480 minutes. If the remainder of clock(min) divided by 480 minutes is between 300 minutes (5 hours) and 390 minutes (6.5 hours), the arriving customers are turned away (disposed). Otherwise, they are allowed into the bank. An IF-THEN-ELSE logic block as described in Section L7.10.1 is used here.

A time-series plot of the contents of the Teller_Q is shown in Figure L7.57. This plot clearly shows how the Teller_Q (and the whole bank) builds up during

FIGURE L7.56

Process and routing logic for the Bank of India.



```
********************************************************************************
*                                 Processing                                   *
********************************************************************************

                                 Process                       Routing

Entity    Location Operation                 Blk  Output    Destination Rule     Move Logic
--------- -------- ------------------        ---- --------- ----------- -------  ----------
Customers Enter    If (clock(min) mod 480 >= 300) and (clock(min) mod 480 <= 390) Then
                            //modulus division gives the current hour of the day.
                   {
                   Route 2
                   }
                   else
                   {
                   Route 1
                   }

                                             1    Customers Teller_Q    FIRST 1
                                             2    Customers EXIT        FIRST 1
Customers Teller_Q                           1    Customers Amar        TURN 1
                                                  Customers Akbar       TURN
                                                  Customers Anthony     TURN
Customers Amar     Wait U(10,6) min          1    Customers EXIT        FIRST 1
Customers Akbar    Wait U(10,6) min          1    Customers EXIT        FIRST 1
Customers Anthony  Wait U(10,6) min          1    Customers EXIT        FIRST 1
```

FIGURE L7.57

Time-series plot of Teller_Q at the Bank of India.

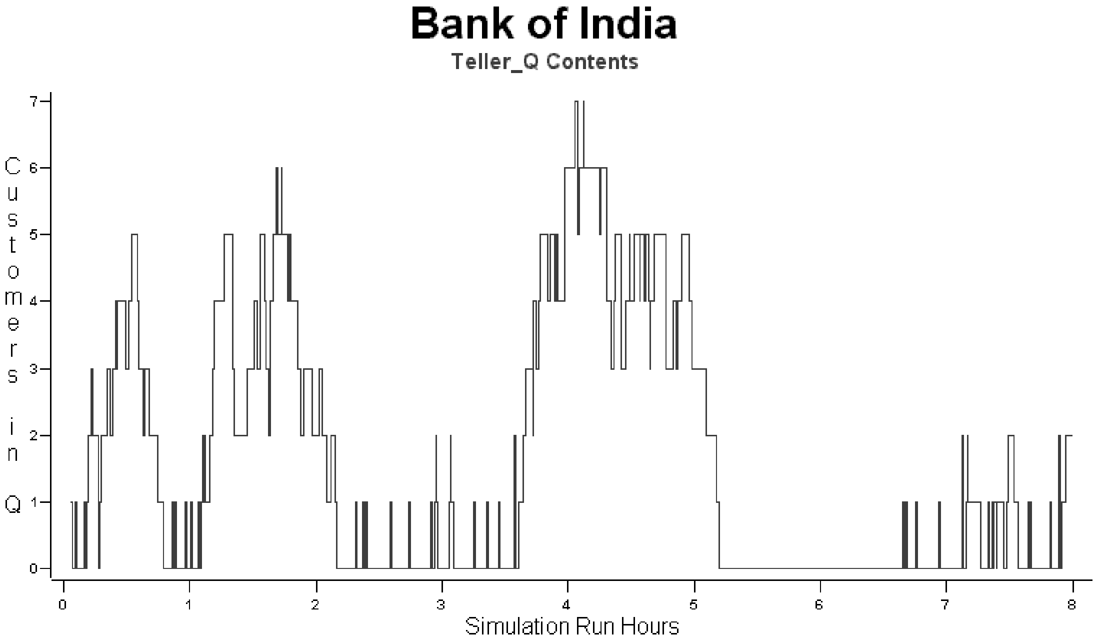
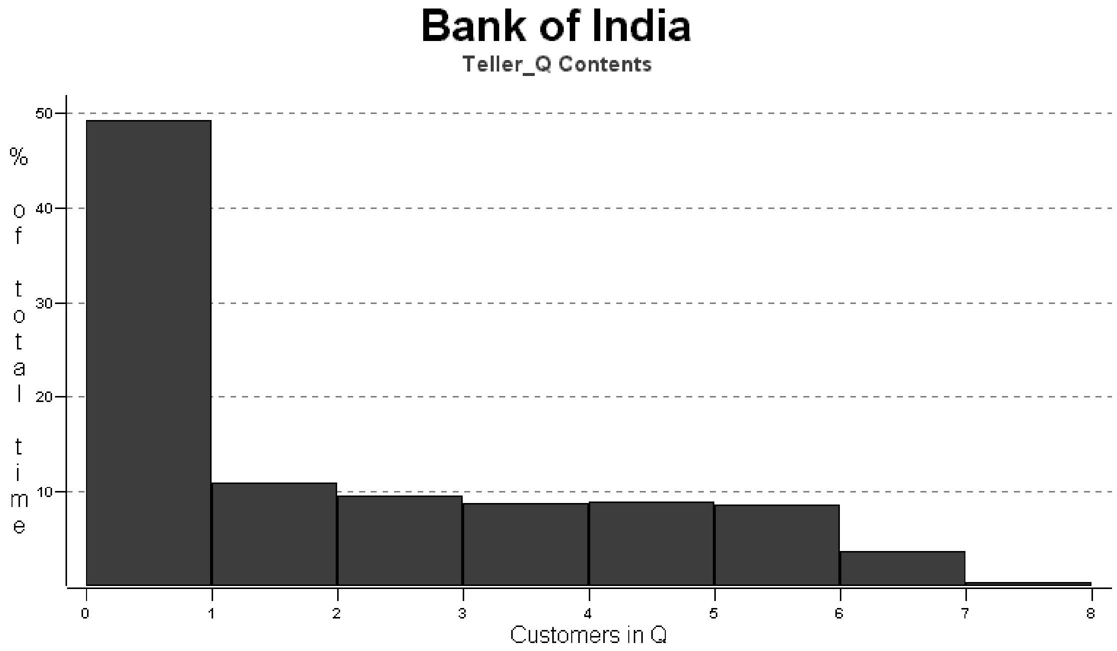


FIGURE L7.58

Histogram of Teller_Q contents at the Bank of India.



the day; then, after the front door is locked at the fifth hour (300 minutes) into the simulated day, customers remaining in the queue are processed and the queue length decreases (down to zero in this particular simulation run). The queue length picks back up when the bank reopens the front door at simulation time 6.5 hours (390 minutes).

The histogram of the same queue (Figure L7.58) shows that approximately 49% of the time the queue was empty. About 70% of the time there are 3 or fewer customers waiting in line. What is the average time a customer spends in the bank? Would you recommend that the bank not close the door after 5 hours of operation (customers never liked this practice anyway)? Will the average customer stay longer in the bank?

L7.12 Exercises

1. Visitors arrive at **Kid's World** entertainment park according to an exponential interarrival time distribution with mean 2.5 minutes. The travel time from the entrance to the ticket window is normally distributed with a mean of three minutes and a standard deviation of 0.5 minute. At the ticket window, visitors wait in a single line until one

of four cashiers is available to serve them. The time for the purchase of tickets is normally distributed with a mean of five minutes and a standard deviation of one minute. After purchasing tickets, the visitors go to their respective gates to enter the park. Create a simulation model, with animation, of this system. Run the simulation model for 200 hours to determine

a. The average and maximum length of the ticketing queue.
b. The average number of customers completing ticketing per hour.
c. The average utilization of the cashiers.
d. Whether management should add more cashiers.

2. A consultant for **Kid's World** recommended that four individual queues be formed at the ticket window (one for each cashier) instead of one common queue. Create a simulation model, with animation, of this system. Run the simulation model for 200 hours to determine
   a. The average and maximum length of the ticketing queues.
   b. The average number of customers completing ticketing per hour.
   c. The average utilization of the cashiers.
   d. Whether you agree with the consultant's decision. Would you recommend a raise for the consultant?
3. At the **Kid's World** entertainment park in Exercise 1, the operating hours are 8 A.M. till 10 P.M. each day (all week). Simulate for a whole year (365 days) and answer questions *a–d* as given in Exercise 1.
4. At **Southern California Airline**'s traveler check-in facility, three types of customers arrive: passengers with e-tickets (Type E), passengers with paper tickets (Type T), and passengers that need to purchase tickets (Type P). The interarrival distribution and the service times for these passengers are given in Table L7.5. Create a simulation model, with animation, of this system. Run the simulation model for 2000 hours. If separate gate agents serve each type of passenger, determine the following:
   a. The average and maximum length of the three queues.
   b. The average number of customers of each type completing check-in procedures per hour.
   c. The average utilization of the gate agents.

TABLE L7.5 Interarrival and Service Time Distributions at Southern California Airline

Type of Traveler	*Interarrival Distribution*	*Service Time Distribution*
Type E	Exponential (mean 5.5 min.)	Normal (mean 3 min., std. dev. 1 min.)
Type T	Exponential (mean 10.5 min.)	Normal (mean 8 min., std. dev. 3 min.)
Type P	Exponential (mean 15.5 min.)	Normal (mean 12 min., std. dev. 3 min.)

d. The percentage of time the number of customers (of each type of customer) waiting in line is ≤ 2.

e. Would you recommend one single line for check-in for all three types of travelers? Discuss the pros and cons of such a change.

5. **Raja & Rani,** a fancy restaurant in Santa Clara, holds a maximum of 15 diners. Customers arrive according to an exponential distribution with a mean of 5 minutes. Customers stay in the restaurant according to a triangular distribution with a minimum of 45 minutes, a maximum of 60 minutes, and a mode of 75 minutes. Create a simulation model, with animation, of this system. The restaurant operating hours are 3 P.M. till 12 P.M. Run the simulation model for 50 replications of 9 hours each.

   a. Beginning empty, how long (average and standard deviation) does it take for the restaurant to fill?

   b. What is the total number of diners (average and standard deviation) entering the restaurant before it fills?

   c. What is the total number of guests (average and standard deviation) served per night?

   d. What is the average utilization of the restaurant?

6. **Woodland Nursing Home** has six departments—initial exam, X ray, operating room, cast-fitting room, recovery room, and checkout room. The probabilities that a patient will go from one department to another are given in Table L7.6. The time patients spend in each department is given in Table L7.7. Patients arrive at the average rate of 5 per hour (exponential interarrival time). The nursing home remains open 24/7.

TABLE L7.6 Probabilities of Patient Flow

From	*To*	*Probability*
Initial exam	X ray	.35
	Operating room	.20
	Recovery room	.15
	Checkout room	.30
X ray	Operating room	.15
	Cast-fitting room	.25
	Recovery room	.40
	Checkout room	.20
Operating room	Cast-fitting room	.30
	Recovery room	.65
	Checkout room	.05
Cast-fitting room	Recovery room	.55
	X ray	.10
	Checkout room	.35
Recovery room	Operating room	.10
	X ray	.20
	Checkout room	.70

TABLE L7.7 Patient Processing Time

Department	*Patient Time in Department*
Initial exam	Normal (5,2) minutes
X ray	Uniform (6,1) minutes
Operating room	Normal (30,5) minutes
Cast-fitting room	Uniform (10,4) minutes
Recovery room	Normal (8,2) minutes

TABLE L7.8 Process Time Distributions at United Electronics

Assembly	Uniform (13.5±1.5) minutes
Soldering	Normal (36,10) minutes
Painting	Uniform (55±15) minutes
Inspection	Exponential (8) minutes

However, every seven hours (420 minutes) the front door is locked for an hour (60 minutes). No new patients are allowed in the nursing home during this time. Patients already in the system continue to get served. Simulate for one year (365 days, 24 hours per day).

a. Figure out the utilization of each department.

b. What are the average and maximum numbers of patients in each department?

c. Which is the bottleneck department?

d. What is the average time spent by a patient in the nursing home?

7. **United Electronics** manufactures small custom electronic assemblies. Parts must be processed through four stations: assembly, soldering, painting, and inspection. Orders arrive with an exponential interarrival distribution (mean 20 minutes). The process time distributions are shown in Table L7.8.

   The soldering operation can be performed on three jobs at a time. Painting can be done on four jobs at a time. Assembly and inspection are performed on one job at a time. Create a simulation model, with animation, of this system. Simulate this manufacturing system for 100 days, eight hours each day. Collect and print statistics on the utilization of each station, associated queues, and the total number of jobs manufactured during each eight-hour shift (average).

8. In **United Electronics** in Exercise 7, 10 percent of all finished assemblies are sent back to soldering for rework after inspection, five percent are sent back to assembly for rework after inspection, and one

percent of all assemblies fail to pass and are scrapped. Create a simulation model, with animation, of this system. Simulate this manufacturing system for 100 days, eight hours each day. Collect and print statistics on the utilization of each station, associated queues, total number of jobs assembled, number of assemblies sent for rework to assembly and soldering, and the number of assemblies scrapped during each eight-hour shift (average).

9. Small toys are assembled in four stages (Centers 1, 2, and 3 and Inspection) at **Bengal Toy Company.** After each assembly step, the appliance is inspected or tested; if a defect is found, it must be corrected and then checked again. The assemblies arrive at a constant rate of one assembly every two minutes. The times to assemble, test, and correct defects are normally distributed. The means and standard deviations of the times to assemble, inspect, and correct defects, as well as the likelihood of an assembly error, are shown in Table L7.9. If an assembly is found defective, the defect is corrected and it is inspected again. After a defect is corrected, the likelihood of another defect being found is the same as during the first inspection. We assume in this model that an assembly defect is eventually corrected and then it is passed on to the next station. Simulate for one year (2000 hours) and determine the number of good toys shipped in a year.
10. **Salt Lake City Electronics** manufactures small custom communication equipment. Two different job types are to be processed within the following manufacturing cell. The necessary data are given in Table L7.10. Simulate the system to determine the average number of jobs waiting for different operations, number of jobs of each type finished each day, average cycle time for each type of job, and the average cycle time for all jobs.
11. Six dump trucks at the **DumpOnMe** facility in Riverside are used to haul coal from the entrance of a small mine to the railroad. Figure L7.59 provides a schematic of the dump truck operation. Each truck is loaded by one of two loaders. After loading, a truck immediately moves

TABLE L7.9 Process Times and Probability of Defects at Bengal Toy Company

	Assembly Time		Inspect Time		Correct Time		
Center	*Mean*	*Standard Deviation*	*Mean*	*Standard Deviation*	*P(error)*	*Mean*	*Standard Deviation*
1	.7	.2	.2	.05	.1	.2	.05
2	.75	.25	.2	.05	.05	.15	.04
3	.8	.15	.15	.03	.03	.1	.02

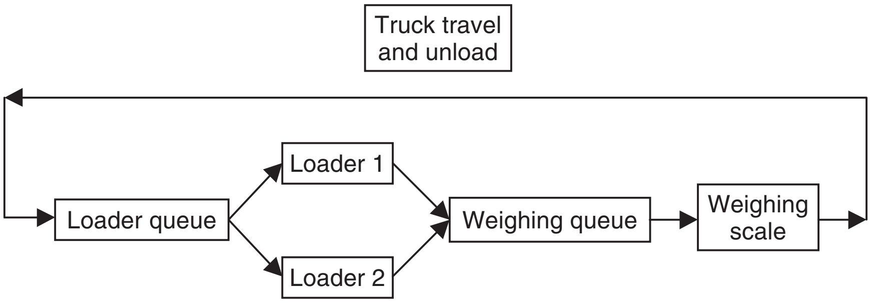


FIGURE L7.59
Schematic of dump truck operation for DumpOnMe.

TABLE L7.10 Data Collected at Salt Lake City Electronics

Job Type	*Number of Batches*	*Number of Jobs per Batch*	*Assembly Time*	*Soldering Time*	*Painting Time*	*Inspection Time*	*Time between Batch Arrivals*
1	15	5	Tria (5,7,10)	Normal (36,10)	Uniform (55±15)	Exponential (8)	Exp (14)
2	25	3	Tria (7,10,15)		Uniform (35±5)	Exponential (5)	Exp (10)

Note: All times are given in minutes.

TABLE L7.11 Various Time Measurement Data at the DumpOnMe Facility

Loading time	Uniform (7.5±2.5) minutes
Weighing time	Uniform (3.5±1.5) minutes
Travel time	Triangular (10,12,15) minutes

to the scale to be weighed as soon as possible. Both the loaders and the scale have a first-come, first-served waiting line (or queue) for trucks. Travel time from a loader to the scale is considered negligible. After being weighed, a truck begins travel time (during which time the truck unloads), and then afterward returns to the loader queue. The distributions of loading time, weighing time, and travel time are shown in Table L7.11.

a. Create a simulation model, with animation, of this system. Simulate for 200 days, eight hours each day.
b. Collect statistics to estimate the loader and scale utilization (percentage of time busy).
c. About how many trucks are loaded each day on average?

12. At the **Pilot Pen Company,** a molding machine produces pen barrels of two different colors—red and blue—in the ratio of 3:2. The molding time is triangular (3,4,6) minutes per barrel. The barrels go to a filling machine, where ink of appropriate color is filled at the rate of 20 pens per hour (exponentially distributed). Another molding machine makes caps of the same two colors in the ratio of 3:2. The molding time is triangular (3,4,6) minutes per cap. At the next station, caps and filled barrels of matching colors are joined together. The joining time is exponentially distributed with a mean of 1 min. Simulate for 2000 hours. Find the average number of pens produced per hour. Collect statistics on the utilization of the molding machines and the joining equipment.
13. Customers arrive at the **NoWaitBurger** hamburger stand with an interarrival time that is exponentially distributed with a mean of one minute. Out of 10 customers, 5 buy a hamburger and a drink, 3 buy a hamburger, and 2 buy just a drink. One server handles the hamburger while another handles the drink. A person buying both items needs to wait in line for both servers. The time it takes to serve a customer is N(70,10) seconds for each item. Simulate for 100 hours. Collect statistics on the number of customers served per hour, size of the queues, and utilization of the servers. What changes would you suggest to make the system more efficient?
14. Workers who work at the **Detroit ToolNDie** plant must check out tools from a tool crib. Workers arrive according to an exponential distribution with a mean time between arrivals of five minutes. At present, three tool crib clerks staff the tool crib. The time to serve a worker is normally distributed with a mean of 10 minutes and a standard deviation of 2 minutes. Compare the following servicing methods. Simulate for 2000 hours and collect data.
    a. Workers form a single queue, choosing the next available tool crib clerk.
    b. Workers enter the shortest queue (each clerk has his or her own queue).
    c. Workers choose one of three queues at random.
15. At the **ShopNSave,** a small family-owned grocery store, there are only four aisles: aisle 1—fruits/vegetables, aisle 2—packaged goods (cereals and the like), aisle 3—dairy products, and aisle 4—meat/fish. The time between two successive customer arrivals is exponentially distributed with a mean of 5 minutes. After arriving to the store, each customer grabs a shopping cart. Twenty percent of all customers go to aisle 1, 30 percent go to aisle 2, 50 percent go to aisle 3, and 70 percent go to aisle 4. The number of items selected for purchase in each aisle is uniformly distributed between 2 and 8. The time spent to browse and pick up each item is normally distributed: N(5,2) minutes. There are three identical checkout counters; each counter has its own checkout

line. The customer chooses the shortest line. Once a customer joins a line, he or she is not allowed to leave or switch lines. The checkout time is given by the following regression equation:

$$\text{Checkout time} = \text{N}(3,0.3) + (\#\text{of items}) * \text{N}(0.5,0.15) \text{ minutes}$$

The first term of the checkout time is for receiving cash or a check or credit card from the customer, opening and closing the cash register, and handing over the receipt and cash to the customer. After checking out, a customer leaves the cart at the front of the store and leaves. Build a simulation model for the grocery store. Use the model to simulate a 14-hour day.

a. The percentages of customers visiting each aisle do not add up to 100 percent. Why?
b. What is the average amount of time a customer spends at the grocery store?
c. How many customers check out per cashier per hour?
d. What is the average amount of time a customer spends waiting in the checkout line?
e. What is the average utilization of the cashiers?
f. Assuming there is no limit to the number of shopping carts, determine the average and maximum number of carts in use at any time.
g. On average how many customers are waiting in line to checkout?
h. If the owner adopts a customer service policy that there will never be any more than three customers in any checkout line, how many cashiers are needed?

Embellishments:

I. The store manager is considering designating one of the checkout lines as Express, for customers checking out with 10 or fewer items. Is that a good idea? Why or why not?

II. In reality, there are only 10 shopping carts at the ShopNSave store. If there are no carts available when customers arrive they leave immediately and go to the more expensive ShopNSpend store down the street. Modify the simulation model to reflect this change. How many customers are lost per hour? How many shopping carts should they have so that no more than 5 percent of customers are lost?

16. Planes arrive at the **Netaji Subhash Chandra Bose International Airport, Calcutta** with interarrival times that are exponentially distributed with a mean time of 30 minutes. If there is no room at the airport when a plane arrives, the pilot flies around and comes back to land after a normally distributed time having a mean of 20 minutes and a standard deviation of 5 minutes. There are two runways and three gates at this small airport. The time from touchdown to arrival at a gate is normally distributed having a mean of five minutes and a standard

deviation of one minute. A maximum of three planes can be unloaded and loaded at the airport at any time. The times to unload and load a plane are uniformly distributed between 20 and 30 minutes. The pushoff, taxi, and takeoff times are normally distributed with a mean of six minutes and a standard deviation of one minute. The airport operates 24/7. Build a simulation model and run it for one year.

a. How much will adding a new gate decrease the time that airplanes have to circle before landing?

b. Will adding a new gate affect the turnaround time of airplanes at the airport? How much?

c. Is it better to add another runway instead of another gate? If so, why?