7 Case Study: DePorres Tours

Chapter 7 focuses on taking a simulation project from start to finish. Although the system might seem simplistic and could probably be adequately modeled using a spreadsheet, the intent is to demonstrate the thought process used in defining, analyzing, and applying modeling principles. The system to be analyzed provides a good framework for discussion and will be interesting to a wide variety of simulation (or potential simulation) analysts. The project will follow the simulation life cycle illustrated in Chapter 4 (See Figure 7.1).

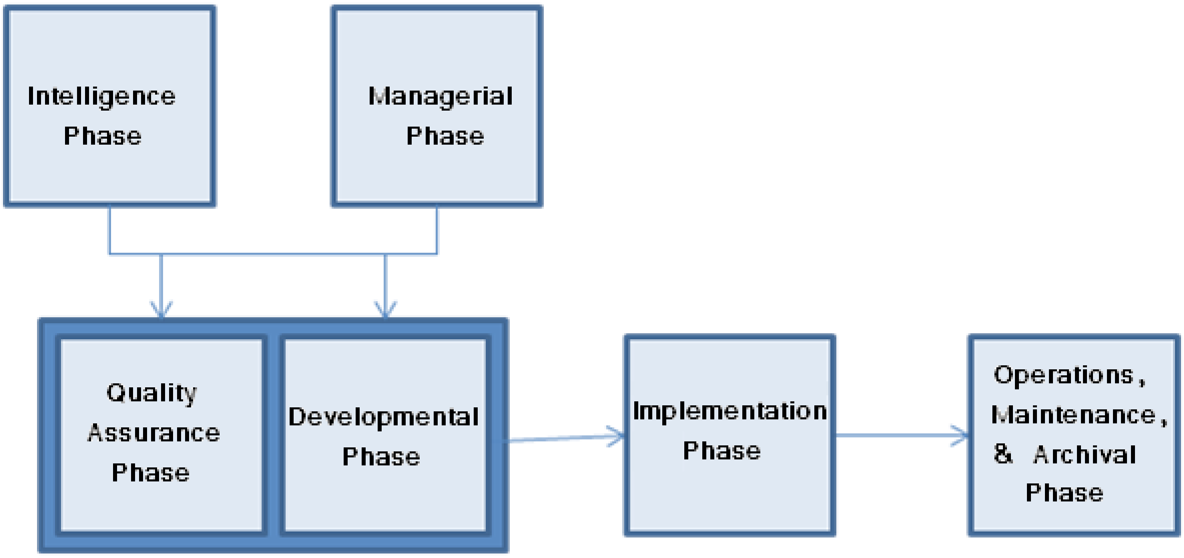


Figure 7.1 Phases in the Simulation Life Cycle

7.1 Intelligence Phase

The intelligence phase of the simulation life cycle involves understanding the environment and determining problems to be solved. This often starts with a general understanding of the system to be modeled together with a problem definition and feasibility considerations.

> DePorres Tours: General Scenario
>
> DePorres Tours provides a small minibus shuttle system in downtown Chicago. In general, they sell customers a day pass that allows access to a minibus as it shuttles between five major stops in the city. The pass is unlimited for the day, enabling tourists to disembark, enjoy visiting a Chicago landmark at their leisure, then re-enter the bus and shuttle to another location. The minibus stops at 5 locations: Stop 1: The new Millennium Park, Stop 2: Lincoln Park Zoo and Gardens, Stop 3: The Sears and Hancock Towers, Stop 4: The Magnificent Mile, Stop 5: Navy Pier.

7.1.1 Problem Definition

DePorres Tours wants to create a simulation to help determine bus capacity for their tour operation and then be able to utilize the model in the future to pre-test any changes to their operation. Their current problem statement becomes:

"What size bus will best accommodate expected customer traffic?"

Specifically, they are looking at buying either a 24 or a 48 passenger bus.

7.1.2 General Feasibility

Before beginning the simulation project, DePorres assessed general feasibility using the TELOS approach (See Figure 7.2):

Technical – A staff member took a simulation class and has the technical skills needed to develop a model. Current available simulation software is certainly capable of modeling a system such as the tour bus route.

Economic – A preliminary cost study indicated software and wages will not be excessive and will meet budget constraints.

Legal – No legal issues will result from developing and using this model.

Operational – The model will be routine and the developed system does not appear to have any operational issues. An accurate model can be developed.

Schedule – Model development is not under a strict timeline and should meet the required purchase date for the new bus.

Figure 7.2 Feasibility Questions

7.2 Managerial Phase

DePorres Tours recently sent Telly O'Sullivan to a simulation training course and has allocated 4 hours a day for him to gather information and conduct the simulation project. Since it is his first project, he has been given a month to complete the model. DePorres Tours owner, the indomitable Kafy DePorres has informed all drivers, tour guides, and office personnel about the project and strongly encouraged their full cooperation. Additionally, a small team of one driver, one tour guide, and Kafy herself will meet with Telly weekly to assess his progress and help with any problems. Kafy has set aside adequate funds for the project and plans to acquire a simulation software package for model development.

7.3 Developmental Phase

Telly, anxious to apply what he learned in class, developed a working view of the system to be modeled. He used the following definition to help break the tour bus system into its major components.

System: A set of components or elements that are related to each other in such a manner as to create a connected whole.

He saw the system as having these related parts (Table 7.1):

Bus
Bus Driver
Tourist
Bus Route
Bus Stops

Table 7.1 Related Subsystems

He also defined relationships between elements (See Table 7.2)

Entity	Bus	Bus Driver	Tourist	Bus Route	Bus Stops
Bus		Bus operated by Driver	Bus ridden by Tourist	Bus follows Bus Route	Bus stops at Bus Stops
Bus Driver	Bus Driver operates Bus		Bus Driver checks Tourist ticket	Bus Driver guides Bus along Route	Bus Driver interacts with Tourists at Stops
Tourist	Tourist rides Bus	Tourist shows ticket to Driver		Tourist follows Bus Route while on Bus	Tourists embarks and disembarks at Bus Stops and interacts with Driver at Stops
Bus Route	Bus Route followed by Bus	Bus Route followed by Driver	Bus Route followed by Tourist		Bus Route incorporates Stops
Bus Stops	Bus Stops are used by Buses	Bus Driver halts Bus at Stops	Bus Stops are where tourists get on and off bus	Bus Stops are along the Bus Route	

Table 7.2 Relationships between Subsystem Elements

Although the table contains some repetitive information, it becomes a working document to help understand the system and its basic elements.

7.3.1 Environment and Boundary

Based on the data shown in Table 7.2, the system environment can be defined as the road system in Chicago, existing traffic patterns, and tourism patterns. All these things lie outside the system and have an influence on its behavior. The systems boundary separates the tour bus system, together with tourists, driver, route, and stops from the environment.

7.3.2 Model Scaling and Scope

Telly decided the scope and scale of his model would include the following:

Scope:

1) Customer arrivals and departures will be modeled at each bus stop.
2) Bus driver and bus will be modeled.
3) Bus travel times along the route will be modeled.

Scale:

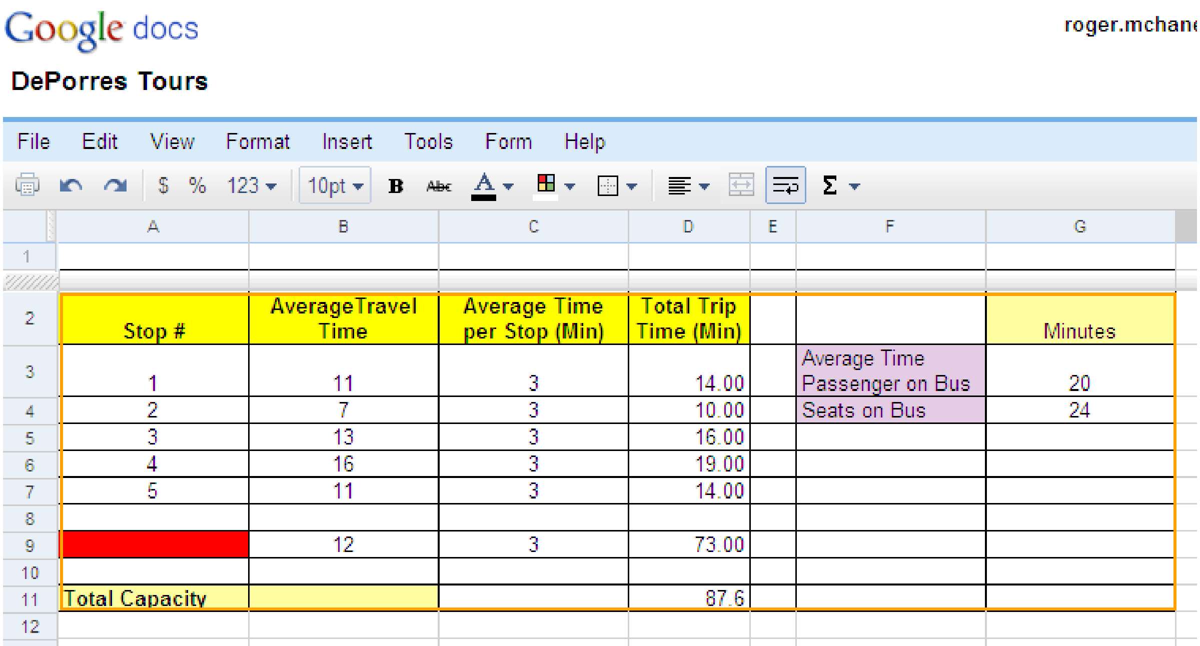
1) Two customer types will be considered: first time riders and those using the bus as a taxi.
2) Bus driver and bus will be modeled as a single resource. Bus driver behavior will be averaged into bus stop times. In the future this could be redeveloped using bus driver persims.
3) Bus travel times will be modeled using collected distributions covering an entire day.

7.3.3 Modeling Views

Telly decided to use a process orientation view of the system. In other words, he will view the bus system as a time ordered sequence of interrelated events separated by passages of time. This will influence his choice of simulation software but is consistent with the recent training he received.

7.3.4 Concept Model

Telly created a couple of concept models to help understand the system better. First he created a spreadsheet that calculated hourly capacity and average route time (Figure 7.2).


Google docs
roger.mchane
DePorres Tours

	A	B	C	D	E	F	G
1							
2	Stop #	AverageTravel Time	Average Time per Stop (Min)	Total Trip Time (Min)			Minutes
3	1	11	3	14.00		Average Time Passenger on Bus	20
4	2	7	3	10.00		Seats on Bus	24
5	3	13	3	16.00			
6	4	16	3	19.00			
7	5	11	3	14.00			
8							
9		12	3	73.00			
10							
11	Total Capacity			87.6			
12							

Figure 7.2 Concept Model with Google Docs

Next he used an Excel spreadsheet with Paul Jensen's ORMM Queuing Add-ins to create a rough model of the bus route if viewed as a queuing system (See Figure 7.3).

Queue Station	Bus_Seats
Arrival Rate	84
Service Rate/Channel	4
Number of Servers	24
Max. Number in System	***
Number in Population	***
Type	M/M/24
Mean Number at Station	23.97426
Mean Time at Station	0.285408
Mean Number in Queue	2.974259
Mean Time in Queue	0.035408
Mean Number in Service	21
Mean Time in Service	0.25
Throughput Rate	84
Efficiency	0.875

Figure 7.3 ORMM Queuing Add-in for MS-Excel

Both models created quick approximations of the system. However, neither sufficiently captured the dynamics of the system adequately. This left Telly with no option other than to move ahead with a full simulation. Both concept models demonstrated a 24 passenger bus could move around 80-84 people per hour but this was in contrast to what had been observed in the real world. Telly suspected the variance in the system had not been included in spreadsheets realistically.

7.3.5 Model Inputs

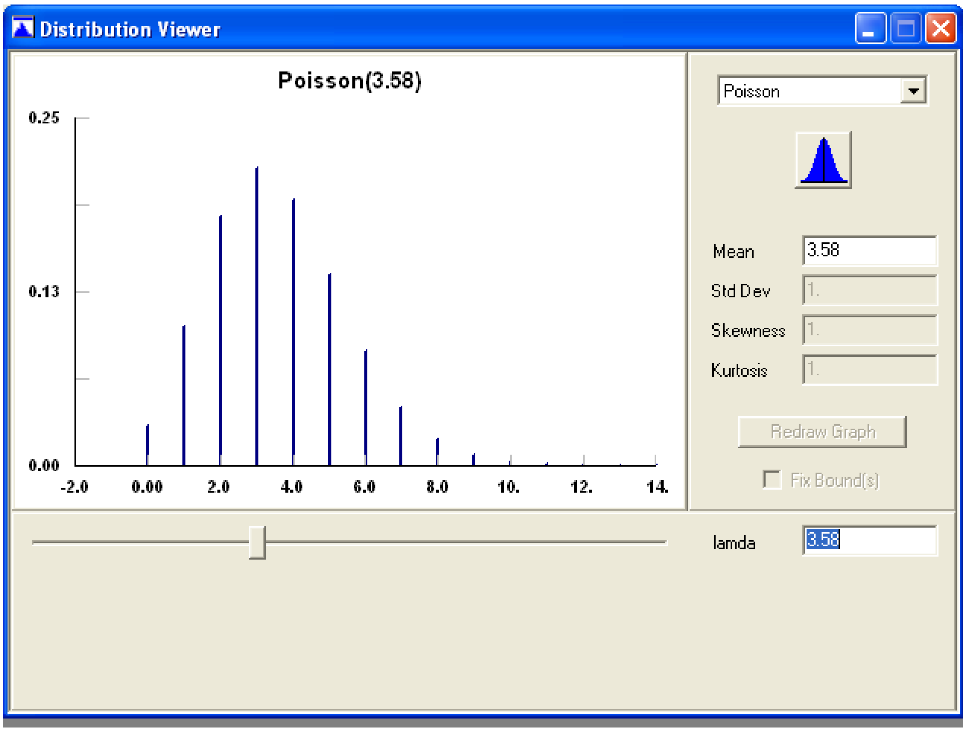
Telly's next task was to begin data collection in earnest. Although he had calculated quick averages he used in the concept models, he wanted to develop more accurate information. He did this by visiting each bus stop and collecting customer arrival times as they waited for the bus. Table 7.3 provides a look at the data he collected at Bus Stop #1. He loaded this into Stat::Fit 2 and determined the tourists were arriving according to a Poisson distribution with a mean of 3.58 minutes (See Figure 7.4).

1	3	5	4	5
5	1	3	4	4
5	2	2	5	1
3	4	4	6	5
3	3	1	4	2
3	3	6	4	4
1	9	3	7	1
4	3	6	1	1
5	9	5	3	1
2	6	2	0	5

Table 7.3 Tourist Interarrival Times in Minutes

Figure 7.4 Customer Interarrival Data Fit to a Poisson Distribution

Telly repeated his data collection for the other 4 stops and determined the following interarrival rates (see Table 7.4).



Stop #	Distribution	Mean	Spread
1	Poisson	3.58	n/a
2	Poisson	3.22	n/a
3	Uniform	4	2
4	Poisson	3.34	n/a
5	Poisson	3.12	n/a

Table 7.4 Tourist Interarrival Times

Telly continued to collect input data throughout the week and was able to determine the following quantitative information would need to be included in the model:

Average Time Bus Spends at a Stop: Normally distributed 2 minutes with a standard deviation of .3 minutes.

Time between Stops (see Table 7.5).

Time Between Stop #	Distribution	Mean / Spread (seconds)
1–2	Exponential	12,4
2–3	Exponential	8,3
3–4	Exponential	14,4
4–5	Exponential	8,3
5–1	Exponential	15,4

Table 7.5 Time per Bus Routes

- Passenger Route Decision Percentages (see Table 7.7).

Stop # Destination %	1	2	3	4	5
1	0	80	10	5	5
2	2	0	76	20	2
3	4	2	0	67	27
4	15	6	3	0	76
5	88	8	2	2	0

Table 7.7 Destination Percentages

- Qualitative Data. The following assumptions were also gathered:

1) The bus always goes in sequence 1-2-3-4-5 then back to 1.
2) The bus stops even if no passengers are visible at the stop or desire to be let off.
3) Bus drivers change shifts without disrupting the schedule.

7.3.6 Simulation Input Data Validation

As part of the Quality Phase of his simulation project, Telly validated his input data in the following ways:

Observation: He observed the current system and double checked his input data against what he observed.
Expert's Opinions: He showed the current drivers and ticket sellers his assumptions and input data for evaluation.

Intuition and Experience: He also used his own experience to double check the data.
Telly continued to check for validation throughout his model development process.

7.3.7 Selection of a Language or Tool

Telly was now ready to select a simulation tool for model development. He initially looked into both simulation languages and simulator packages. After doing a little research and since he had recently taken a GPSS simulation class, he decided to concentrate on general purpose simulation languages. Telly used a standard approach to the software evaluation process. The four steps in his procedure were:

1) Define needs of project
2) Research market
3) Compare available options
4) Select software

What will the frequency of future simulation work be? Will this be a onetime project, or will simulation be used regularly from now on? The bus simulation will be a onetime project. It will be maintained so that it might be modified and used for future analysis.

Step one: Define needs – Telly began the software selection process by analyzing the needs of his current simulation project and the anticipated needs of his future simulation projects. He did this by answering some questions.

What type of system will be modeled? Is it unique in function? A unique system is being modeled but a simulator package or language could be located to perform the task.

Who will be doing the simulation work? Telly would be doing the work himself. He was comfortable with his new computer skills and desired to use a full power simulation language.

What budgetary constraints exist? Telly had already obtained permission to purchase a software package within reason and had access to a computer system. So, budgetary constraints were not really a major consideration for this project.

Who will be using the results? The results of the study would be used by Kafy DePorres as a means of analyzing the bus route and making a purchase.

Step two: Research – Telly compiled a list of several simulation software packages including the one he had recently trained on – GPSS/World.

Step three: Compare – Telly developed an evaluation template to help compare available options in the software selection process. The weighting factors were subjectively developed from attributes shown in Figure 7.5.

	Weighting Factor		Product Score		Total
Graphics	1	X	____	=	____
User-Friendliness	7	X	____	=	____
Software capability	7	X	____	=	____
Ease of Use	7	X	____	=	____
Power	5	X	____	=	____
Cost	6	X	____	=	____
Output reports	9	X	____	=	____
Ease of Debugging	8	X	____	=	____
Statistics	7	X	____	=	____
Customer Support	6	X	____	=	____
Training	10	X	____	=	____
Documentation	7	X	____	=	____
Vendor stability	4	X	____	=	____
Application specific modules	1	X	____	=	____
			Total Product Score =		____

Figure 7.5 Simulation Software Evaluation Template

Step four: Make Selection – Telly used his template to evaluate four simulation products and decided, largely due to his training, that GPSS/World would be most suited to his current application.

7.3.8 Model Construction

Telly created his model using GPSS World. He broke his model into several segments to represent different elements of the system and initially used a block diagram to help conceptualize the code (Figure 7.6).

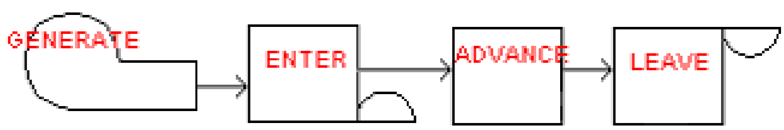


Figure 7.6 GPSS Block Diagram

The first segment represented the Bus Route (see Figure 7.7).

```
**********************************************************************
*  Bus Route With Starts and Stops and Waits for Passengers
**********************************************************************
         GENERATE   ,,,1                      ;Create Bus Transaction

Again    ADVANCE    (Exponential(1,12,4))   ; Bus Travels to Stop #1
         ENTER      STOP1                   ; Bus at Stop
         ADVANCE    (Normal(2,2,.3))        ; Passenger Loading/Unloading Time
         LEAVE      STOP1                   ; Leave Bus Stop

         ADVANCE    (Exponential(3,8,3))    ; Bus Travels to Stop #2
         ENTER      STOP2                   ; Bus at Stop
         ADVANCE    (Normal(4,2,.3))        ; Passenger Loading/Unloading Time
         LEAVE      STOP2                   ; Leave Bus Stop

         ADVANCE    (Exponential(5,14,4))   ; Bus Travels to Stop #3
         ENTER      STOP3                   ; Bus at Stop
         ADVANCE    (Normal(6,2,.3))        ; Passenger Loading/Unloading Time
         LEAVE      STOP3                   ; Leave Bus Stop

         ADVANCE    (Exponential(7,8,3))    ; Bus Travels to Stop #4
         ENTER      STOP4                   ; Bus at Stop
         ADVANCE    (Normal(8,2,.3))        ; Passenger Loading/Unloading Time
         LEAVE      STOP4                   ; Leave Bus Stop

         ADVANCE    (Exponential(9,15,4))   ; Bus Travels to Stop #5
         ENTER      STOP5                   ; Bus at Stop
         ADVANCE    (Normal(10,2,.3))       ; Passenger Loading/Unloading Time
         LEAVE      STOP5                   ; Leave Bus Stop
         TRANSFER   ,Again                  ; Move to Travel Back to Stop #1
```

Figure 7.7 GPSS Code for Bus Route

The second segment represented the passengers and their decision regarding where to get off the bus.

```
************************************************************************
         GENERATE   (POISSON(11,3.58))          ; Passengers at Stop 1
         QUEUE      STOP1Q                      ; Wait for Bus
         TEST E     BV$STOP1V,1                 ; Bus Arrives, If Space Get Onboard
         ENTER      BusSeats                    ; Sit in Seat
         DEPART     STOP1Q                      ; No Longer in Queue
         Transfer   .8,Next1a,OutAt2            ; Some go to Stop 2
Next1a   Transfer   .5,Next1b,OutAt3            ; Some go to Stop 3
Next1b   Transfer   .5,OutAt4,OutAt5            ; Some go to Stop 4 or 5

         GENERATE   (POISSON(12,3.22))          ;Passengers at Stop 2
         QUEUE      STOP2Q                      ; Wait for Bus
         TEST E     BV$STOP2V,1                 ; Bus Arrives, If Space Get Onboard
         ENTER      BusSeats                    ; Sit in Seat
         DEPART     STOP2Q                      ; No Longer in Queue
         Transfer   .76,Next2a,OutAt3           ; Some go to Stop 3
Next2a   Transfer   .834,Next2b,OutAt4          ; Some go to Stop 4
Next2b   Transfer   .5,OutAt5,OutAt1            ; Some go to Stop 5 or 1

         GENERATE   12,2                        ;Passengers at Stop 3
         QUEUE      STOP3Q                      ; Wait for Bus
         TEST E     BV$STOP3V,1                 ; Bus Arrives, If Space Get Onboard
         ENTER      BusSeats                    ; Sit in Seat
         DEPART     STOP3Q                      ; No Longer in Queue
         Transfer   .67,Next3a,OutAt4           ; Some go to Stop 4
Next3a   Transfer   .818,Next3b,OutAt5          ; Some go to Stop 5
Next3b   Transfer   .67,OutAt1,OutAt2           ; Some go to Stop 1 or 2

         GENERATE   (POISSON(13,3.34))          ;Passengers at Stop 4
         QUEUE      STOP4Q                      ; Wait for Bus
         TEST E     BV$STOP4V,1                 ; Bus Arrives, If Space Get Onboard
         ENTER      BusSeats                    ; Sit in Seat
         DEPART     STOP4Q                      ; No Longer in Queue
         Transfer   .76,Next4a,OutAt5           ; Some go to Stop 5
Next4a   Transfer   .625,Next4b,OutAt1          ; Some go to Stop 1
Next4b   Transfer   .67,OutAt2,OutAt3           ; Some go to Stop 2 or 3

         GENERATE   (POISSON(14,3.12))          ;Passengers at Stop 5
         QUEUE      STOP5Q                      ; Wait for Bus
         TEST E     BV$STOP5V,1                 ; Bus Arrives, If Space Get Onboard
         ENTER      BusSeats                    ; Sit in Seat
         DEPART     STOP5Q                      ; No Longer in Queue
         Transfer   .88,Next5a,OutAt1           ; Some go to Stop 1
Next5a   Transfer   .67,Next5b,OutAt2           ; Some go to Stop 2
Next5b   Transfer   .5,OutAt3,OutAt4            ; Some go to Stop 3 or 4
```

Figure 7.8 GPSS Code for Passengers

The third segment represents the passengers leaving the bus when it arrives at their expected stop.

```
OutAt1   QUEUE      WAIT41                ; Tracking Travel Time
         GATE SF    STOP1                 ; Passenger Waits for Stop
         LEAVE      Busseats              ; Gets out of Seat
         DEPART     WAIT41                ; Leaves Bus
         TERMINATE                        ; Is No Longer in Model

OutAt2   QUEUE      WAIT42                ; Tracking Travel Time
         GATE SF    STOP2                 ; Passenger Waits for Stop
         LEAVE      Busseats              ; Gets out of Seat
         DEPART     WAIT42                ; Leaves Bus
         TERMINATE                        ; Is No Longer in Model

OutAt3   QUEUE      WAIT43                ; Tracking Travel Time
         GATE SF    STOP3                 ; Passenger Waits for Stop
         LEAVE      Busseats              ; Gets out of Seat
         DEPART     WAIT43                ; Leaves Bus
         TERMINATE                        ; Is No Longer in Model

OutAt4   QUEUE      WAIT44                ; Tracking Travel Time
         GATE SF    STOP4                 ; Passenger Waits for Stop
         LEAVE      Busseats              ; Gets out of Seat
         DEPART     WAIT44                ; Leaves Bus
         TERMINATE                        ; Is No Longer in Model

OutAt5   QUEUE      WAIT45                ; Tracking Travel Time
         GATE SF    STOP5                 ; Passenger Waits for Stop
         LEAVE      Busseats              ; Gets out of seat
         DEPART     WAIT45                ; Leaves Bus
         TERMINATE                        ; Is No Longer in Model
```

Figure 7.9 GPSS Code for Departing Passengers

The fourth segment tracks model timing.

```
**********************************************************************
    GENERATE   600           ; 10 Hour Shift in Minute
    TERMINATE  1             ; Stop for Statistics after Each Shift
**********************************************************************
```

Figure 7.10 GPSS Timing Transaction

7.4 Quality Phase

In order to ensure the model was representative of its real world equivalent, Telly spent time on verification. First, during model construction and earlier activities he used preventative verification to ensure the model would not have many errors. He documented his code and used a structured approach to carefully organize the model. He tested individual sections of the code to ensure they operated as expected.

Telly also used appraisal verification to check the programming after it was been coded. He stepped through the model using GPSS World's step function to make sure the model operated as expected. He ran the model under simplified conditions to make sure it operated as expected. Finally, he checked the output reports and looked for any anomalies that might indicate a problem. After a few minor fixes, he was satisfied the model operated correctly.

7.5 Implementation

After having conceptualized, coded, verified, and validated his model, Telly was ready to start running a set of experiments to answer the primary questions about the system – in this instance, what size bus should be purchased. Telly initialized key parameters in the model and set up production runs so he could make inferences about the behavior of the system. He used the sequence of steps shown in Figure 7.11:

Figure 7.11 Experimentation Process

7.5.1 Experimental Design

The primary value of interest for Telly's simulation is "Size of Bus." This value can be obtained in several different ways. Telly decided to run his model with a variety of bus seat capacities and then examine related queuing times to determine the impact of using a smaller bus. Additionally, he set the following experimental conditions:

- Length of simulated run time – Telly used 10 hours for his model run time. The model matches th real world run time and is set to terminate after 10 hours of simulated time.
- Replications – Telly wanted to gather enough data to develop a 95% confidence interval. He decided to start with 30 replications which represents a month of 10 hour days.
- Reseeding random number streams – To ensure independence, Telly reset and cleared the model at the end of each 10 hour day. Each new day used fresh random number streams to ensure the same numbers were not reused for subsequent runs.
- Initial conditions – No special initial conditions were placed into the model. The people started to arrive and the bus began its circuit.

7.5.2 Statistical Output Analysis of a Single Model

Telly knew that the stochastic nature of his model required enough replications to ensure output variability was accounted for in terms of a mean and standard deviation. He wanted to construct a 95% confidence interval within which the true values of his model are likely to fall. He collected the following "Maximum Seat Capacity Required" based on a 64 seat bus. The output data failed the normality test so Telly was forced to use a larger sample size to more confidently capture all variance (Figure 7.12).

Tests for Normality

Test	--Statistic---		-----p Value------	
Shapiro-Wilk	W	0.834894	Pr < W	0.0003
Kolmogorov-Smirnov	D	0.230424	Pr > D	<0.0100
Cramer-von Mises	W-Sq	0.24772	Pr > W-Sq	<0.0050
Anderson-Darling	A-Sq	1.528084	Pr > A-Sq	<0.0050

Figure 7.12 Failed Test for Normality on Small Data Set

Telly changed to 45 replications and retested for normality. The new Shapiro-Wilk W of .965 does not reject the null hypothesis that the variable is normally distributed (p<.1889). Likewise, Kolmogorov-Smirnov, Cramer-von Mises, and Anderson-Darling tests do not reject the null hypothesis. Additionally, the plots (shown in Figure 7.13) indicate normality cannot be rejected.

Tests for Normality

Test	--Statistic---		-----p Value------	
Shapiro-Wilk	W	0.964989	Pr < W	0.1889
Kolmogorov-Smirnov	D	0.11801	Pr > D	0.1154
Cramer-von Mises	W-Sq	0.058065	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.375992	Pr > A-Sq	>0.2500

Figure 7.13 Test to Establish Normality with Larger Sample Size

He further checked with a Stem and Leaf Plot, a Boxplot, and a Normal Probability Plot. All indicated normality (See Figure 7.14).

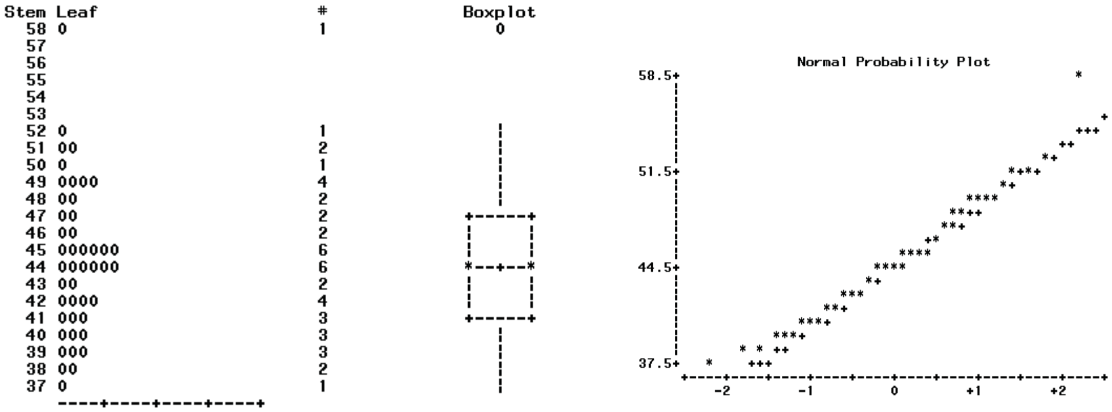


Figure 7.14 Normality Plots for Data

Given that the data is assumed normally distributed, Telly constructed a 95% confidence interval regarding the Maximum number of seats used on a 64 seat bus. SAS indicated a mean of 44.47 with a 95% confidence interval (43.16,45.77) as shown in Figure 7.15.

```
                    Basic Statistical Measures

      Location                    Variability

Mean     44.46667     Std Deviation            4.36203
Median   44.00000     Variance                19.02727
Mode     44.00000     Range                   21.00000
                      Interquartile Range      6.00000
```

The MEANS Procedure

Analysis Variable : SIMRUN

Lower 95% CL for Mean	Upper 95% CL for Mean
43.1561689	45.7771644

Figure 7.15 Confidence Interval for Mean of Maximum Bus Seats

Feeling confident his sample size was large enough, Telly decided to create a baseline statistic representing average waiting time at Bus Stop 1. He selected Bus Stop 1 since this area was visible to customers purchasing tickets and if long lines were visible, it might discourage potential customers. The baseline value was intended to give analysts an idea of the expected average wait time since there is never demand for bus capacity beyond what is possible. In other words, with the current configuration, this is the best the system can perform. This value then can be compared to runs using smaller bus sizes to help determine the best bus to purchase.

Telly established normality for this baseline data set then tabulated an average queue time of 39.89 minutes with a 95% Confidence Interval (39.12, 40.65). Telly wanted to assess the impact of dropping down to 48, 32, and 24 seat buses (those were the available configurations). The simulation was rerun for those scenarios and data was collected. First Telly compared the maximum number of seats used in each scenario. Table 7.8 summarizes the results:

Scenario	Average of Maximum Seats Used
64 Seat Bus	44.47
48 Seat Bus	43.84
32 Seat Bus	32 (Always hit Maximum)
24 Seat Bus	24 (Always hit Maximum)

Table 7.8 Average Maximum Bus Seats Used

Then he used MS-Excel to develop an ANOVA to determine if any significant difference existed between the various seat configurations. Figure 7.16 summarizes:

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
64 Seat Bus	45	1794.96	39.89	6.56
48 Seat Bus	45	1798.233	9.96 7	.36
32 Seat Bus	45	2007.06	44.60	67.18
24 Seat Bus	45	3896.38	86.59	264.91

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	69313	3	23104.31	267.10	2.93E-65	2.66
Within Groups	15224	176	86.50			
Total	84537	179				

Figure 7.16 ANOVA Result for Bus Seat Configurations

The results indicated that on average, customers at Bus Stop #1 waited an average of 39.89 minutes for a 64 seat bus, 39.96 minutes for the 48 seat bus, 44.6 minutes for the 32 seat bus and 86.59 minutes for the 24 seat bus. Telly further analyzed the data and discovered no significant difference existed between the queue times in the 64 and 48 seat buses. Since the 24 seat bus appeared too limited and didn't adequately move the passengers fast enough, Telly wanted to compare the 48 seat and 32 seat buses more closely. Figure 7.17 shows that a significant difference between these two configurations exists and that approximately 5 minutes is saved by using the larger bus.

	48 Seats	**32 Seats**
Mean	39.96	44.60
Variance	7.36	67.18
Observations	45	45
Pooled Variance	37.27	
Hypothesized Mean Difference	0	
Df	88	
t Stat	-3.61	
P(T<=t) one-tail	0.00	
t Critical one-tail	1.66	
P(T<=t) two-tail	0.001	
t Critical two-tail	1.99	

Figure 7.17 48 and 32 Seat Buses Compared

7.5.3 Additional Scenario

Telly developed a report to communicate his findings to Kafy DePorres. In addition to the findings about maximum seats used and average times in queue, other statistics were gathered including average utilization of the bus, average route times, and number of passengers moved.

Kafy spent time reviewing the data and decided that none of the options were really very desirable. The average queue time in the 48 seat bus scenario was still too high at 39.96 minutes. So an additional scenario was devised. Kafy asked Telly to rerun the model with two 24 seat buses that were staggered so multiple locations could be serviced simultaneously. Telly agreed to recode the model so that would work (see code at end in Appendix 7.8). The model was run and compared against the 48 seat version of the single bus model. Figure 7.18 shows the comparison of Bus Stop 1 waiting Times.

	48 Seat	Two 24 Seat Buses
Mean	39.96	29.37
Variance	7.36	66.63
Observations	45	45
Pooled Variance	36.99	
Hypothesized Mean Difference	0	
Df	88	
t Stat	8.26	
P(T<=t) one-tail	6.897E-13	
t Critical one-tail	1.66	
P(T<=t) two-tail	1.379E-12	
t Critical two-tail	1.99	

Figure 7.18 Comparison of Waiting Times Between 2 Bus System and 48 Seat Bus

The new results provided a much better solution – using two 24 seat buses, the average waiting times dropped by over ten minutes. Telly formalized his findings in a report and submitted it to Kafy. Eventually two 24 seat buses were purchased and implemented on the route.

7.6 Operations, Maintenance and Archival Phase

The model was stored, together with documentation and information about the assumptions used. If the routes need to be modified or customer numbers increase, the model can be used to answer additional questions.

7.7 Bibliography

For additional information about the simulation language being used in this chapter, see the following Web resources: http://www.minutemansoftware.com/

7.8 Appendix: GPSS World Source Code Listing (Two Bus Model)

```
***********************************************************************
*          MiniBus Route Model                    *
*          Time units are in Minutes              *
***********************************************************************

           RMULT 9441              ;Reseed Random Number Generators
BusSeat1 STORAGE 24                ;Bus 1 can hold 24 people
BusSeat2 STORAGE 24                ;Bus 2 can hold 24 people
STOP1a STORAGE 1                   ;Stop 1a holds 1 bus
STOP2a STORAGE 1                    ;Stop 2a holds 1 bus
STOP3a STORAGE 1                   ;Stop 3a holds 1 bus
STOP4a STORAGE 1                   ;Stop 4a holds 1 bus
STOP5a STORAGE 1                   ;Stop 5a holds 1 bus
STOP1b STORAGE 1                   ;Stop 1b holds 1 bus
STOP2b STORAGE 1                   ;Stop 2b holds 1 bus
STOP3b STORAGE 1                   ;Stop 3b holds 1 bus
STOP4b STORAGE 1                   ;Stop 4b holds 1 bus
STOP5b STORAGE 1                   ;Stop 5b holds 1 bus

* These Boolean Variables check to see if a bus arrives and if room exists on it.
STOP1V BVARIABLE SF$STOP1a&R$BusSeat1'GE'1|SF$STOP1b&R$BusSeat2'GE'1
STOP2V BVARIABLE SF$STOP2a&R$BusSeat1'GE'1|SF$STOP2b&R$BusSeat2'GE'1
STOP3V BVARIABLE SF$STOP3a&R$BusSeat1'GE'1|SF$STOP3b&R$BusSeat2'GE'1
STOP4V BVARIABLE SF$STOP4a&R$BusSeat1'GE'1|SF$STOP4b&R$BusSeat2'GE'1
STOP5V BVARIABLE SF$STOP5a&R$BusSeat1'GE'1|SF$STOP5b&R$BusSeat2'GE'1

* These Boolean Variables are used to see if it is Bus 1 or 2 that has stopped
BUS1S1 BVARIABLE SF$STOP1a&R$BusSeat1'GE'1
BUS1S2 BVARIABLE SF$STOP2a&R$BusSeat1'GE'1
BUS1S3 BVARIABLE SF$STOP3a&R$BusSeat1'GE'1
BUS1S4 BVARIABLE SF$STOP4a&R$BusSeat1'GE'1
BUS1S5 BVARIABLE SF$STOP5a&R$BusSeat1'GE'1

* These Boolean Variables are used to check if a bus is at station a or b
ATSTOP1 BVARIABLE SF$STOP1a|SF$STOP1b
ATSTOP2 BVARIABLE SF$STOP2a|SF$STOP2b
ATSTOP3 BVARIABLE SF$STOP3a|SF$STOP3b
ATSTOP4 BVARIABLE SF$STOP4a|SF$STOP4b
ATSTOP5 BVARIABLE SF$STOP5a|SF$STOP5b
```

```
*********************************************************************
* Bus Route 1 with Starts and Stops and Waits for Passengers
*********************************************************************

                GENERATE    ,,,1                    ;Create Bus Transaction for Station 1
Start
                ASSIGN      BUSNUM,1
Again           ADVANCE     (Exponential(1,12,4))   ; Bus Travels to Stop #1
                ENTER       STOP1a                  ; Bus at Stop
                ADVANCE     (Normal(2,2,.3))        ; Passenger Loading/Unloading Time
                LEAVE       STOP1a                  ; Leave Bus Stop

                ADVANCE     (Exponential(3,8,3))    ; Bus Travels to Stop #2
                ENTER       STOP2a                  ; Bus at Stop
                ADVANCE     (Normal(4,2,.3))        ; Passenger Loading/Unloading Time
                LEAVE       STOP2a                  ; Leave Bus Stop

                ADVANCE     (Exponential(5,14,4))   ; Bus Travels to Stop #3
                ENTER       STOP3a                  ; Bus at Stop
                ADVANCE     (Normal(6,2,.3))        ; Passenger Loading/Unloading Time
                LEAVE       STOP3a                  ; Leave Bus Stop

                ADVANCE     (Exponential(7,8,3))    ; Bus Travels to Stop #4
                ENTER       STOP4a                  ; Bus at Stop
                ADVANCE     (Normal(8,2,.3))        ; Passenger Loading/Unloading Time
                LEAVE       STOP4a                  ; Leave Bus Stop

                ADVANCE     (Exponential(9,15,4))   ; Bus Travels to Stop #5
                ENTER       STOP5a                  ; Bus at Stop
                ADVANCE     (Normal(10,2,.3))       ; Passenger Loading/Unloading Time
                LEAVE       STOP5a                  ; Leave Bus Stop
                TRANSFER    ,Again                  ; Move to Travel Back to Stop #1
```

```
**********************************************************************
* Bus Route 2 With Starts and Stops and Waits for Passengers
**********************************************************************

        GENERATE    ,,,1                    ;Create Bus Transaction for Station 3
Start
        ASSIGN      BUSNUM,2
        TRANSFER    ,Bus2St                 ;Start Day at Station #3

Again2  ADVANCE     (Exponential(1,12,4))   ; Bus Travels to Stop #1
        ENTER       STOP1b                  ; Bus at Stop
        ADVANCE     (Normal(2,2,.3))        ; Passenger Loading/Unloading Time
        LEAVE       STOP1b                  ; Leave Bus Stop

        ADVANCE     (Exponential(3,8,3))    ; Bus Travels to Stop #2
        ENTER       STOP2b                  ; Bus at Stop
        ADVANCE     (Normal(4,2,.3))        ; Passenger Loading/Unloading Time
        LEAVE       STOP2b                  ; Leave Bus Stop

        ADVANCE     (Exponential(5,14,4))   ; Bus Travels to Stop #3
        ENTER       STOP3b                  ; Bus at Stop
        ADVANCE     (Normal(6,2,.3))        ; Passenger Loading/Unloading Time
        LEAVE       STOP3b                  ; Leave Bus Stop

Bus2St  ADVANCE     (Exponential(7,8,3))    ; Bus Travels to Stop #4
        ENTER       STOP4b                  ; Bus at Stop
        ADVANCE     (Normal(8,2,.3))        ; Passenger Loading/Unloading Time
        LEAVE       STOP4b                  ; Leave Bus Stop

        ADVANCE     (Exponential(9,15,4))   ; Bus Travels to Stop #5
        ENTER       STOP5b                  ; Bus at Stop
        ADVANCE     (Normal(10,2,.3))       ; Passenger Loading/Unloading Time
        LEAVE       STOP5b                  ; Leave Bus Stop
        TRANSFER    ,Again2                 ; Move to Travel Back to Stop #1
```

```
*****************************************************************************************
        GENERATE 600                                    ; 10 Hour Shift in Minute
        TERMINATE 1                                     ; Stop for Statistics after Each Shift
*****************************************************************************************

                GENERATE    (POISSON(11,3.58))          ; Passengers at Stop 1 Arrive
                QUEUE       STOP1Q                      ; Wait for Bus
                TEST E      BV$STOP1V,1                 ; Bus Arrives, If Space Get Onboard
                TEST E      BV$BUS1S1,1,BUS2S1          ; Is this Bus 1?
                ENTER       BusSeat1                    ; If Bus 1, Get in your Seat
                ASSIGN      BUSNUM,1                    ; Passenger is marked with Bus 1
                TRANSFER    ,ONBUS1                     ; Bus 1 Passenger set

BUS2S1          ENTER       BusSeat2                    ; If Bus 2, Get in your Seat
                ASSIGN      BUSNUM,2                    ; Passenger is marked with Bus 2

ONBUS1          DEPART      STOP1Q                      ; No Longer in Queue
                Transfer    .8,Next1a,OutAt2            ; Some go to Stop 2
Next1a          Transfer    .5,Next1b,OutAt3            ; Some go to Stop 3
Next1b          Transfer    .5,OutAt4,OutAt5            ; Some go to Stop 4 or 5

                GENERATE    (POISSON(12,3.22))          ; Passengers at Stop 2
                QUEUE       STOP2Q                      ; Wait for Bus
                TEST E      BV$STOP2V,1                 ; Bus Arrives, If Space Get Onboard
                TEST E      BV$BUS1S2,1,BUS2S2          ; Is this Bus 1?
                ENTER       BusSeat1                    ; If Bus 1, Get in your Seat
                ASSIGN      BUSNUM,1                    ; Passenger is marked with Bus 1
                TRANSFER    ,ONBUS2                     ; Bus 1 Passenger set

BUS2S2          ENTER       BusSeat2                    ; If Bus 2, Get in your Seat
                ASSIGN      BUSNUM,2                    ; Passenger is marked with Bus 2

ONBUS2          DEPART      STOP2Q                      ; No Longer in Queue
                Transfer    .76,Next2a,OutAt3           ; Some go to Stop 3
Next2a          Transfer    .834,Next2b,OutAt4          ; Some go to Stop 4
Next2b          Transfer    .5,OutAt5,OutAt1            ; Some go to Stop 5 or 1
                GENERATE    12,2                        ; Passengers at Stop 3
                QUEUE       STOP3Q                      ; Wait for Bus
                TEST E      BV$STOP3V,1                 ; Bus Arrives, If Space Get Onboard
```

```
            TEST E      BV$BUS1S3,1,BUS2S3      ; Is this Bus 1?
            ENTER       BusSeat1                ; If Bus 1, Get in your Seat
            ASSIGN      BUSNUM,1                ; Passenger is marked with Bus 1
            TRANSFER    ,ONBUS3                 ; Bus 1 Passenger set

BUS2S3      ENTER       BusSeat2                ; If Bus 2, Get in your Seat
            ASSIGN      BUSNUM,2                ; Passenger is marked with Bus 2

ONBUS3      DEPART      STOP3Q                  ; No Longer in Queue
            Transfer    .67,Next3a,OutAt4       ; Some go to Stop 4
Next3a      Transfer    .818,Next3b,OutAt5      ; Some go to Stop 5
Next3b      Transfer    .67,OutAt1,OutAt2       ; Some go to Stop 1 or 2
            GENERATE    (POISSON(13,3.34))      ; Passengers at Stop 4
            QUEUE       STOP4Q                  ; Wait for Bus
            TEST E      BV$STOP4V,1             ; Bus Arrives, If Space Get Onboard
            TEST E      BV$BUS1S4,1,BUS2S4      ; Is this Bus 1?
            ENTER       BusSeat1                ; If Bus 1, Get in your Seat
            ASSIGN      BUSNUM,1                ; Passenger is marked with Bus 1
            TRANSFER    ,ONBUS4                 ; Bus 1 Passenger set
```

```
BUS2S4    ENTER      BusSeat2                 ; If Bus 2, Get in your Seat
          ASSIGN     BUSNUM,2                 ; Passenger is marked with Bus 2

ONBUS4    DEPART     STOP4Q                   ; No Longer in Queue
          Transfer   .76,Next4a,OutAt5        ; Some go to Stop 5
Next4a    Transfer   .625,Next4b,OutAt1       ; Some go to Stop 1
Next4b    Transfer   .67,OutAt2,OutAt3        ; Some go to Stop 2 or 3
          GENERATE   (POISSON(14,3.12))       ; Passengers at Stop 5
          QUEUE      STOP5Q                   ; Wait for Bus
          TEST E     BV$STOP5V,1              ; Bus Arrives, If Space Get Onboard
          TEST E     BV$BUS1S5,1,BUS2S5       ; Is this Bus 1?
          ENTER      BusSeat1                 ; If Bus 1, Get in your Seat
          ASSIGN     BUSNUM,1                 ; Passenger is marked with Bus 1
          TRANSFER   ,ONBUS5                  ; Bus 1 Passenger set

BUS2S5    ENTER      BusSeat2                 ; If Bus 2, Get in your Seat
          ASSIGN     BUSNUM,2                 ; Passenger is marked with Bus 2

ONBUS5    DEPART     STOP5Q                   ; No Longer in Queue
          Transfer   .88,Next5a,OutAt1        ; Some go to Stop 1
Next5a    Transfer   .67,Next5b,OutAt2        ; Some go to Stop 2
Next5b    Transfer   .5,OutAt3,OutAt4         ; Some go to Stop 3 or 4

***********************************************************************

OutAt1    QUEUE      WAIT41                   ; Tracking Travel Time
          TEST E     BV$ATSTOP1,1             ; Passenger Waits for Stop
          TEST E     P$BUSNUM,1,OBus2A1       ; On Bus 1?
          LEAVE      Busseat1                 ; Gets out of Seat on Bus 1
          TRANSFER   ,OutAt1b                 ; Leave Bus 1

OBus2A1   LEAVE      Busseat2                 ; Gets out of Seat on Bus 2
OutAt1b   DEPART     WAIT41                   ; Leaves Bus Area
          TERMINATE                           ; Is No Longer in Model
```

```
OutAt2     QUEUE      WAIT42               ; Tracking Travel Time
           TEST E     BV$ATSTOP2,1         ; Passenger Waits for Stop
           TEST E     P$BUSNUM,1,OBus2A2   ; On Bus 1?
           LEAVE      Busseat1             ; Gets out of Seat on Bus 1
           TRANSFER   ,OutAt2b             ; Leave Bus 1

OBus2A2    LEAVE      Busseat2             ; Gets out of Seat on Bus 2
OutAt2b    DEPART     WAIT42               ; Leaves Bus Area
           TERMINATE                       ; Is No Longer in Model

OutAt3     QUEUE      WAIT43               ; Tracking Travel Time
           TEST E     BV$ATSTOP3,1         ; Passenger Waits for Stop
           TEST E     P$BUSNUM,1,OBus2A3   ; On Bus 1?
           LEAVE      Busseat1             ; Gets out of Seat on Bus 1
           TRANSFER   ,OutAt3b             ; Leave Bus 1

OBus2A3    LEAVE      Busseat2             ; Gets out of Seat on Bus 2
OutAt3b    DEPART     WAIT43               ; Leaves Bus Area
           TERMINATE                       ; Is No Longer in Model

OutAt4     QUEUE      WAIT44               ; Tracking Travel Time
           TEST E     BV$ATSTOP4,1         ; Passenger Waits for Stop
           TEST E     P$BUSNUM,1,OBus2A4   ; On Bus 1?
           LEAVE      Busseat1             ; Gets out of Seat on Bus 1
           TRANSFER   ,OutAt4b             ; Leave Bus 1

OBus2A4    LEAVE      Busseat2             ; Gets out of Seat on Bus 2
OutAt4b    DEPART     WAIT44               ; Leaves Bus Area
           TERMINATE                       ; Is No Longer in Model

OutAt5     QUEUE      WAIT45               ; Tracking Travel Time
           TEST E     BV$ATSTOP2,1         ; Passenger Waits for Stop
           TEST E     P$BUSNUM,1,OBus2A5   ; On Bus 1?
           LEAVE      Busseat1             ; Gets out of Seat on Bus 1
           TRANSFER   ,OutAt5b             ; Leave Bus 1

OBus2A5    LEAVE      Busseat2             ; Gets out of Seat on Bus 2
OutAt5b    DEPART     WAIT45               ; Leaves Bus Area
           TERMINATE                       ; Is No Longer in Model
```

Acknowledgements

I would like to thank several people for their contributions to this textbook. Amy Rucker, a graduate assistant at K-State, provided help developing illustrations used in the text. My wife, Annette McHaney, proofread the manuscript and provided a valuable critique, provided inspiration, and suggested helpful changes.