10-2017

Using Microsoft Excel to Teach Simulation Concepts to Business Students

Robert F. Gordon Ph.D.
Molloy College, rfgordon@molloy.edu

Using Microsoft Excel to Teach Simulation Concepts to Business Students

Robert F. Gordon
Department of Mathematics and Computer Studies
Molloy College
1000 Hempstead Avenue
Rockville Centre, NY USA 11571
(516) 323-3431
rfgordon@molloy.edu

ABSTRACT
The application of computers to solving business problems, the area of study known as decision support systems, is an important component in the education of business students today. One major type of decision support system is computer simulation, which is the technique most often used to solve queuing problems in industry. This paper describes how to teach the concepts of computer simulation, explain the key components of simulation software, and provide hands-on experience to solve these problems by using Microsoft Excel.

Keywords
Computer simulation, decision support systems, pedagogy

1 INTRODUCTION

Discrete event simulation is used to solve a wide range of business problems that involve queuing. These problems include designing manufacturing lines, communications systems, transportation networks, computer configurations, and customer service centers, basically, any situation where customers (people, messages, vehicles, products) are competing for limited service resources and thus waiting occurs. Except for a restricted class of model formulations, queuing problems cannot be solved analytically, but can only be solved numerically, requiring a large amount of computation and replication (Gross 1998). Therefore, computer simulation, specifically Monte Carlo discrete event simulation, is the method used to solve these problems.

In this approach, the business analyst, perhaps using time-and-motion studies of a facility, provides the input data, consisting of the arrival distribution of customers, the processing rules and service distribution, and the routing of customers from one service area to the next. The simulation software uses sequences of random numbers to repeatedly sample from the arrival and service distributions, so that over the long run the steady-state results will mimic the real-world activity. The simulation software tracks the resulting performance measures, including utilization of servers, waiting time, time in the facility. Then the analyst presents alternative scenarios to the computer simulation software by changing the inputs to the model, such as the number of servers, inter-arrival times and/or service times. By comparing the performance measures from each scenario, the analyst decides how to design the facility to get the desired performance at an appropriate cost.

Typically, to teach how to use computer simulation to solve queuing problems, students need to have access to specialized simulation software. Simulation software provides them with the hands-on experience of creating, manipulating, and evaluating simulation models. This capability would require a school to acquire an expensive simulation product, such as RESQME (Gordon 1986), Simul8 (Concannon 2003), or Arena (Kelton 2014). In addition to the expense, these software products have a steep learning curve to be used effectively, taking away from class time, delaying when the simulation concepts can be taught, and reducing student interest.

The methodology, described in this paper, is to create components with Microsoft Excel that students can assemble to produce their own simulation models. Excel will not provide the visual programming, higher-level modeling, or animation capabilities (Gordon 1994) of the simulation products mentioned above, but will provide the necessary tools to create and analyze a simulation model.

We chose Microsoft Excel, because it is readily available, low cost, and familiar to most students. The objective is to provide students with the ability to quickly create simulation models with Excel. So instead of spending time learning a simulation modeling tool, students can devote their time to learning how to model queuing problems and how to use the results to make decisions. Interacting with the Excel simulation model, students can concentrate on the techniques to capture data, set up distributions, replicate events, formulate alternative scenarios, evaluate performance measures, and apply the results to real-world problems.

2 REQUIRED SOFTWARE COMPONENTS

Simulation software must provide tools to:

a. Sample from arrival distributions and service distributions to reflect reality,
b. Implement logic to select the next customer to process at the service centers,
c. Set up a clock mechanism to move time forward through events,

d. Summarize the performance measures,
e. Replicate the experiment to account for different random number streams, and
f. Easily change the inputs in order to compare alternatives.

We use the built-in functions and calculating environment of Excel to create these necessary components of simulation software. The implementation of each of the required components in Excel is described in the following paragraphs.

The ability to sample from distributions is accomplished by combining a random number generating function, such as RANDBETWEEN, the lookup function, VLOOKUP, and the table facility. Specifically, the inter-arrival table consists of a row for each possible inter-arrival time and two columns: a Monte Carlo number representing the cumulative probability for a given inter-arrival time and the corresponding inter-arrival time. The table is sequenced in the order of increasing Monte Carlo numbers (cumulative probabilities). The VLOOKUP function using the approximate option (last argument is 1) has, as its inputs, the random number generated (Monte Carlo number) and the inter-arrival table, which it then uses to select the corresponding inter-arrival time (column 2) in the table. Thus the VLOOKUP function with an inter-arrival table, for example named Table1, would be coded as follows: VLOOKUP(RANDBETWEEN(0,9),Table1,2,1). The range of the random numbers depends on the required number of decimal places of the cumulative probabilities in Table1: one decimal place would require ten random numbers 0 to 9 as above; two decimal places would require one hundred random number 0 to 99, etc. The arrival time of a customer would be calculated iteratively as the prior customer's arrival time plus the value provided by the VLOOKUP function.

A similar approach is taken with service time. Service time might be a probability distribution represented in a table as above or could be determined by an equation whose variables follow probability distributions. The service time equation might be a function of variables, such as the number of transactions and/or type of transactions. With each variable in the equation following a probability distribution, its value would be determined by the VLOOKUP function as it samples from the distribution table using a generated random number.

For either arrivals or service, if a variable follows a standard statistical distribution, then instead of listing the possibilities in a table and using the VLOOKUP function, we can use that Excel statistical function with a random number argument to generate values for the variable. So, suppose service time = .2 + .7 x transactions with the number of transactions uniformly distributed between 1 and 4, we would use the formula = .2 + .7 * RANDBETWEEN(1,4) to calculate the service time for a given customer.

The student can use variations of these constructs to quickly learn the concept of sampling from probability distributions and to see how to use this mechanism to reflect real-world events.

Process logic is accomplished in Excel by using imbedded IF statements combined with logical operators. As an example, the IF statement might be used first to determine which line to join based on criteria, such as which server is idle or requirements of the customer to be serviced. The MIN function can be used to select the shortest line to join. Once on line, to determine when service starts (assuming customers do not switch lines), the start of service time for the current customer in the queue would be the greater of the time when either the customer in front of it completes service or when the current customer arrives in the queue. That could be given by a function such as MAX(B3,E2), where E2 is the end of service time of the prior customer and B3 is the arrival time of the current customer. The student can use these logical constructs to investigate the impact of different processing rules (such as first come, first served or priority processing) to see for themselves which works best in different situations.

The clock mechanism to advance time is handled in Excel by using the replicate feature (Autofill) to move forward through future customers and events. When the complete set of events for a given customer is formulated on one row in Excel by using random numbers, VLOOKUP functions, IF statements, and logical operators as described above, the student can replicate the row for any number of future customers by the autofill feature. Autofill is accomplished by highlighting the row for the current customer and dragging the mouse down one row for each future customer. With relative addressing, this essentially creates a DO loop, iterating the formulas and updating the events as we move down rows of the spreadsheet. The random number generator produces different values on each row, and the IF and logical statements allow the computer to determine arrival, service, and completion times based on the previous events. This is equivalent to writing a computer program to iteratively process the events of a specified number of customers.

To easily analyze the results of a simulation run, we use the Excel arithmetic and statistical functions to summarize the performance measures. We calculate, for example, overall performance measures with the AVERAGE function, such as the average waiting time and average length of the queue and with the ratio of sums, the percent utilization of the servers. We create built-in graphic displays using the Excel bar chart facility to show performance measures over the time horizon of the model. The displays automatically change to reflect changes in input. Such charts might show how time in the facility varies by time over the model horizon. The student can see directly the impact of different input levels in the graphic displays. In addition, the student can use function key 9 to change the random number and regenerate the events, so the student can view the impact of the random number sequence on the resulting graphs instantaneously.

Replicating the experiment with different random number streams is necessary to get more accurate results by reducing sampling error. This is accomplished in Excel by using the What-if Analysis menu item and the Data Table submenu selection. This technique generates multiple trials with different random number streams and stores the resulting performance measures for each individual model run in a table, one row for each trial. Those measures can then be summarized across the multiple trials. Averages and standard deviations are automatically calculated for the student.

Lastly, the student can easily modify the input data to reflect alternative scenarios and immediately see the results numerically and graphically in the performance measures. The input distributions and equations are set up with parameters to easily change the tables for arrival times and service times. The number of servers can be varied as well. This gives the student hands-on experience in evaluating alternatives to arrive at the problem solution.

3 SPECIFIC EXAMPLE

Our approach is to provide students with an Excel platform that contains the simulation components, described in the previous section. The Excel platform is a partially-constructed spreadsheet that the student fills and modifies to complete the specification of a simulation model. The platform contains the processing logic for the first few customers, the initial arrival and service distribution tables, the output graph structures, the Data Table setup, and the formulas to summarize the performance measures.

First, the student autofills this shell to as many customers as desired in the experiment. As that is completed, the results are simultaneously reflected in the performance measures and the graphs. Then the student replicates the experiment a number of times with the What-if Analysis menu item to view the average and standard deviation of the performance measures across the multiple trials. The student can then modify the inputs of arrivals and service to create alternative scenarios. The student then views the resulting system performance measures of each scenario to compare the results and arrive at a decision as to which of the presented scenarios is best.

We now give a specific example of such an Excel platform by simulating customers arriving at a bank with one teller, waiting on line, and being serviced.

The inter-arrival table set up in Excel is shown in the figure for Table 1. The first column shows the Monte Carlo number cut-off points for the cumulative probabilities, and the second column lists the corresponding inter-arrival times. In this example, 20% of the customers arrive within 1 minute of the prior customer, 50% within 2 minutes, and 30% within 3 minutes. The number of possible inter-arrival times, their values and probability of occurrence can be changed by the student for alternative scenarios.

Table 1.

Inter-arrival Table	
MC	Minutes
0	1
2	2
7	3

The formula for service time that is used in this example is: Service Time =.2+.7*RANDBETWEEN(1.4). As previously explained, it calculates the service time for a customer based on the number of transactions that customer has, which varies uniformly from 1 to 4. Service time for a customer is .2 minutes setup plus .7 minutes per transaction. The student can change this formula to examine other scenarios.

Row 1, columns A through G, respectively, of the Excel spreadsheet display the headings for the events of each customer in the simulation as follows: Customer_Number, Arrival_Time, Service_Time, Start_Service, End_Service, Time_In_Bank, Wait_Time.

This layout is followed for each customer, beginning in row 2 for the first customer. The general formulas that will be replicated for all the future customers are shown as follows for the second customer (row 3):

Arrival_Time in B3: VLOOKUP(RANDBETWEEN(0,9), Table1, 2, 1)+B2
In the above formula, B2 is the arrival time of customer 1.

Service_Time in C3: N12+L12*RANDBETWEEN(1,4)
In the above formula, the cells N12 and L12, absolutely addressed, contain the initial values .2 and .7, respectively.

Start_Service in D3: MAX(B3,E2)
In the above formula, B3 is the arrival time of customer 2, and E2 is the end of service time of customer 1.

End_Service in E3: C3+D3
In the above formula, C3 is the Service time of customer 2, and D3 is the start of service time of customer 2.

Time_In_Bank in F3: E3-B3
In the above formula, E3 is the end of service time of customer 2, and B3 is the arrival time of customer 2.

Wait_Time in G3: D3-B3
In the above formula D3 is the start of service time of customer2, and B3 is the arrival time of customer 2.

As the student autofills the rows for the first 50 customers, the bar graph shows the time in the bank for each of the customers, and the overall performance measures are produced. Here is the resulting graph for one set of random numbers. The average wait time is under 1 minute, and the overall time in the bank for customers is under four minutes.

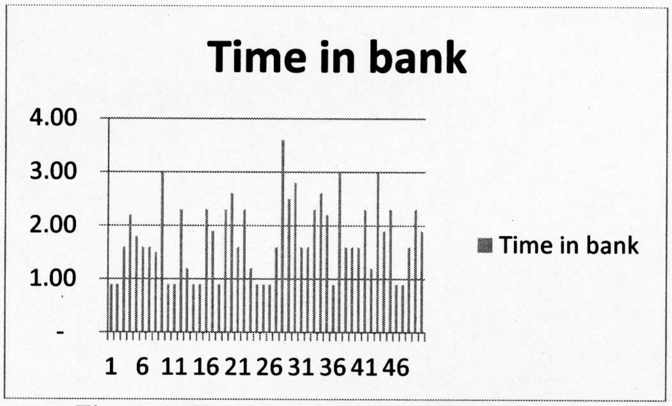

Figure 1. Graphic output shell for scenario 1.

Pressing function key 9 will generate new random numbers and automatically revise the events, performance measures, and resulting graphs. The results of each trial are stored in a Data Table by the student using the What-if Analysis menu item to replicate the experiment any number of times. The resulting performance measures are averaged across the multiple runs and summary statistics produced to more accurately estimate the performance measures.

The student can easily modify the inputs at this point, such as the arrival distribution or service equation, and then rerun the simulation to view its impact on the performance measures.

For example, here is the resulting graph, when the student changes the service equation's parameter from .7 minutes to 1 minute per transaction. The average wait time is now 20 minutes, time in the bank is approaching 50 minutes, and it is obvious from the graph for scenario 2 that wait times continue to increase as time goes on. The server is too slow to keep up with the demand, and the system does not reach a steady-state.

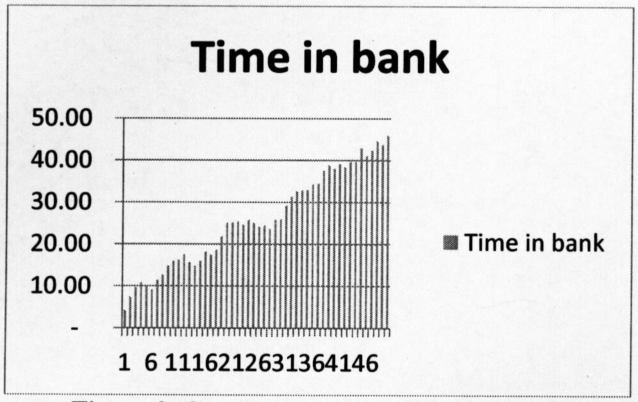

Figure 2. Graphic output shell for scenario 2.

The platform incorporates Excel statistical functions for the student to compare the performance measures of the alternative scenarios and analyze summary statistics. In this way, the student learns how to use the results to make meaningful decisions.

4 SUMMARY

In this paper, we described how to adapt and incorporate specific functions and control mechanisms of Excel to create the functionality of Monte Carlo discrete event simulation software. The student can use the extended Excel platform to model real-world queuing problems, experiment with different inputs, examine the resulting performance measures, and as a result, decide on the proper action to take. The student learns first-hand the components of a computer simulator and the framework and concepts for solving queuing problems in business.

REFERENCES

Concannon, K., et al., 2003, *Simulation Modeling with SIMUL8*, Visual Thinking International Ltd.

Gordon, R. F., et al., 1986, *Examples of Using the RESearch Queueing Package Modeling Environment*, Proceedings of the Winter Simulation Conference, Washington, D.C.

Gordon, R.F., 1994, *End-User Object-Oriented Programming By Means of an Object-Oriented Interface,* OOPS Messenger, ACM SIGPLAN.

Gross, D. and Harris, C., 1998*, Fundamentals of Queueing Theory*, Wiley.

Kelton, W. D., et al., 2014, *Simulation with Arena,* McGraw-Hill.