Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A MANAGER-FRIENDLY PLATFORM FOR SIMULATION MODELING AND ANALYSIS OF CALL CENTER QUEUEING SYSTEMS

Robert Saltzman
Vijay Mehrotra

Department of Decision Sciences
College of Business
San Francisco State University
San Francisco, CA 94132, U.S.A.

ABSTRACT

Call center operational performance is measured largely through queue times and customer abandonment rates, and thus managers have an acute need to understand how both management policies and stochastic factors affect these performance statistics. Simulation is an excellent vehicle for examining these relationships, but a lack of programming ability can be a barrier that prevents call center managers from making use of such models. To address this problem, we have developed a user-friendly Excel interface for a dynamic discrete event simulation model. The underlying model is a general queuing system for which analytical results are often unavailable, and the Excel interface enables managers to interactively specify a wide range of system parameters and analyze results, all without exposing them to the simulation model's components. Based on input from call center operations managers, we have also been able to utilize this framework to ask, and answer, some important empirical questions.

1 INTRODUCTION

Simulation is well established as a powerful modeling and analysis technology, and has been most widely adopted for applications such as manufacturing, distribution, and defense. While there have been a number of notable successes in analyzing service systems, the technical requirements for building simulation models, either from scratch or through a simulation modeling environment, have been a significant barrier to entry. Though increasingly easy for IEOR professionals to utilize, a definite barrier remains between what we "experts" can develop and what our customers in the call center (managers and executives in charge of operations) can absorb (see Gulati and Malcolm 2001 and Klungle 1999 for examples).

Call center managers struggle to manage operations that are, at heart, relatively straightforward queuing systems. A number of different elements – including uncertain and time-dependent arrival patterns, variable service times, routing policies, staffing levels, and customer patience factors – conspire to make such systems rather difficult to effectively analyze and manage (see Mehrotra 2003 for additional details about call center simulation modeling).

One way to increase the likelihood of model usage is to allow users to run the simulation through an interface that is already familiar to them, such as a spreadsheet. To address this need, we set out to provide such an interface, showing how to transfer data between a widely used spreadsheet environment, Excel, and a popular discrete-event dynamic simulation software package, Arena (Kelton, Sadowski and Sadowski 2002). Once it had been created, we were able to leverage this interface to address a number of key empirical questions that call center managers often struggle with.

2 THE UNDERLYING QUEUEING SYSTEM AND ASSOCIATED SIMULATION MODEL

Our simulation model can be summarized in standard queuing notation either as X/X/N/K with abandonment, or X/X/N/K+X (Gans, Koole and Mandelbaum 2003), meaning that it represents an elementary waiting line system with inter-arrival and service time distributions that may take a variety of forms, and N identical servers working in parallel. The system capacity K, which may be finite, equals the number of servers plus the maximum queue length, i.e., K = N + MaxQ. Balking occurs when an arriving customer finds MaxQ customers already in line: the arriving customer departs without joining the queue. Furthermore, customers who have joined the queue abandon the line without being served if they have waited more than they care to, as specified by their abandonment time (an attribute assigned in the simulation model upon creation).

Some researchers have developed analytical queuing formulas for this model under fairly restrictive assumptions, e.g., Garnett, Mandelbaum and Reiman (2002) assume that the inter-arrival, service and abandonment time

distributions are all Exponential, and that K is infinite. However, simulation has several advantages for analyzing these types of problems. First of all, the simulation model enables users to specify any of Arena's 12 input distributions, including discrete and continuous versions of an empirical distribution, and any desired queue capacity, thereby enabling the user to focus on representing the underlying business operations as accurately as possible. Another advantage is that the user is not restricted to examining steady-state performance: shorter simulation runs that are significantly influenced by the transient or warm up phase can be made.

Figure 1 shows the corresponding Arena 5.0 model. Except for its abandonment component, most students could build this model a few weeks into a first course in simulation.

Abandonment is modeled by creating a clone of the original customer (in the Separate module) just prior to joining the queue associated with the Process module. This "search clone" is delayed until the customer's abandonment time, at which time the queue is examined. If the original customer is still in line at the time of abandonment, the customer is removed from the line and the abandonment count is updated; otherwise, the search clone is disposed of. An alternative approach to modeling abandonment is given in Saltzman and Mehrotra (2001).

The Outputs data element sequences the performance measures in a specific order so that the Visual Basic (VB) subroutine described below can use them correctly.

3 EXCEL USER INTERFACE AND INTERACTION WITH ARENA MODEL

Our interface is the Excel spreadsheet shown in Figure 2. Here, users may change any of nine input parameters in cells B6:B14 as long as they either fall within permitted bounds (specified in cells D6:E14) or are legitimate Arena probability distributions. After running the simulation, model output is automatically written into cells B19:F30. To keep the simulation details hidden from the user, a key issue is how to transfer data between the spreadsheet user interface and the underlying Arena model. Figure 3 shows a portion of a VB subroutine called "ArenaRun" that reads the user's input data, runs the Arena model, and reports model output back to the spreadsheet. Initially, sections A-C of the code declare variables and set or read in values for many of them. Code in section D starts Arena, opens the particular model file of interest ("XXNKAb.doe") and keeps Arena visible during execution, thus leveraging the animation capabilities of Arena to provide additional insights.

Input is then transferred from the spreadsheet into the Arena simulation model. Some data, such as the length and number of replications, and warm up period (section E), which correspond to the Arena model object variables ReplicationLength, NumberOfReplications, and WarmUpPeriod, respectively, may be directly assigned after conversion into strings. Other input data, however, such as expressions and variables, must be loaded into the model indirectly. For example, to get a particular inter-arrival time distribution into the model, two steps must be taken.

First, within Arena, a unique tag must be placed on the corresponding row of the Expression data module so that it can be identified by the VB code. Specifically, one must right click in the first row of the Expression module (bottom of Figure 1), select the Properties option, and modify the object's default tag to a name (e.g., "InterArrivalDistrib") that will be referenced in the VB code. The second step is shown in section F: locate the desired tag's index with the Find method, set the variable oModule to point to the Arena module associated with this tag, and then set this module's "Value" operand to the user-specified input (IATDist).

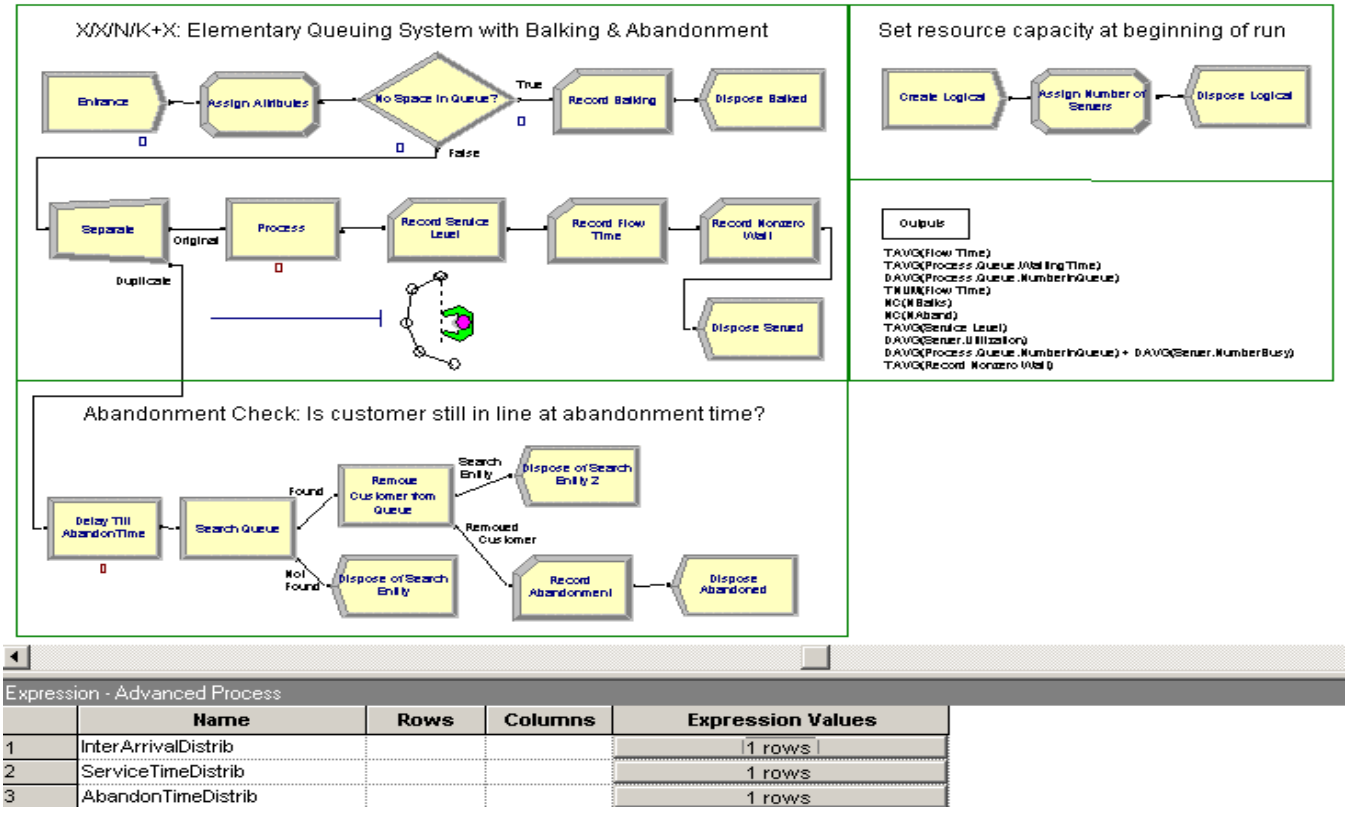
Identical steps are followed for transferring the input service and abandonment time distributions, which are also expressions. The steps for assigning values to *variables* (used for the number of servers, queue capacity and service level target) are similar except that the operand's name is "Initial Value" rather than "Value". For brevity's sake the sections that perform these steps are not shown in Figure 3.

If desired, the Arena model can also be executed in batch mode (no animation) to speed the run, as shown in section G of Figure 3. Once execution has finished, output is transferred back to the spreadsheet via a subroutine called "CopyResults," part of which is shown in Figure 4. For each performance measure, the SIMAN object "OutputAverageAcrossReplications" is used to return the mean value across the replications, along with related objects to provide the half-width of the 95% CI for the mean, and the minimum and maximum values across the replications. After copying output, the model file is closed and the application is exited in the last section (H) of the ArenaRun subroutine.

At this point, it is relatively easy to experiment by *manually* changing input cells and re-running the model by calling the ArenaRun subroutine. However, as an aid to interactive investigation, we found it very valuable to write another subroutine in Visual Basic that *automatically* calls ArenaRun for a series of related scenarios, (e.g., a range of service level target time values), and copies model output to a worksheet. Doing so in a variety of situations yields both quantitative and qualitative insights about the impact that the input has on the performance of an X/X/N/K+X queuing system, as illustrated below.

4 MODEL VALIDATION

To validate this simulation model and the interface, we compared its output to analytical queuing model results for similar sets of input. For example, Garnett, Mandelbaum and Reiman (2002) present analytically derived steady-state



Expression - Advanced Process

	Name	Rows	Columns	Expression Values
1	InterArrivalDistrib			1 rows
2	ServiceTimeDistrib			1 rows
3	AbandonTimeDistrib			1 rows

Figure 1: Arena Model of Call Center Queueing System

	A	B	C	D	E	F
3						
4	*Parameter*			*Lower*	*Upper*	
5	*(Model Input)*	User Input	Units	*Bound*	*Bound*	Comment
6	Interarrival Time Distribution	EXPO(1/48)	minutes	0.01	99999	See (*) below
7	Service Time Distribution	EXPO(1)	minutes	0	99999	See (*) below
8	Number of Servers	50	integer	1	100	
9	Maximum Queue Length	1000	integer	0	99999	
10	Abandonment Time Distribution	EXPO(2)	minutes	0	10000	See (*) below
11	Service Level Target Time	1	minutes	0	99999	
12	Replication Length	22	hours	0	240	
13	Warm Up Time	2	hours	0	20	
14	Number of Replications	20	integer	2	100	
15						
16		Run model now				
17	*Performance Measure*	NReps		95% CI	NReps	NReps
18	*(Model Output)*	Average	Units	Half-Width	Min	Max
19	Ave. Server Utilization (ρ)	93.1%	%	0.1%	92%	94%
20	Ave. Number in Queue (L_q)	3.00	customers	0.09	2.68	3.28
21	Ave. Number in System (L_s)	49.53	customers	0.14	48.94	50.03
22	Ave. Wait Time in Queue (W_q)	0.06	minutes	0.00	0.06	0.07
23	Ave. Wait Time in System (W_s)	1.06	minutes	0.00	1.04	1.07
24	Service Level (given served)	100.0%	%	0.0%	100.0%	100.0%
25	No. Customers Served	55803.2	customers	91.0	55437.0	56227.0
26	No. Customers Who Balk	0.0	customers	0.0	0.0	0.0
27	No. Customers Who Abandon	1785.6	customers	57.7	1564.0	2007.0
28	Fraction Queuing	0.454		0.007	0.429	0.485
29	Fraction Balking	0.000		0.000	0.000	0.000
30	Fraction Abandoning	0.031		0.001	0.027	0.035

Figure 2: Excel Interface to Call Center Simulation Model

```
Sub ArenaRun()

' Runs an Arena model and reports results to Excel worksheet

' A: Declare variables
Dim oArenaApp As Arena.Application
Dim oModel As Arena.Model, oSIMAN As Arena.SIMAN
Dim oModule As Arena.Module
Dim oExcelApp As Excel.Application
Dim oWorkbook As Excel.Workbook
Dim oWorksheet As Excel.Worksheet
Dim nIndex As Long, ModName As String
Dim IATDist As Variant, STDist As Variant, AbTDist As Variant
Dim NServ As Integer, MaxQ As Integer, NReps As Integer
Dim RepLength As Double, SLTarget As Double, WarmUp as Dou-
                                                    ble

' B: Set value of some variables
Set oExcelApp = Excel.Application
Set oWorkbook = Excel.ActiveWorkbook
Set oWorksheet = Excel.ActiveSheet

' C: Read input from Excel worksheet
With oWorksheet
  IATDist = .Range("B6")
  STDist = .Range("B7")
  NServ = .Range("B8")
  MaxQ = .Range("B9")
  AbTDist = .Range("B10")
   SLTarget = .Range("B11")
  RepLength = .Range("B12")
  WarmUp = .Range("B13")
  NReps = .Range("B14")
End With

' D: Start Arena, open model, make Arena active & visible
Set oArenaApp = CreateObject("Arena.Application")
ModName = oWorkbook.Path & "\XXNKAb.doe"
Set oModel = oArenaApp.Models.Open(ModName)
Set oSIMAN = oModel.SIMAN
oArenaApp.Activate
oArenaApp.Visible = True

' E: Place run-time parameters into Arena model
oModel.ReplicationLength = Str(RepLength)
oModel.WarmUpPeriod = Str(WarmUp)
oModel.NumberOfReplications = Str(NReps)

' F: Find Expression Module, row with "InterArrivalDistrib" tag
nIndex = oModel.Modules.Find(smFindTag, "InterArrivalDistrib")
Set oModule = oModel.Modules(nIndex)
oModule.Data("Value") = IATDist

'   … similar steps are taken for transferring the other inputs

' G: Run model in batch mode and send results back to Excel
oModel.BatchMode = True          ' Turn off animation
oModel.QuietMode = True          ' Do not ask final question
oModel.Go (smGoWait)             ' Suspend VB until run ends
Call CopyResults(oSIMAN)

' H: End model run and exit Arena
oModel.End
oArenaApp.Visible = False

End Sub
```

Figure 3: ArenaRun Subroutine (Partial Listing)

```
Sub CopyResults(oSIMAN As Arena.SIMAN)

'   Copies output performance measures from Arena model to Excel
'   For each PM, copy the mean, 95% CI half-width, min & max:

With ActiveSheet
'   Utilization
   .Range("B19")=oSIMAN.OutputAverageAcrossReplications(10)
   .Range("D19")=oSMAN.OutputHalfWidthAcrossReplications(10)
   .Range("E19")=oSIMAN.OutputMinimumAcrossReplications(10)
   .Range("F19")=oSMAN.OutputMaximumAcrossReplications(10)

'   Lq: Average number of customers in line
   .Range("B20")=oSIMAN.OutputAverageAcrossReplications(3)
   .Range("D20")=oSIMAN.OutputHalfWidthAcrossReplications(3)
   .Range("E20")=oSIMAN.OutputMinimumAcrossReplications(3)
   .Range("F20")=oSIMAN.OutputMaximumAcrossReplications(3)

'   … similar steps are taken for other performance measures
End With

End Sub
```

Figure 4: CopyResults Subroutine (Partial Listing)

performance measures for an M/M/N/∞+M queuing system, where the mean arrival and service rates are 48 customers and 1 customer per minute, respectively, while the number of agents varies from 35 to 70. They also compare system performance when there is no abandonment against the case where customers' patience is represented by an exponential distribution with a 2-minute mean.

These scenarios were simulated with our model by specifying an input abandonment time distribution of either 1000 or EXPO(2), along with MaxQ = 1000 (to eliminate balking). The simulation model's results (both here and in subsequent experiments) are based on twenty 22-hour replications, each with a 2-hour warm up period. Output is shown graphically in Figure 5 below.

Comparing the output in Figure 5 to that given in Figure 2 of Garnett, Mandelbaum and Reiman (2002) reveals very close agreement in all cases for the fraction of customers who wait in line for some amount of time. Note the impact that even a relatively small amount of abandonment can make on system performance. For example, when N = 50 servers, 68% of the customers arriving to the M/M/N system wait in line, and the average number in line is 15.1 customers. By contrast, when customer abandonment time follows an EXPO(2) distribution, only 45% of the customers arriving wait in line, and just 3 customers are in line on average (see cell B20 in Figure 2).

The model was also validated against steady state analytical results for a finite capacity multi-server queuing system, i.e., an M/M/N/K queuing system, as presented in Hillier and Lieberman (1990). We computed performance from the formulas and compared them to output from the simulation model for a system with mean arrival and service rates of 95 and 25 customers per minute, respectively, 4 servers, and various system capacities. The results in the

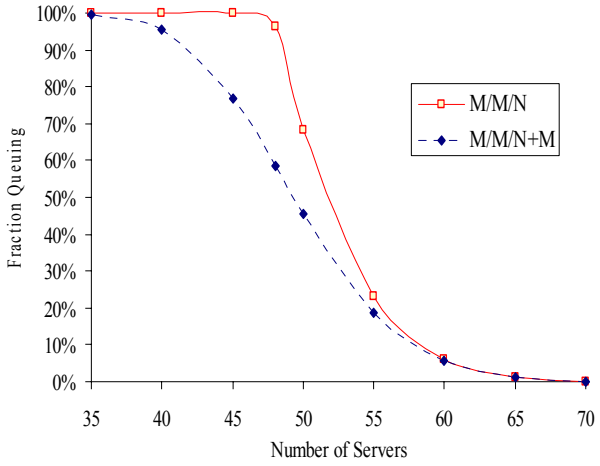


Figure 5: Fraction Queuing vs. Number of Agents

middle two columns of Table 1 again indicate excellent agreement between the steady-state analytical formulas and the simulation model.

One added benefit of the simulation model is that we also acquire output on the fraction of customers who balk (shown in the last column of Table 1). Based on these two validation steps, we are confident that our Arena simulation model is an accurate representation of a queuing system with both balking and abandonment.

Table 1: Analytical Queuing Formulas vs. Simulation Model for M/M/4/K

Maximum Queue Capacity	Ave. Queue Length		% Balking
	Analytical Formula	Simulation Model	Simulation Model
0	0.000	0.000	29.1%
1	0.217	0.217	21.7%
2	0.521	0.521	17.1%
3	0.867	0.867	13.9%
4	1.233	1.233	11.7%
5	1.610	1.609	10.0%
6	1.991	1.990	8.7%
7	2.373	2.370	7.6%
8	2.753	2.745	6.7%
9	3.129	3.124	6.0%
10	3.501	3.496	5.4%

5 EXPERIMENTAL INSIGHTS

Once we had validated both our underlying simulation model and the effectiveness of our user interface in exchanging data with this simulation, we were in a position to address many of the different questions that call center managers routinely pose to us about their operations. Our Excel-Arena "analysis platform" was then used to conduct a number of experiments with the simulation model that yield insight into call center performance under various operating conditions.

5.1 Service Level Achieved vs. SL Target

A classic performance measure within the call center industry is "Service Level," which is defined as the percentage of customers served within some fixed time period (known as "the Service Level Target").

However, in our experience, many call center managers have little or no feeling for the distribution of the customer waiting times, causing them to wonder how the Service Level achieved might vary if the Service Level Target were increased. In conjunction with these dynamics, many call centers also "busy-out" customers by setting some finite maximum queue size (anyone calling when the queue is full receives a busy signal), which is analogous to balking in our modeling framework.

To address these questions, our first experiment examined the relationship between Service Level changes and the Service Level Target time for specific MaxQ levels, *i.e.*, how the percentage of customers served within x minutes changes as the target service time x changes. To demonstrate, we used a busy M/N/3/K system (with input utilization factor $\rho = 0.95$). Figure 6 below shows the results found for the 27 scenarios run.

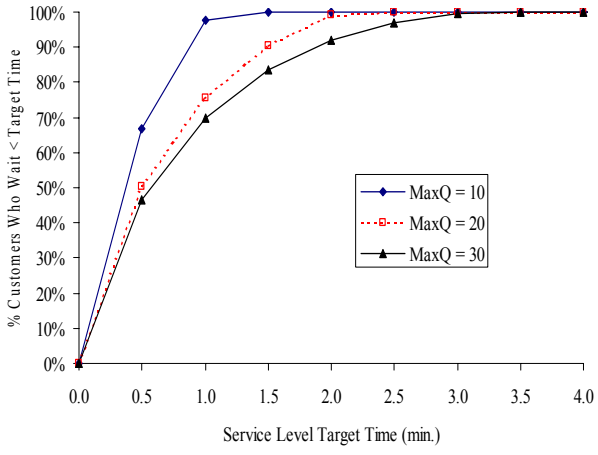


Figure 6: Service Level Achieved vs. SL Target Time

More specifically, when MaxQ = 30, approximately 70% of customers reach a server within 1 minute, while 92% are served within 2 minutes. Other output (not shown) indicates that, when MaxQ = 30, only 0.2% of customers balk upon arriving.

However, when MaxQ = 10, nearly 98% of customers who wait in queue are served within 1 minute while 100% are served within 2 minutes. In this case, though, service levels are elevated because almost 3% of customers balk immediately upon arriving.

5.2 Effect of Staffing Level on Waiting Times and Abandonment Rates

Our second experiment examined how the number of servers affects the average waiting time in line in an

M/M/N/∞+M system in which there are a large number of servers and an essentially unlimited queue capacity. This model, sometimes called an "Erlang A" (Gans, Koole, and Mandelbaum 2003), is a good approximation for many large call centers. However, the analytic model does not provide estimates for customer abandonment, which is a key performance measure of call center operations.

To investigate this, we ran the model with several different mean abandonment times, combined with various numbers of servers, and show the results below in Figures 7 and 8. In these scenarios, the mean arrival and service rates were 400 and 4 customers per hour, respectively, as in Saltzman and Mehrotra (2001).

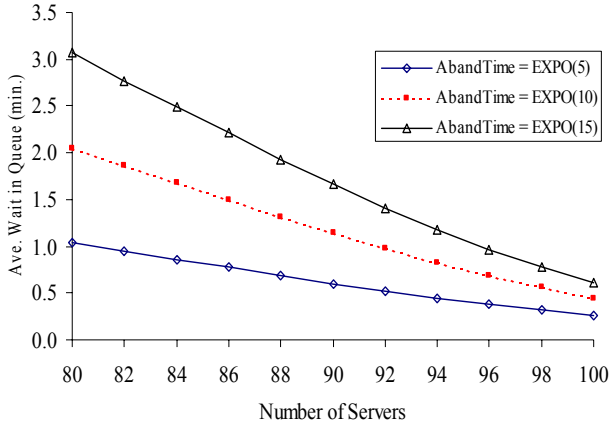


Figure 7: Average Wait vs. N for M/M/N/∞ +M

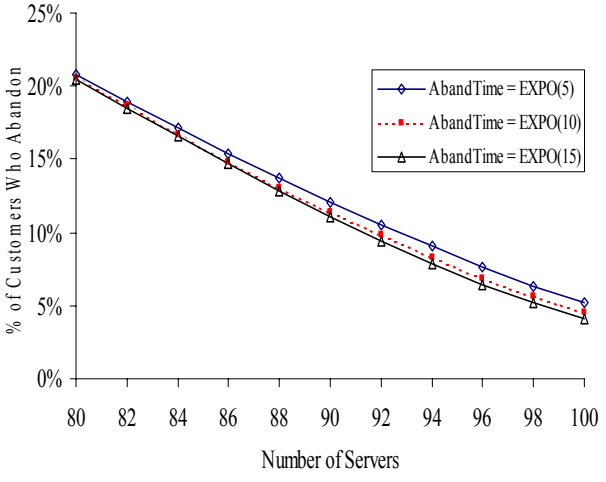


Figure 8: Expected Abandon % vs. N for M/M/N/∞ +M

For each abandonment time distribution, Figure 7 shows that when there are a large number of servers the relationship between average wait in queue and the number of servers is nearly linear. For instance, when customer patience is accurately represented by the EXPO(10) distribution, each additional server above 80 reduces the average wait in line by about 5 seconds, and also lowers the abandonment rate by roughly 0.75%.

Furthermore, 92 servers are required to get the average queuing time down to one minute. On the other hand, if customer patience more accurately fits an EXPO(15) distribution, then 96 servers are required to bring the average queuing time down to one minute. Similarly, Figure 8 shows that the percentage of customers who abandon is also approximately linear in the number of servers N, when N is large. Results such as these could be very useful to managers trying to determine the best way to staff their call center.

5.3 Impact of Higher Interarrival Time Variability

One assumption typically made by modelers of call centers and many other service systems is that the time between customer arrivals is exponentially distributed, which in turn implies a coefficient of variation of 1 for the time between arriving calls. However, many call center managers have told us that in their experience the variance in interarrival times and call arrival volumes is often far greater than this.

Therefore, in our next experiment we considered the impact on system performance of using an interarrival time (IAT) distribution whose coefficient of variation is *twice* that of the exponential. In particular, we compared results from using an EXPO(0.15) IAT distribution to those from using a LOGN(0.15, 0.30) IAT distribution for the M/M/N/∞+M queuing system described in the previous subsection.

Table 2 below shows the impact on the abandonment rate for various numbers of agents N (assuming an EXPO(10) abandonment time distribution). We can see that doubling the coefficient of variation in the time between arrivals increases the abandonment rate by 1-2 percentage points, which translates into an increase in the number of abandoned calls by anywhere from 4% (when N = 80) to 47% (when N = 100). For a call center receiving several thousand calls per day, this increase could amount to a significant jump in lost revenues and/or customer dissatisfaction from abandoned calls, and decidedly worse service than achieved with staffing levels that were based on the assumption of exponentially distributed interarrival times.

Table 2: Impact of Doubling Interarrival Time Variability on the Abandonment Rate

Number of Agents	Interarrival Time Distribution	
	EXPO(0.15)	LOGN(0.15, 0.30)
80	20.6%	21.4%
82	18.6%	19.7%
84	16.7%	17.9%
86	14.9%	16.3%
88	13.1%	14.6%
90	11.4%	13.1%
92	9.8%	11.6%
94	8.3%	10.3%
96	6.9%	9.0%
98	5.6%	7.8%
100	4.5%	6.6%

These results have significant implications, suggesting clearly that the standard assumption of using exponential interarrival times should be revisited by call center modelers. In particular, our simulation results suggest that, at a minimum, call forecasts should use historical data not only to estimate average call volumes per ¼, ½, or one hour interval, but also to estimate the variance in forecasted volumes per period, and that staffing models should take these second moment results into account as well.

5.4 Effect of Mean Time to Abandon on System Performance

Call center managers that we have worked with struggle with questions about customer's tolerance for waiting. Specifically, we have been often been asked "How much of an impact does customers' willingness to wait have on our service levels?"

To explore this question, we examined how mean abandonment time affects the performance of an M/M/10+M queuing system in which the mean arrival and service rates are 9 customers and 1 customer per minute, respectively. Model output shown in Figure 9 indicates that as customer patience increases (higher mean abandonment times), the percentage of customers who abandon decreases while the average wait in line of those who do not abandon also increases. For example, when the mean abandonment time is 6 minutes, 4.2% of customers abandon the line, while the remaining customers wait in line for an average of 0.25 minutes. If customer patience doubles, i.e., the mean abandonment time is 12 minutes, only 2.8% of customers abandon the line (1/3 less than previously), while the remaining customers wait in line for an average of 0.34 minutes (5 seconds more than previously). Since there are costs are associated with both abandonment and waiting in line, the graph brings up the interesting question of whether or not there exists some optimal degree of customer patience that best balances these two measures of performance.

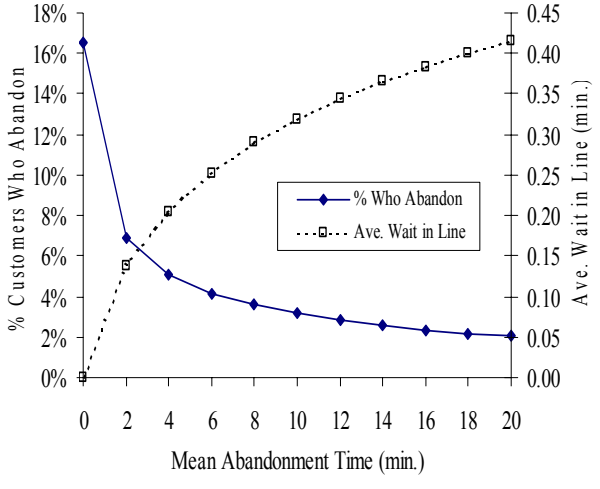


Figure 9: Effect of Mean Abandonment Time on the Performance of an M/M/10+M System

6 CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

In conducting the research described in this paper, we have had two main objectives. First of all, we have sought to create a manager-friendly environment for rapid simulation modeling and analysis of call center operations. In addition, we have made use of this modeling and analysis environment to address several initial empirical questions that call center managers have brought to our attention.

From here, this research can progress in many different directions. The platform (both the spreadsheet interface and the simulation model) can, of course, be modified or extended easily to incorporate multiple queues, multiple customer types, and/or time-of-day and day-of-week dependent call arrival and service time distributions.

We could also explore any number of additional empirical questions that are generated by call center managers on an on-going basis, including:

- How do these results vary under different load conditions?
- What is the impact of increased service time variability on waiting time and abandonment?
- If agents are cross-trained to make outbound calls, what is the impact of different outbound calling policies on inbound service levels?
- What is the risk associated with a particular staffing level? That is, given a particular forecast distribution and some number of agents scheduled, what is the likelihood that a particular service level objective (e.g., 90% of calls handled within 60 seconds) is achieved?

Finally, we can utilize results from our simulation platform as inputs into planning and optimization models (for example, see Saltzman 2004).

REFERENCES

Gans, N., G. Koole and A. Mandelbaum. 2003, Telephone Call Centers: Tutorial, Review and Research Prospects. *Manufacturing and Service Operations Management*, 5 (2): 79-142.

Garnett, O., A. Mandelbaum and M. Reiman. 2002. Designing a Call Center with Impatient Customers. *Manufacturing and Service Operations Management*, 4(3): 208-227.

Gulati, S. and S. A. Malcolm. 2001. Call Center Scheduling Technology Evaluation Using Simulation, In *Proceedings of the 2001 Winter Simulation Conference*, ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer. 1438-1442. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hillier, F. S. and G. J. Lieberman. 1990. *Introduction to Stochastic Models in Operations Research*. New York: McGraw-Hill.

Kelton, W. D., R. P. Sadowski and D. A. Sadowski. 2002. *Simulation with Arena,* 2nd Edition, New York: McGraw-Hill.

Klungle, R. Simulation of a Claims Call Center: A Success and a Failure. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Harrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans. 1648-1653. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Mehrotra, V. 2003. Call Center Simulation Modeling: Methods, Challenges, and Opportunities. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 135-143. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Saltzman, R. M. and V. Mehrotra. 2001. A Call Center Uses Simulation to Drive Strategic Change. *Interfaces*, 31(3): 87-101.

Saltzman, R. M. 2004. A Hybrid Approach to Minimizing the Total Costs of Staffing a Call Center. Currently under review with *International Journal of Operations and Quantitative Management*.

AUTHOR BIOGRAPHIES

ROBERT M. SALTZMAN is a Professor in the Decision Sciences Department at San Francisco State University where he teaches courses in Simulation, Operations Management, and Statistics. He received his Ph.D. in Operations Research from Stanford University, and his B.S. degree in Applied Mathematics from Brown University. Rob's main research interests are in decision making via animated simulation and optimization modeling, as well as in effective techniques for mathematical education. His e-mail address is <saltzman@sfsu.edu>.

VIJAY MEHROTRA is an Assistant Professor in the College of Business at San Francisco State University. Prior to joining the SFSU faculty in the fall of 2003, Vijay had been a Vice President at Blue Pumpkin Software and CEO of Onward Inc. Vijay holds a B.A. degree in Mathematics, Economics, and Statistics from St. Olaf College and an M.S. and Ph.D. in Operations Research from Stanford University. He writes a regular column in *OR/MS Today* entitled "Was It Something I Said?" Vijay's research interests include applications of stochastic processes and optimization, queuing networks, and the adoption of models and information technology. His e-mail address is <vmehrotra@sfsu.edu>.