Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

ACE: A DECISION TOOL FOR RESTAURANT MANAGERS

Todd C. Whyte
Darrell W. Starks

I•SYS Engineering
14419 Broad Oak Place
Louisville, Kentucky 40245-5136, U.S.A.

ABSTRACT

A simulation based decision support tool called ACE has been developed for Pepsico Restaurants International (PRI) allowing restaurant managers to quickly and accurately analyze a given restaurant. ACE gives the restaurant manager a tool to evaluate the customer point of contact within a given restaurant. This simulation based tool uses Arena for the model and Lotus 123 for the interface. PRI has installed this tool in two of the six international regions and will by early 1997 have it in all of the international regions.

1 INTRODUCTION

PRI needed a decision tool which could be used by the KFC restaurant managers. This tool had to be user friendly and at the same time had to produce results that could be used by a restaurant manager to determine the impact of changes in the customer arrivals upon the service time for the customers. The challenge was to produce a model that was accurate for a variety of restaurants and an interface that would allow a non operations research individual to parameterize the model and use the model results to analyze a given restaurant.

2 OBJECTIVE

The primary objective of this project was to develop a decision tool for the KFC restaurant manager which would be accurate for a given restaurant and would be useable by the restaurant manager to make specific decisions for that restaurant. In particular, a manager would be able to interface with the model in a user friendly environment, parameterize the model, run the model, and see the results in a timely manner. The primary decision to be made would be the number of restaurant personnel required to achieve a given service level for the customers, in particular the percent of customers that achieved this service level.

3 SYSTEM DESCRIPTION

The system to be modeled included the arrival of customers to the restaurant either at the inside front counter or at the drive through window. The front counter and drive through personnel performed the following tasks: take the order, pack the order, exchange money and hand-off the order.

Three different restaurant setups were considered at the front counter and at the drive through. At the front counter there could be only cashiers, cashiers with packers, or a pick system. In the case of the cashiers only the cashiers performed all the above tasks. In the case of the cashiers with packers the cashiers performed the take order and money exchange tasks while the packers performed the packing and hand-off tasks. In this case the ratio of cashiers to packer could be one to one or two to one respectively. The pick system case was the same as the cashiers only except instead of packing, the items were prepacked, requiring less time than the packing operation in the first case. For the drive through window the first case had only one cashier who performed all the tasks. In the second case a cashier performed all the tasks except the packing which was performed by a packer. Finally in the third case called split cashiering, a cashier took the order, a packer did the packing of the order, and a third person did the money exchange and hand-off. For each individual restaurant the front counter and drive through window can be configured in any of the three ways.

4 ARENA MODEL DESCRIPTION

The models were developed in the Arena modeling framework. There are six basic models: a cashiers only model with a straight counter configuration and an angled counter configuration, a cashiers and packers model with a straight counter configuration and an angled counter configuration, and a pick model with a straight counter configuration and an angled counter configuration. Figures 1 and 2 illustrate the two different counter configurations.

The model looked only at the front counter and drive through operations with the customers. The model assumed that product is always available. Each customer transaction consisted of one to five orders based upon historical probabilities. Because packing times depend upon the type of order, eight meal type categories were developed:
small meal, large meal, small snack, large snack, sandwich, strips/wings, pot pie, and other. Depending upon the region all may or may not be used. The probability for each category was dependent upon the menu mix which is input by the user.

Each model consists of two primary modules: the front counter module and the drive through module. Two other modules read the model parameters from a file created by the interface and write out specific output values to a file to be read by the interface to create the user output.

The front counter module models the arrival of customers to the front counter, the placing of the order, the money exchange, the packing(picking) of the order, and the hand-off of the order. The times for each of the above tasks were developed from time studies performed at various restaurants from various regions. Some of the task times have been found to be regionally dependent. The user inputs the following data to parameterize the front counter: length of peak period, number of arrivals per ten minute interval, type of model, number of registers, and capacity of queuing area.

The drive through module models the arrival of cars to the menuboard where the customer places the order, the movement of the

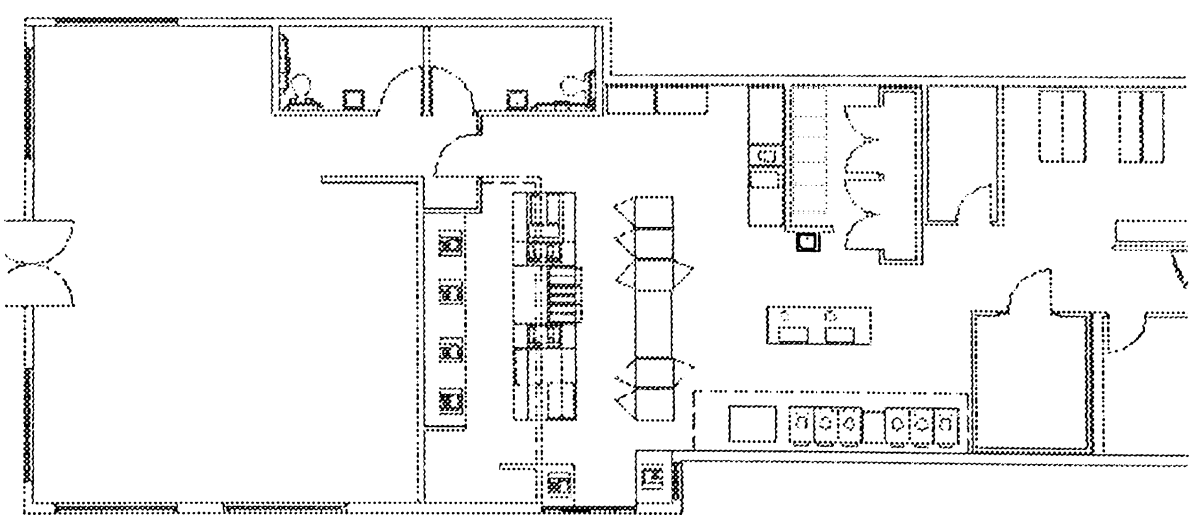

Figure 1 - Straight Front Counter

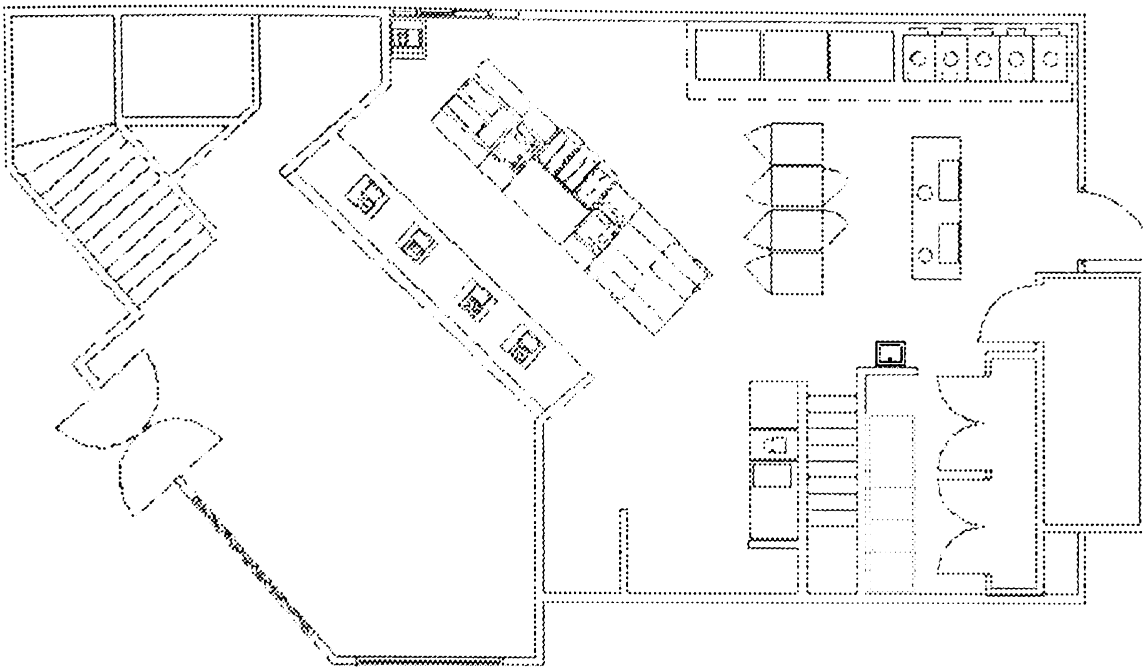

Figure 2 - Angled Front Counter

cars to the drive through window, the packing of the order, the money exchange, the hand-off of the order, and the movement of the car out of the system. As in the front counter logic the times for the above tasks were developed from time studies and some have been found to be regionally dependent. The user inputs the following data to parameterize the drive through window: length of peak period, number of arrivals per ten minute interval, type of model, capacity of queue before the menuboard, and the capacity of the queue between the menuboard and the window.

The model animation was easily achieved by importing a cad drawing as the background and building the model around this drawing. The user has the choice of either animating or not. The animation is also illustrated in Figures 1 and 2.

5 USER INTERFACE

As stated previously the intended user of the simulation will be restaurant managers and other field personnel with little or no knowledge of simulation modeling. Thus the people who could benefit from simulation should be able to perform a simulation experiment when and where it is needed. In lieu of developing an expensive training program to instruct personnel on how to load the parameters, execute and evaluate simulations, our philosophy was to develop an interface to reduce the skill level required to use the models.

All of the intended users were assumed to have a working knowledge with various Windows applications, through normal job routines (e.g., corporate reports, communications, etc.) The software that every user can access is Lotus SmartSuite, in particular the Lotus 123 package. Thus, if an interface could be developed with Windows constructs (pop-up windows, dialog boxes, list boxes, etc.) then the conjecture was that implementation and acceptance in the field should be greatly enhanced. The language that was selected for the initial development of the interface was the Lotus macro programming language.

A general purpose programming language was initially proposed for the development of the interface; however, the spreadsheet macro was ultimately selected by the design team. The advantages of the macro language compared to a general purpose language were rapid development, lack of additional software costs, and increase user acceptance (i.e., the application will be perceived as "just another spreadsheet"). The development of the interface using macro approach proved to be more difficult than originally thought due in part to several limitations in the macro language itself. However, despite the few difficulties encountered in the design phase, an effective interface was developed and implemented.

The interface is driven by the user selecting one of the various buttons that call a particular routine. The input screens (a total of two exist) are composed of approximately 16 buttons and has been illustrated in Figure 3.

To begin an input session the user simply clicks on the following buttons:

- **Peak Duration** - activates a list box that allows the user to specify the duration of the simulation (from 10 minutes to three hours in 10 minute increments).
- **Menu Mix** - reveals a worksheet that allows the user to input actual menu items sold
- **Ticket Average** - average transaction amount used to calculate potential income based on the number of balks.
- **Registers** - activates a list box that allows a user to select the number of registers on the front counter.
- **Drive-through** - activates a dialog box to specify the presence of a drive-through window.
- **Worker Assignment** - activates a dialog box that is used to specify customer service worker assignments (a separate button is used for both front counter and drive-through).
- **Self - Serve Drinks** - activate a dialog box to specify yes/no
- **Queue Lengths** - activates text boxes that allow the user to specify queue capacities for front counter, drive-through menuboard and drive-through window.
- **Arrival Patterns** - reveals a worksheet that is

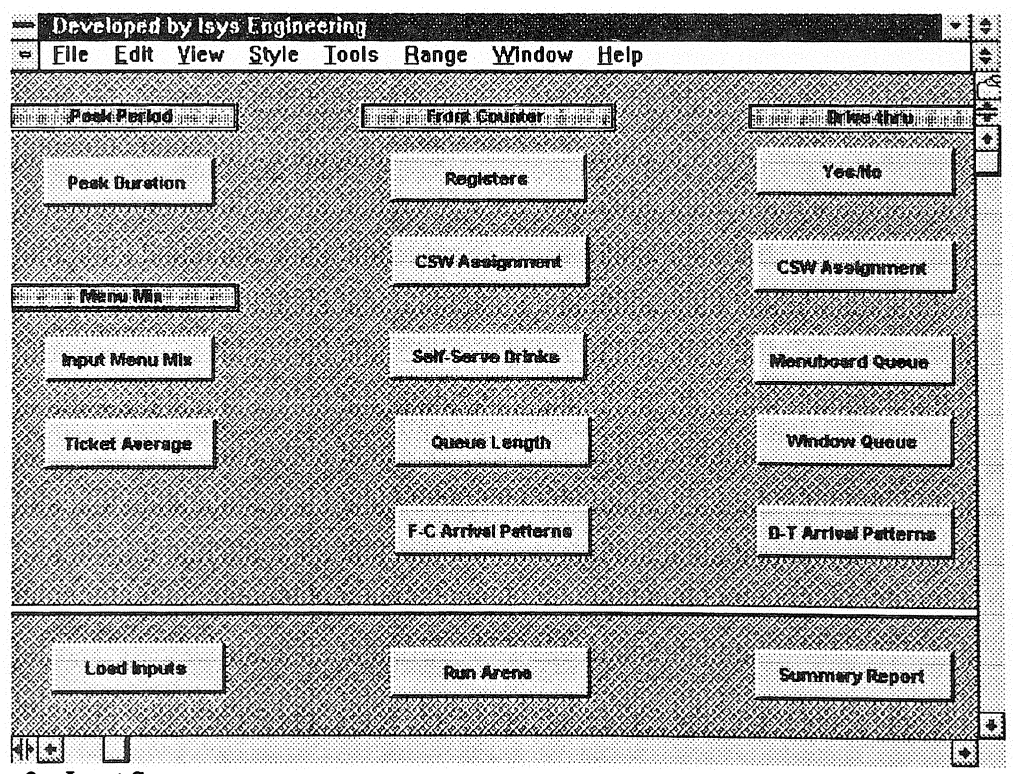

Figure 3 - Input Screen

used to input the number of customers that arrived in 10 minute increments, for the duration of the simulation.

- **Load Inputs** - takes the inputs that were input and writes to an ASCII data file.
- **Run Arena** - based upon the inputs, the button selects the proper model and launches Arena .
- **Summary Report** - used to review the statistics after Arena has been executed

Once Arena has been successfully launched with the run Arena button, the user simply clicks on the appropriate Arena control to execute the model. After the simulation has run the specified number of runs, the simulation session terminates and the user can exit Arena. The interface resumes control and the input screen (Figure 2) reappears, permitting the user to select the Summary Report button to expose the customer statistics that were collected, as illustrated by Figure 4.

In addition to the customer statistics, server statistics are collected (by position description) as illustrated in Figure 5. An additional section was created on the output report to attempt to capture lost revenue. The total number of balks are multiplied by the ticket average and then annualized. This monetary measure serves as an additional performance measure for performing a what-if style of analysis. To facilitate this iterative style of analysis, a button has been created to input a new session (e.g., alter menu mix, customer service assignments, number of registers, etc.). This iterative analysis gives the restaurant manager a tool to determine staffing levels and worker assignments to better serve the customer.

6 Conclusion

The initial implementations have been successful, with minimal amount of training time (usually much less than an eight hour day). The user

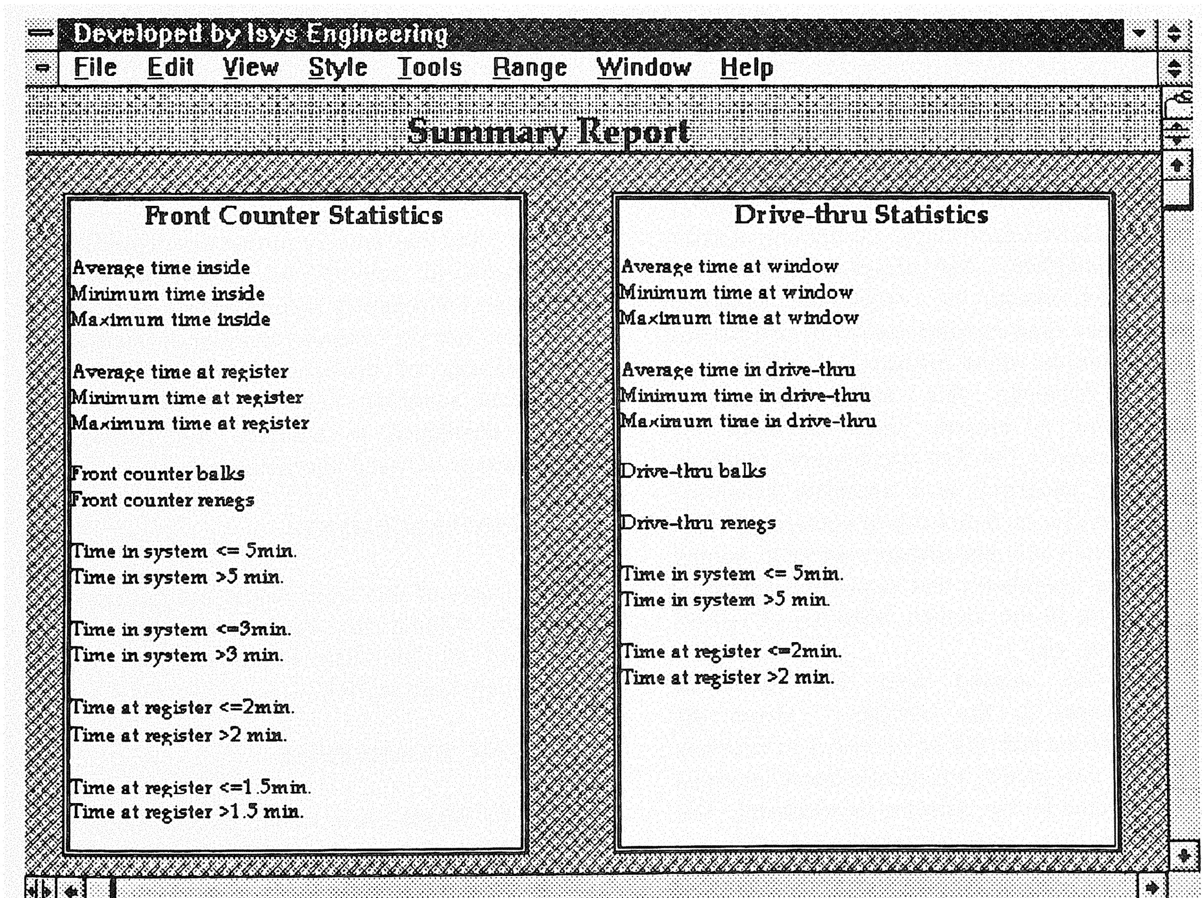

Figure 4 - Summary Report (customer statistics)

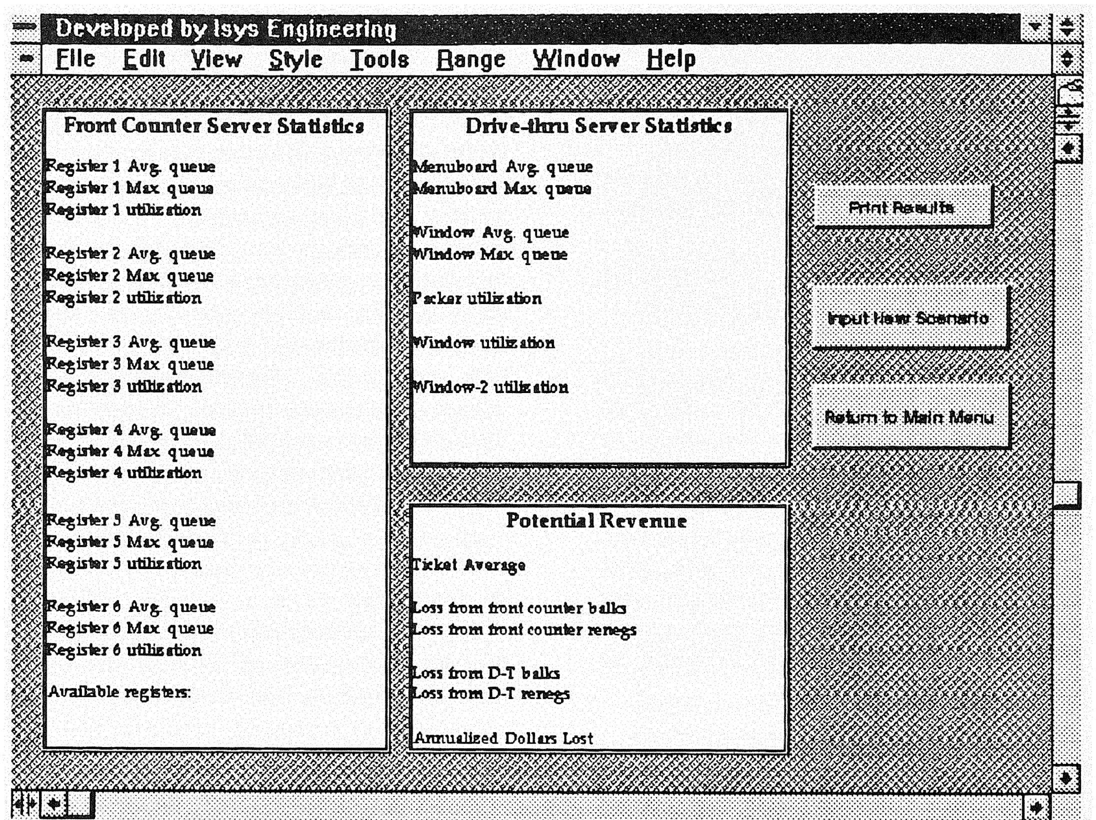

Figure 5 - Summary Report (server statistics)

responses have been extremely favorable, and on occasion have helped shaped the final appearance of the interface. The design team also was privileged enough to experience complete acceptance (and in some instances enthusiasm) throughout the senior management of PRI.

Although this initial effort was considered a success, there is room for improvement. The first improvement which is currently underway is a complete restaurant simulation for an entire day of operations. This added detail will encompass aspects to normal restaurant operations; that is, product shortages, congestion in the kitchen, blocking of critical equipment, etc.

The second area for continuing improvement is the interface. Given the modifications that will be necessary to interface with the new model, a general purpose language was selected for the interface development. The language that was ultimately selected was VisualBasic for Windows 95. Despite changing the language, the interface should be similar to the current version so any retraining will be minimal.

The final area for further development is the statistical analysis of the output data. Currently the summary report consists of means and does not yet compute confidence intervals, or measures of dispersion. In addition to enhanced summary statistics, a procedure is being developed to compare the statistical significance between alternative scenarios.

ACKNOWLEDGEMENTS

The success of this work would not have been possible without the funding and guidance provided by Mr. Ray Lantz and Mr. Ron McGinley, both of PRI.

AUTHOR BIOGRAPHIES

Todd C. Whyte is a senior project engineer with I•SYS Engineering of Louisville and is currently a doctoral candidate at the University of Louisville, Department of Industrial Engineering. His research interests include

simulation, and facilities layout

Darrell W. Starks is a senior consultant with I•SYS Engineering of Louisville and is currently a doctoral candidate at the University of Louisville. Mr. Starks has worked extensively in the application of simulation technology to real world applications while at McDonald-Douglas Corporation, Pritsker Corporation, Biles & Associates, and I•SYS Corporation. Application areas include manufacturing, transportation, logistics, hospitals, and restaurants. Mr. Starks research interests are in the area of interfacing simulation to optimization techniques for real world application.