Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

ENHANCING SIMULATION MODELS FOR EMERGENCY ROOMS USING VBA

Adriana M. Alvarez
Martha A. Centeno

Department of Industrial and Systems Engineering
Florida International University
Miami, FL 33199, U.S.A.

ABSTRACT

Increased pressure to control costs and increased competition has prompted health care managers to look for tools to effectively operate their institutions. Managers are increasingly using simulation modeling to understand their processes. A simulation model of an ER has been enhanced with VBA routines, so that it can use real world data. The simulation model has been complemented with a series of decision-making routines. These routines use a hierarchical approach to organize the various scenarios under which the model may run and to partially reconfigure the ARENA model at run time.

1 INTRODUCTION

Increased pressure to control cost as well as increased competition has prompted health care managers to look for tools to effectively operate their institutions. Emergency rooms (ER) are a vital part of a hospital since they usually are the first points of contact between the community and the institution. According to McGuire (1994), most of the patients that are admitted to a hospital come through the Emergency Room. Therefore, it is vital to pay close attention to decisions made with respect to its operations. These decisions include policies regarding the number of physicians, scheduling of nurses and clerks, physical layout, and the adoption of new policies such as the implementation of a fast-track system.

ER managers have turned to computer simulation to aid them in their decision-making process. In the past 30 years, simulation modeling has been progressively used to support operational decision-making in the health care industry. However, managers need to make decisions using information that exists in other types of information systems. Hence, they need the means of using such information in conjunction with simulation. By using simulation in health care, managers have been able to evaluate "what-if" scenarios without actually having to interrupt the daily operations of the facility. We present a simulation model of an ER system, which has been enhanced to function as a Decision Support System.

The second section of this paper describes how simulation has been used in healthcare. Section three presents a framework for enhancing simulation models.

2 SIMULATION IN THE HEALTHCARE

The healthcare industry has adopted simulation very slowly compared to other industries such as manufacturing. One of the principal reasons is management's reluctance to reduce complex processes to a model representation. Another obstacle is their resistance to change, especially when such change requires the use of unfamiliar technical and statistical concepts.

In healthcare systems, simulation has been often used to address problems in the areas of facility design and development, staffing levels and scheduling, new policy evaluation, scheduling of patient admissions, and disease and epidemic control. The most common objectives of these studies have included the reduction of patient's time in the system (ER, outpatient clinics), improvement on customer service, better resource utilization, and reduction of operating costs.

McGuire (1994) describes the use of simulation to test several operating alternatives in a SunHealth Alliance Hospital with the objective of reducing length of stay for patients visiting the ER. His work identified which alternatives had the greatest impact on patient's length of stay and which ones had no significant impact. Freedman (1994) used discrete simulation to study the effects of changing operations on average length of stay in an ER at two different hospitals. At St. Joseph Hospital near Baltimore, a change in the communication system between nurses and housekeeping staff was simulated. This new computerized system reduced the admission delay of patients from the ER to the hospital, which also reduced the patient's average length of stay.

At the Washington Adventist Hospital, simulation was employed to evaluate an expansion in the number of beds in the ER, which resulted in a reduction of .6 hours for average length of stay. At the Peninsula Regional Medical Center in Salisbury, Md., Kirtland et al. (1995) used simulation to improve performance by reducing the patient's time in the system and determining the appropriate staffing levels. They studied eleven different alternatives, which resulted in a reduction of thirty-eight minutes on the average.

Kittell and Pallin (1992) describe the development of a simulation study at Mercy Hospital in Miami, Fl. Their study evaluated several alternatives with the intent of getting more patients through the ER while making more efficient use of the department's resources and still provide good quality services. The study showed that a 50% reduction of resources could be accomplished by implementing a fast-track policy without risking the quality of service provided to its customers. Similarly, Centeno et al. (1995) utilized simulation to evaluate the effect on length of stay in the ER at Mercy Hospital if they were to implement a fast track lane. Their study resulted in a 25% time reduction for patients visiting the ER.

At St. Lukes Hospital ER, simulation was used to evaluate facility design together with operational design to assess the impact of such designs on patient care. Simulation also aided them in evaluating the effects of a new system for the triage process called The Triage Plus system. The model showed that by implementing this system twenty percent of patients would be kept out of the ER. In addition, a reduction in length of stay of as much as forty percent was obtained (Lange 1997). In 1997, Durham Regional Hospital in North Carolina used simulation to model the impact of an Express Service Area on their other outpatient areas. The simulation model helped to see which changes could be made before opening this area. It also helped in exploring modifications to some of the already proposed changes to realize savings of approximately $150,000 (Lowder 1997).

3 ENHANCING ER MODELS

A framework to enhance ER models was conceptualized as in Figure 1. This framework consists of four basic functional sections: inputs, processing engines, outputs and results, and the user interface. Each section is subdivided into different modules. Some of the components work sequentially, others interact with each other, and there are others that work in parallel, independently.

An ER facility is typically part of a hospital; thus, the required data may be available in the hospital's database, or it may be confined to paper reports, or it may be residing in a database of an external information system supplier to the hospital. Depending on where the source of the data is, different permits and paperwork may take place. Once all permits have been obtained, the necessary *communication protocols* may be devised. These protocols may include on-line and off-line data extraction from the hospital's database. From Figure 1, it can be seen that a simulation model is at the heart of the framework. The model is surrounded by components that feed data to it, modify it, and take simulated data out of it.

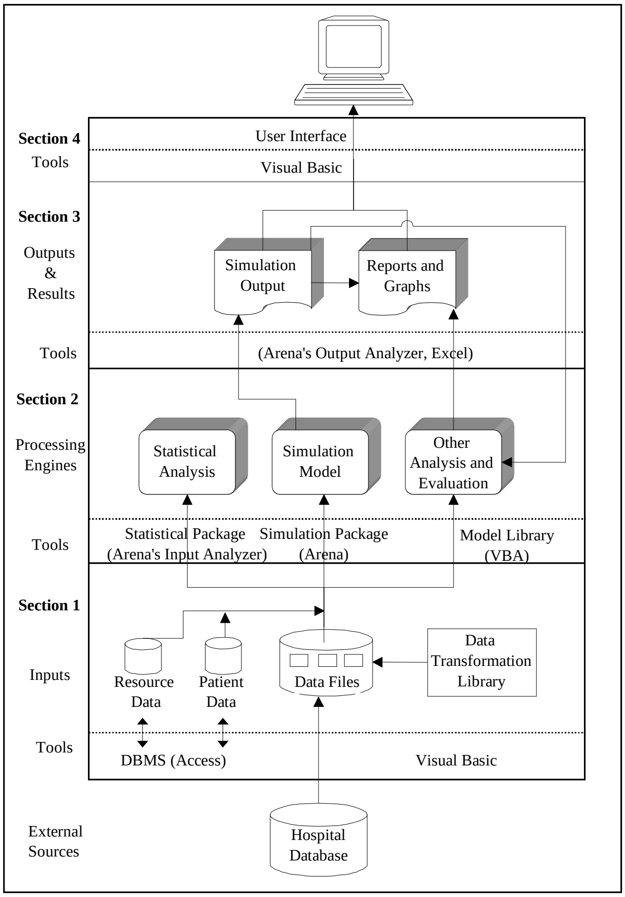


Figure 1: Proposed SB-DSS Framework

The model used for this work is relatively general. Patients arrive to the facility by ambulance or car. The highest priority is assigned to patients arriving in ambulance or rescue (priority 1) and critical patients arriving by car (priority 2). A bed must be available for priority 1 patients arriving by ambulance; otherwise, they are taken to another institution. Priority 2 patients arriving by their own means must also be placed on a bed upon arrival. The patients condition is assessed by a triage nurse, who determines the priority of the user, based on their vital signs as well as symptoms. Patients arriving to the facility with a non life-threatening condition (priority 3, 4, and 5) go through the registration process next. Afterwards, the, patient is sent to the waiting area and waits until a regular bed or a fast-track bed is available. Fast track beds vary in the hours of operation. In some instances, they operate 24 hors a day, in other cases they operate under a fixed schedule. Patients are sent to the fast-track bed whenever all other legal beds in the ER are busy. When a patient is called to a bed, he/she is seen by a nurse and by a doctor and receives adequate treatment. Afterwards a decision is made on whether the patient should be admitted to the hospital or should be discharged from the facility.

To develop a prototype of this framework we chose the decision of *Temporarily Closing the ER* within the *Facility Operating Policies* category. Managers must decide if they continue or stop accepting patients with life threatening conditions that are brought to the facility by ambulance. In general, the ER is considered to be full when almost all beds are occupied, or there are no more beds with a monitor, or the waiting areas are full. If these conditions are suspected to extend over a significant period of time, the manager is forced to bypass the facility to properly service all types of patients. Once the decision is made to close the facility, the proper authorities are informed to divert critical patients to other institutions.

4 FEEDING DATA TO THE MODEL

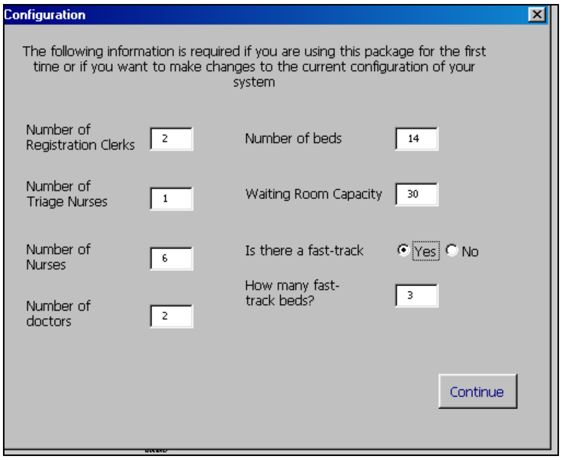
The function of the **inputs section** is to hold or be able to acquire the data required by the model. Thus, its main components are the *ER database*, a *library of data transformation routines*, and *data files*.

Each category of decisions requires a different set of data. The data may be available from the hospital information system department, or it may reside in a database of an external information system supplier to the hospital, or it may be in the ER databases. These databases are usually located at separate places and are stored on different hardware and software platforms (Forgionne and Kohli, 1996). Consequently, the data must be transformed into a proper format through routines compiled in a *data transformation library*. We developed a library of routines written in Visual Basic for Applications (VBA), which are capable of transforming the extracted data from the database and the user into the proper format required by the model. Data from the user is acquired through a user interface written in VBA.

An example of these routines is the one for processing data files that contain values such as time between arrivals, type of patients, service times, and so forth. Figure 2 shows the VBA code to control the statistical package that comes with ARENA and transferring to the model the probability function yielded by it.

```
Sub OpenInput()
'---------------------------------------------
Dim MyAppl, Pathfile
Pathfile = "P:\HOME\" &
information.Fileslist.Value
    MyAppl = Shell("C:\ARENA\INPUT.EXE", 1)
    SendKeys "%{F}", True     'Show file menu
    SendKeys "N", True        'Open new file
    SendKeys "%{F}", True     'Show file menu
    SendKeys "F", True        'Get data file
    SendKeys "U", True   'Use existing data
    SendKeys Pathfile   'Copy name of file
    SendKeys "%{O}", True  'Open the file
  SendKeys "%{T 2}{ENTER}", True 'Show Fit menu
    SendKeys "A", True       'Fit distributions
   SendKeys "%{E}", True       'Show edit menu
    SendKeys "E", True 'Copy distribution
  SendKeys "%{F4}", True    'Close inp. analyzer
    SendKeys "N", True      'Do not save file
If information.distribution.CanPaste=False Then
        MsgBox "Can't paste distribution"
 Else
        information.Label4.Visible = True
        information.distribution.Text = ""
        information.distribution.Visible = True
        information.distribution.Paste
End If
    information.Button1.Enabled = False
    information.PreviousBtn.Enabled = False
    information.Button2.Enabled = True
End Sub
```

Figure 2: VBA Routine to Trigger Input Analyzer


Configuration

The following information is required if you are using this package for the first time or if you want to make changes to the current configuration of your system

Number of Registration Clerks: 2

Number of Triage Nurses: 1

Number of Nurses: 6

Number of doctors: 2

Number of beds: 14

Waiting Room Capacity: 30

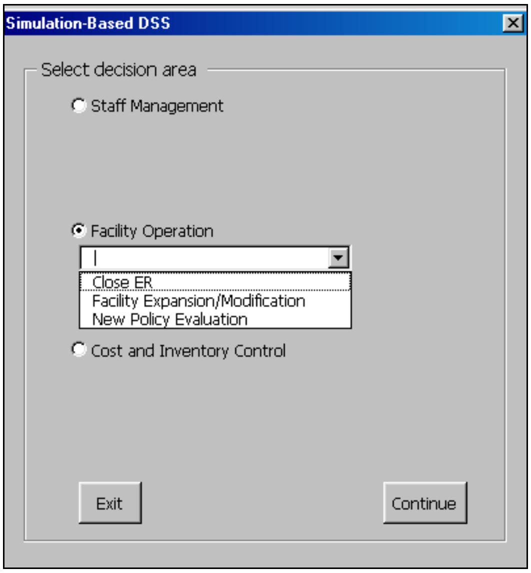
Is there a fast-track ● Yes ○ No

How many fast-track beds? 3

Figure 3: Configuration Form

The data acquisition process is given in Figure 5. After pressing the run begin button in ARENA, the *Welcome* form is loaded into the screen. The user is given the option of reconfiguring the ER system. If the user chooses to configure the system then the *Configuration* form is shown (Figure 3). Afterwards, the *Main Menu* form of the DSS is shown (Figure 4). The main menu presents three categories of ER decisions: human resources, facility operation, and cost and inventory control. Within each category, the user can select among different decisions. All of these forms are handled with a series of VBA routines similar to the one given in Figure 6.

The user interface together with the VBA routines collect information with respect to the types of analysis desired by the user and required by the model.



○ Staff Management

● Facility Operation

○ Cost and Inventory Control

Figure 4: Main Menu Form

5 MODIFYING MODEL AT RUN-TIME

VBA within the ARENA/SIMAN environment presents a series of events which can be programmed to retrieve information from the model at the beginning of a simulation run, while at the same time it allows to change the parameters of the simulation modules at run time. These events include: RunBegin, RunEnd, RunEndReplication, RunEndSimulation, and RunPause.

An example of modifying the model at run-time is given in Figure 7. In this particular scenario, at the end of every replication we needed to modify the value of one of the entries in the VARIABLES element. Hence, we used the RunEndReplication event to modify the behavior of the model while it was executing. This modification actually changes the parameters under which the model runs the next replication.

The simulation length of a replication in each run varies every time the user needs to evaluate this decision. In the *Close ER* form, the user needs to specify the time in the future for when he/she wants to evaluate if the facility should be closed. Next, the user is asked to enter whether the data used for the simulation run will be historical data or the data for that same day. If the user chooses to use the same day's data, then the time of the day for which the system is to start collecting data needs to be entered. With the information collected in this form, a VBA routine calculates the proper simulation length and updates the REPLICATE element in the simulation model.

For the case that the user selects to use historical data, the system will take him/her into the *AskOutput* form, which will ask information on the parameters of interest to the user. By using historical data, the simulation model will run using data distributions that have been already calculated and stored in the appropriate modules within the ARENA model. If the user chooses to use the day's data then he/she is taken to the *Information* form, where the user specifies the data files where the data is stored. This means the user needs to specify if the distribution will be calculated for the inter-arrival times or service times by choosing the proper file. Via a VBA routine, the Input Analyzer application is launched. The best distribution for the data is determined and its expression is then transferred to the appropriate modules in the simulation model (CREATE module for inter-arrival times, DELAY modules for service times).

Afterwards the user is taken to the *Ask Output* form. The *Ask Output* form also gives the choice of using performing the analysis based on averages or on percentiles. For the prototype, the user is only given the choice of using averages: however, for a full-scale system the option of the percentiles would need to be developed. It is important to include this option because during the surveying process it was found that nowadays ER managers are interested in analyzing the performance of their facility based on percentiles. Once all of the information required in this form is collected the inputs section is done with its job and the experimentation part of the system begins.

6 PROCESSING AND MODELING ENGINES

Even though ER systems are considered to be non-terminating systems, the simulation experimentation part of the SB-DSS for the decision of closing the ER was treated as a terminating system. The reason being that to evaluate this decision a specific time is set as the ending time of the operations of the ER system. The ending time is equivalent to the time at which the manager is considering closing the facility. At this set time the statistics of the system are collected and evaluated without considering what happens to the system afterwards. For a terminating system, the simulation is run for an initial number (*n*) of replications. Then the half width (***h***) of a (1-α)% confidence interval, around the mean of the measures of interest, is computed and compared to the half width desired by the user ($\boldsymbol{h}^*$). If *n* replications are enough, then the model run is ended and the results are shown to the user. On the other hand, if *n* replications are not enough to meet the desired half width, then the number of replications required to obtain $\boldsymbol{h}^*$ are calculated ($\boldsymbol{n}^*$) and the model is run once again for that many replications. However, if $\boldsymbol{n}^*$ is too large, the number of replications is truncated to 100. This is done because after 100 replications the value of ***h*** does not get significantly closer to $\boldsymbol{h}^*$, but it would require a higher execution time, which is of utmost importance when deciding what actions to take in the immediate future.

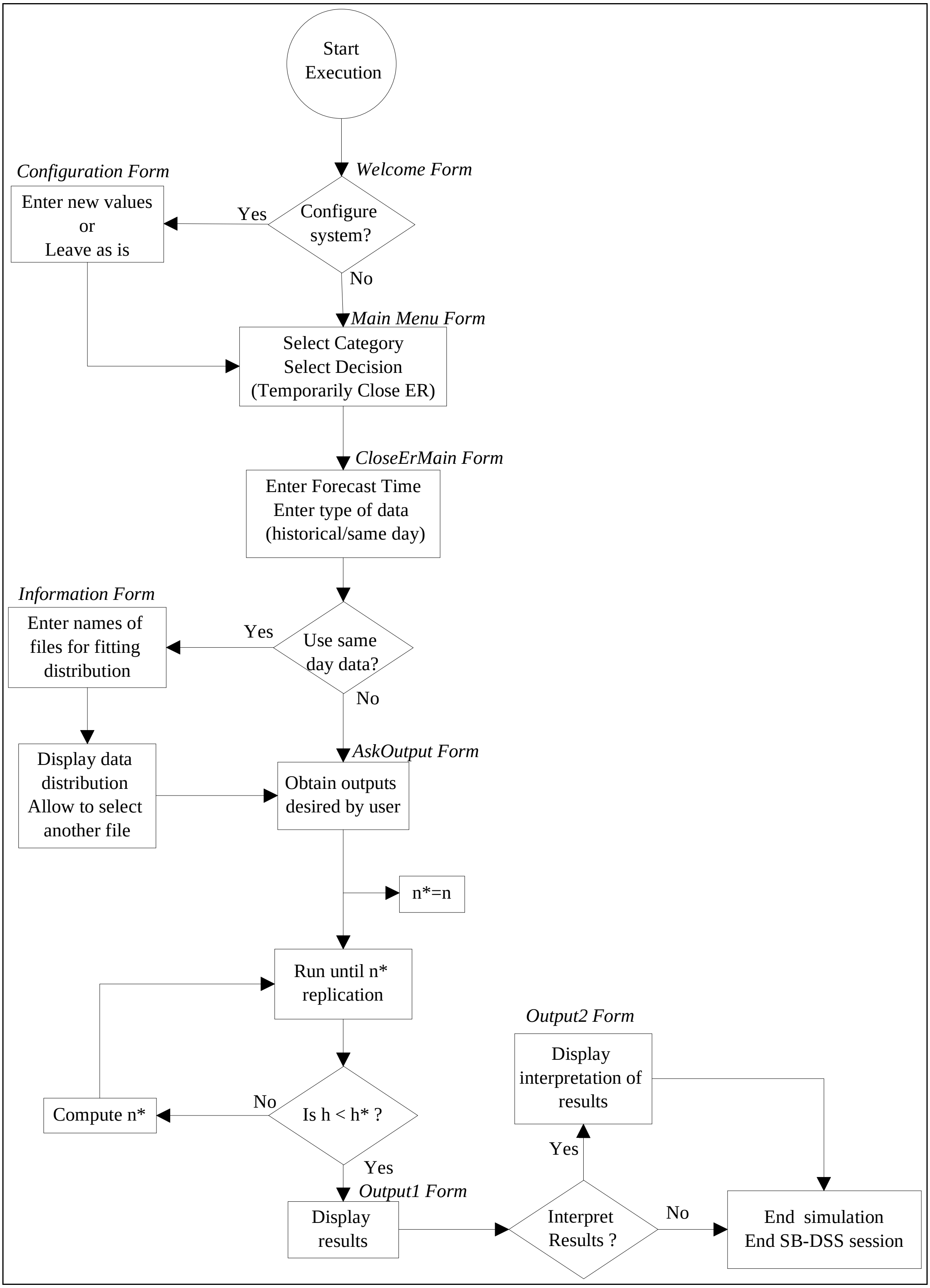


Figure 5: User Interaction

```
Sub Abrir()
    '----------------------------
    'Welcome user to DSS
    '----------------------------
    Dim flag1, mychoice, mychoice2, dummy As Integer
    Dim interdist As String
    Set modelo = ThisDocument.Model
    Set out = modelo.SIMAN
    modelo.QuietMode = True
    Load Welcomeform
    Welcomeform.Show
    Unload Welcomeform
    '----------------------------
    'Display DSS Main Menu
    '----------------------------
    Load MainMenu
    flag = 1
Startmain:
    Do While flag = 1
     MainMenu.CommandButton1.Cancel = False
     MainMenu.Show
     If MainMenu.CommandButton1.Cancel = True Then
            modelo.Pause
            End
        End If
    Loop
    If MainMenu.OptionButton1.Value = True Then mychoice = 1
    If MainMenu.OptionButton2.Value = True Then mychoice = 2
    If MainMenu.OptionButton3.Value = True Then mychoice = 3
    Select Case mychoice
        Case 1
            StaffManagement.Show
        Case 2
            If MainMenu.ComboBox2.ListIndex = 0 Then
                Load CloseERmain
Startcloseer:
                flag1 = -1
                CloseERmain.PreviousBtn.Cancel = False
                CloseERmain.Show
                If CloseERmain.PreviousBtn.Cancel = True Then
                    Unload CloseERmain
                    flag = 1
                End If  'when user clicks on continue
                If flag <> 1 Then
                    Call CloseERlogica
If CloseERmain.OptionButton2.Value = True Then
                        Load AskOutput
                        AskOutput.Show
                        If AskOutput.PreviousBtn.Cancel = True Then
                            Unload AskOutput
                            flag1 = 2
                        Else
                            flag1 = -1
                            flag = -1
                        End If
                    End If
If CloseERmain.OptionButton1.Value = True Then
                         Load information
Startinfo:
                         information.PreviousBtn.Cancel = False
                         information.PreviousBtn.Enabled = True
                         information.Button1.Enabled = True
                         information.Button2.Enabled = False
                         information.Show
                         If information.PreviousBtn.Cancel = True Then
                             flag1 = 2
```

```
Else
                            Call OpenInput
                            information.Show
                            interdist = information.distribution.Text
                            createmodule.data("interval") = interdist
                            createmodule.UpdateShapes
information.Hide
                            Load AskOutput
                            AskOutput.PreviousBtn.Cancel = False
                            AskOutput.Show
                            If AskOutput.PreviousBtn.Cancel = True Then
                                Unload AskOutput
                                flag1 = 3
                            Else
                                flag1 = -1
                                flag = -1
                            End If
                        End If
                        If flag1 = 3 Then GoTo Startinfo
                    End If
                End If
If flag1 = 2 Then GoTo Startcloseer
            Else
                modification.Show
            End If
        Case 3
            inventory.Show
    End Select

    If flag = 1 Then GoTo Startmain
    Unload AskOutput
    modelo.Application.refresh
End Sub
```

Figure 6: VBA Routine to Handle the User Interface

```
Sub ModelLogic_RunEndReplication()
    '----------------------------
    'myinit1=value of var adriana
    '-----------------------------
    Dim AdrianaVar, RepElement As Integer
    AdrianaVar = modelo.Modules.Find(smFindTag, "variabletag")
    If AdrianaVar > 0 Then
        Set variablemodule = modelo.Modules.Item(AdrianaVar)
    Else
        MsgBox "Did not find the variable module"
    End If
    MyInit1 = Val(variablemodule.data("value(1,6)"))
    RepElement = modelo.Modules.Find(smFindTag, "replicatetag")
    If RepElement > 0 Then
        Set replicatemodule = modelo.Modules.Item(RepElement)
    Else
        MsgBox "Did not find the replicate module"
    End If
    myinit2 = Val(replicatemodule.data("NumReps"))
    repli1 = repli1 + 1
    If repli1 = MyInit1 And MyInit1 = myinit2 Then
        Call Replications1
    ElseIf repli1 = myinit2 Then
        MsgBox "Second Time around"
        Call Replications2
    Else
    End If
End Sub
```

Figure 7: VBA Routine to Modify VARIABLES Element

7 OUTPUTS AND RESULTS

After the simulation run is completed, the output data needs to be analyzed and presented to the user in a helpful manner. The measures of performance of interest for the selected decision are tracked down the simulation run by sending their values to an output file.

By doing this, the average value of each replication is recorded. These averages can then be used to perform analysis on the values across replications based on the central limit theorem. For the prototype, these analyses are based on the average across replications on measures of performance such as the number of beds not occupied by a patient, the number of patients in the waiting room and time in the system for all patients. Confidence intervals are built on these averages. These intervals are built through the execution of a routine in VBA, which calculates them for the specified outputs.

The analysis of outputs can also be performed based on percentiles. Using percentiles is important because ER systems do not behave linearly. When their capacity reaches between 80 and 90 percent, they reach saturation. Therefore, many times the user might be interested in finding out for example how many patients wait more than 80% of the ER standard time.

These results are presented in the *Output* form. The numbers displayed in this form are the average numbers across replications for each specific measure of performance. In addition, a message is given to the user on whether the facility should or should not be closed. The system can draw this conclusion based on rules specified in routines inside VBA. For some institutions, this decision is made when the number of patients in the facility exceeds a fixed number. For others, this decision happens when there are no resources to safely care for additional patients, or patients waiting times exceeds a fixed amount of hours.

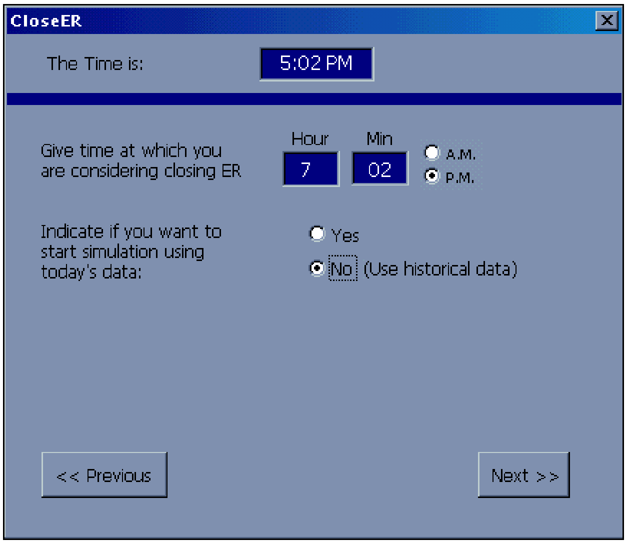
Figure 8 to Figure 11 present a series of screens that serve as example of a given scenario. The parameters of the scenario are self-explanatory.

8 CONCLUSIONS AND FURTHER WORK

One of the characteristics of the model built for the prototype is that it is a flexible. The model is easily customized by allowing the user to reconfigure the ER system via the user interface without the user having to access the simulation software or language at any time.

This effort resulted in the classification of decisions made by ER managers into three major categories. Under each category, decisions that can be evaluated with simulation were identified. This was accomplished by defining the inputs that a simulation model would required to run and evaluate that particular decision together with the simulation outputs that would aid the managers in making the decision.

The use of on-line data to drive the operations of the SB-DSS was another accomplishment of this effort. This allows the system to access data on-line, from the ER information systems, and feed it to the system. This is possible due to the development of a series of data transformation routines that handled the data and communicated it from one tool to the other when necessary. Furthermore, a user-friendly interface enables the user to provide different configurations of the ER facility; hence, the simulation model itself is flexible because the user can actually change it at runtime. This has resulted in an automatic system that does not require the user to be knowledgeable of simulation modeling techniques. Hence, it is expected that users of a full scale DSS will be less hesitant or afraid to use it.

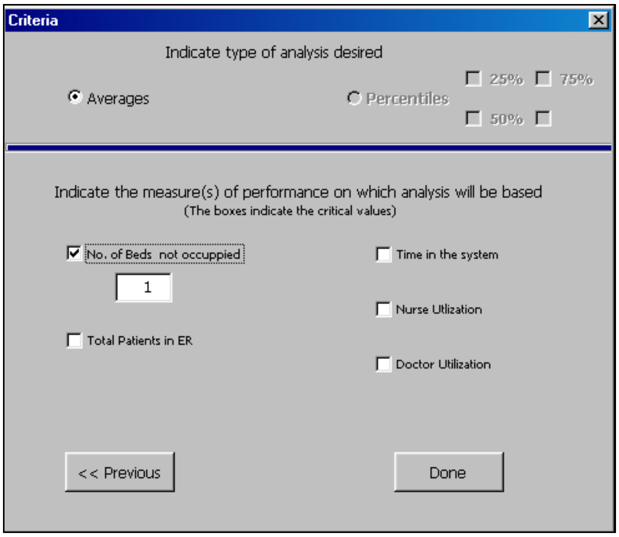

CloseER

The Time is: 5:02 PM

Give time at which you are considering closing ER — Hour: 7 Min: 02 ○ A.M. ● P.M.

Indicate if you want to start simulation using today's data: ○ Yes ● No (Use historical data)

Figure 8: Close ER Form-Scenario1


Criteria

Indicate type of analysis desired

● Averages ○ Percentiles □ 25% □ 75% □ 50% □

Indicate the measure(s) of performance on which analysis will be based
(The boxes indicate the critical values)

☑ No. of Beds not occuppied: 1

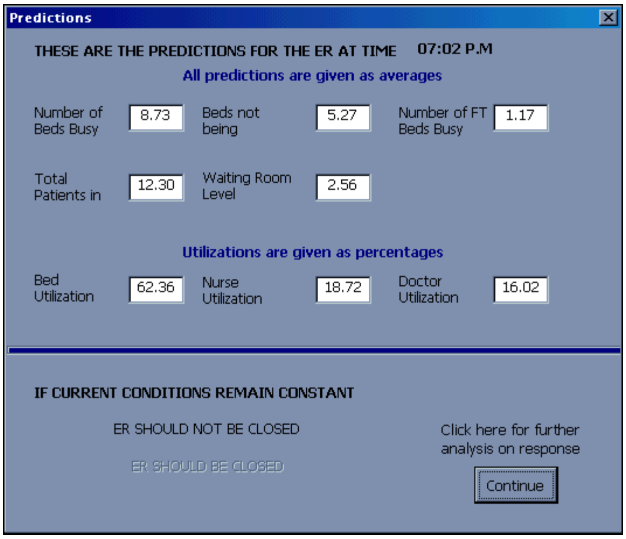
□ Total Patients in ER

□ Time in the system

□ Nurse Utlization

□ Doctor Utilization

Figure 9: AskOutput Form-Scenario1



Predictions

THESE ARE THE PREDICTIONS FOR THE ER AT TIME 07:02 P.M

All predictions are given as averages

Number of Beds Busy: 8.73

Beds not being: 5.27

Number of FT Beds Busy: 1.17

Total Patients in: 12.30

Waiting Room Level: 2.56

Utilizations are given as percentages

Bed Utilization: 62.36

Nurse Utilization: 18.72

Doctor Utilization: 16.02

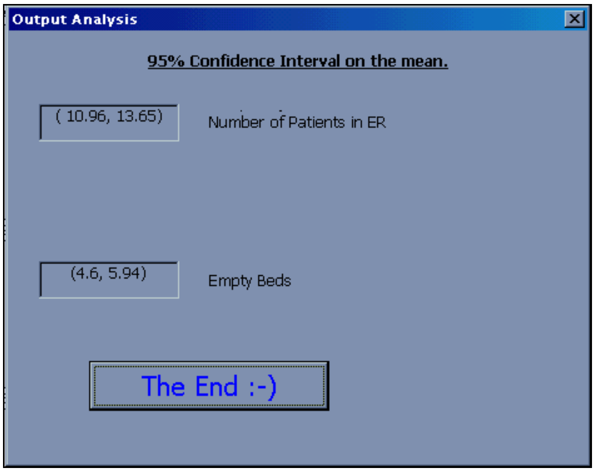
IF CURRENT CONDITIONS REMAIN CONSTANT

ER SHOULD NOT BE CLOSED

ER SHOULD BE CLOSED

Click here for further analysis on response

Figure 10: Output1 form-Scenario



Output Analysis

95% Confidence Interval on the mean.

(10.96, 13.65) Number of Patients in ER

(4.6, 5.94) Empty Beds

Figure 11: Ouput2 form

REFERENCES

Centeno, M.A., M. L Garcia, N. DeCario, and C. Rivera. 1995. Reducing Time in an Emergency Room via a Fast-Track. *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, D. Goldsman, 1048-1053.

Forgionne, G. and R. Kohli. 1996. HMSS: A Management Support System for Concurrent Hospital Decision Making. *Decision Support Systems.* 16. 209-229.

Freedman, R. 1994. Reduction of Average Length of Stay in the Emergency Room Using Discrete Simulation. *Proceedings of the Simulation In Health Sciences Conference*, ed. Anderson and Katzper, 6-8.

Kirtland, A., J. Lockwood, K. Poisker, L. Stamp, and P. Wolfe. 1995. Simulating an Emergency Department is as Much Fun as.. *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, D. Goldsman, 1039-1042.

Kittell, R.P, and A. Pallin. 1992. Mercy Hospital: Simulation Techniques for ER Processes. *Industrial Engineering.* 24(2). 35-37.

Lange, V. 1997. Blueprints Designed to Improve Quality of Care Using simulation Tools. Success Stories. http://www.medmodel.com/stories.html. (09 Sep. 1998).

Lowder, B. Durham Regional Saves $150,000 Annually Using Simulation Tools. Success Stories.1997. http://www.medmodel.com/stories.html. (09 Sep. 1998).

McGuire, F. 1994. Using Simulation to Reduce Length of Stay in Emergency Departments. *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, A. Sadowski, A. F. Seila, 861-867.

AUTHOR BIOGRAPHIES

ADRIANA M. ALVAREZ has a B.S. in Industrial and Systems Engineering and a M.S. in Industrial Engineering from Florida International University. She has been a Student Chapter president for IIE and INFORMS at F.I.U. Her research interests are in the area of simulation of health care systems. She is a member of IIE, INFORMS, and AΠM.

MARTHA A. CENTENO is an associate professor in the Department of Industrial and Systems Engineering at Florida International University. She has a B.S. in Chemical Engineering from ITESO (Guadalajara, Mexico), a M.S. in Industrial Engineering from LSU (Baton Rouge, LA), and a Ph.D. in Industrial Engineering from Texas A&M University (College Station, TX). Her current research interests are in the design and development of integrated simulation systems, on-line goal driven simulation, data mining, and engineering education. She is a member of TBΠ, AΠM, IIE, SCS, ASA, and ASEE.