

3-11-1999

# A simulation-based decision support system for evaluating operating policies in an emergency room facility

Adriana M. Alvarez

*Florida International University*

**DOI:** 10.25148/etd.FI13101613

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>



Part of the [Industrial Engineering Commons](#)

---

## Recommended Citation

Alvarez, Adriana M., "A simulation-based decision support system for evaluating operating policies in an emergency room facility" (1999). *FIU Electronic Theses and Dissertations*. 1228.

<https://digitalcommons.fiu.edu/etd/1228>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

A SIMULATION-BASED DECISION SUPPORT SYSTEM FOR EVALUATING  
OPERATING POLICIES IN AN EMERGENCY ROOM FACILITY

A thesis submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

by

Adriana M. Alvarez

1999


To: Dean Gordon Hopkins  
College of Engineering

This thesis, written by Adriana M. Alvarez, and entitled A Simulation-Based Decision Support System for Evaluating Operating Policies in an Emergency Room Facility, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

  
Dr. Joe Chow

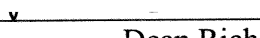
  
Dr. Khokiat Kengskool

  
Dr. Martha A. Centeno, Major Professor

Date of Defense: March 11, 1999

The thesis of Adriana M. Alvarez is approved.

  
Dean Gordon Hopkins  
College of Engineering

  
Dean Richard L. Campbell  
Division of Graduate Studies

Florida International University, 1999

©COPYRIGHT 1999 by Adriana M. Alvarez

All rights reserved.

## DEDICATION

To *my mother*, for being my angel. Without her love, patience, support, and encouragement I wouldn't have been able to finish this work; and to my brother for being there for me whenever I needed help. I love you both.

## ACKNOWLEDGMENTS

I would like to thank the members of my committee for their help and contributions. I would also like to thank the ARISE Center for allowing me to use all of its resources to successfully complete this work.

To my major professor, Dr. Martha Centeno, thank you, thank you, and thank you. With all of her patience, support, encouragement, and above all her confidence in me, I was able to see the light. Now I really believe I am here *Doctora*.

I would also like to thank all of my friends for being with me throughout the good and the bad times. Their constant words of encouragement and caring attitudes helped me throughout this whole process. Even the cruelest jokes about my thesis helped. Special thanks go to M. Florencia Reyes, Daisy Correa, Marelys Garcia, Gabriela Peñaloza, Ma. Ruth Jimenez, Libia Ortiz, Roberto Perez, Emil Mejia, Ignacio Goris, Cristina Albacete, Evelyn Ortiz, Lylliam Robelo. You are the best friends anyone can have. I love you all.

Very special thanks must also go to the ISE staff: Nancy Urbina and Madlyn Downs. I really appreciate their willingness to help every time I needed, especially in the most stressful moments.

ABSTRACT OF THE THESIS

A SIMULATION-BASED DECISION SUPPORT SYSTEM FOR EVALUATING  
OPERATING POLICIES IN AN EMERGENCY ROOM FACILITY

by

Adriana M. Alvarez

Florida International University, 1999

Miami, Florida

Professor Martha A. Centeno, Major Professor

Increased pressure to control costs and increased competition has prompted health care managers to look for tools to effectively operate their institutions. This research sought a framework for the development of a Simulation-Based Decision Support System (SB-DSS) to evaluate operating policies. A prototype of this SB-DSS was developed. It incorporates a simulation model that uses real or simulated data. ER decisions have been categorized and, for each one, an implementation plan has been devised. Several issues of integrating heterogeneous tools have been addressed. The prototype revealed that simulation can truly be used in this environment in a timely fashion because 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. Hence, the SB-DSS tailors its responses to each node in the hierarchy.

## TABLE OF CONTENTS

CHAPTER	PAGE
<b>INTRODUCTION.....</b>	<b>3</b>
1.1 Problem Statement.....	3
1.2 Goal and Specific Objectives.....	3
<b>LITERATURE REVIEW .....</b>	<b>3</b>
2.2 Simulation Modeling Overview.....	3
2.3 Simulation in the Healthcare Industry .....	3
2.4 Decision Support Systems Overview.....	3
2.5 Decision Support Systems in the Healthcare Industry.....	3
2.6 DSS and Simulation.....	3
2.7 Summary.....	3
<b>DESIGNING A SB-DSS .....</b>	<b>3</b>
3.1 A SB-DSS Framework.....	3
3.2 Data Acquisition .....	3
3.3 Processing and Modeling Engines .....	3
3.4 Outputs and Results .....	3
3.5 Communication Protocols.....	3
<b>A PROTOTYPE SB-DSS.....</b>	<b>3</b>
4.1 General Specifications .....	3
4.2 Inputs Section.....	3
4.3 Processing and Modeling Engines .....	3
4.4 Outputs and Results .....	3



4.5 Testing and Validation.....	3
4.6 Sample Scenarios .....	3
<b>CONCLUSIONS AND FURTHER WORK.....</b>	<b>3</b>
<b>REFERENCES.....</b>	<b>3</b>
<b>APPENDICES .....</b>	<b>3</b>

## LIST OF TABLES

<b>TABLE.....</b>	<b>PAGE</b>
Table 1: Interviews Summary of ER Decisions.....	3
Table 2: E-mail Survey Summary of ER Decisions .....	3
Table 3: Human Resources: Inputs and Outputs.....	3
Table 4: Inventory Control: Inputs and Outputs .....	3
Table 5: Facility Operation Policies: Inputs and Outputs.....	3
Table 6: Patient Categories .....	3
Table 7: Simulation Results .....	3

## LIST OF FIGURES

<b>FIGURE .....</b>	<b>PAGE</b>
Figure 1: Patient Flow in an Emergency Room.....	3
Figure 2 : Proposed SB-DSS Framework .....	3
Figure 3: Classification of ER Decisions.....	3
Figure 4: Process of adding a new category/decision to SB-DSS .....	3
Figure 5: SB-DSS User Interaction.....	3
Figure 6: Patient Flow through ER .....	3
Figure 7: Welcome form-Scenario1 .....	3
Figure 8: MainMenu form-Scenario 1 .....	3
Figure 9: Configuration form-Scenario1 .....	3
Figure 10: Close ER form-Scenario1 .....	3
Figure 11: AskOutput form-Scenario1 .....	3
Figure 12: Output1 form-Scenario .....	3
Figure 13: Ouput2 form-Scenario 1 .....	3
Figure 14: Welcome form-Scenario2.....	3
Figure 15: Main menu form-Scenario2.....	3
Figure 16: Close ER form-Scenario2.....	3
Figure 17: Information form-Scenario2.....	3
Figure 18: Information form(b)-Scenario2 .....	3
Figure 19: AskOutput form-Scenario2 .....	3
Figure 20: Output1 form-Scenario2.....	3

Figure 21: Output 2 form-Scenario2.....	3
---	---

## **CHAPTER 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 point 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 in an Emergency Room. These decisions require the use of tools that will allow quantifying the impact of changes to the current system on operating costs and quality of service.

Managers and administrators in a variety of industries have turned to computer simulation to aid them in their decision-making process. The health care industry is not the exemption because they do not necessarily possess the required expertise. Consequently, they need flexible tools to further aid them in making the right decisions. A simulation model is able to mimic the behavior of a real world system, so that actual or proposed changes to the operation of a system can be modeled to determine the effect of these changes on system performance. A well-designed Decision Support Systems (DSS) is capable of retrieving, handling and analyzing relevant data and information

about the system being studied (Mitra, 1986). Therefore, simulation and DSS complement each other to support their users and allow them to make more rational and appropriate decisions.

A DSS, centered on a simulation model, has been designed for ER systems. A prototype of it was developed to understand the various issues relating the integration of simulation models and traditional information systems. The prototype demonstrated that SB-DSS is feasible, and it seems to be a promising tool for the management of ER systems.

The first section of this chapter describes in details the problem this effort seeks to address, followed by a description of the goal and specific objectives of this effort. Chapter two presents a review of pertinent literature on simulation and Decision Support Systems and their applications in the health care industry. Chapter three presents the methodology used in this research and the design of a SB-DSS. Chapter four describes the components of the prototype developed in this research. Finally, chapter five discusses the results derived from this effort and further research path.

## **1.1 Problem Statement**

The problem being addressed in this work is the lack of flexible tools for stochastic decision making in the management and operation of Emergency Rooms (ER). Nowadays, hospitals find it necessary to improve and optimize the operations of their Emergency Department to provide good quality services to its customers at the least cost. In the 1990's, increased competition, reimbursement problems, and healthcare reform have forced managers to search for diverse methodologies and

techniques to reduce operating cost and increase customer satisfaction (McGuire, 1994). Reduction in the number of patient visits, increasing patient dissatisfaction with extended stays in ER, and limited number of resources are some additional issues that have prompted health care administrators and hospitals managers to look for systematic ways to obtain feasible and better solutions to these problems.

A significant number of hospital admissions come through the ER. Hence, the need to pay close attention to the operations of this facility in trying to provide good customer service to all types of patients. Emergency Rooms are visited by patients in need of true emergency medical care and, in some instances, by patients with relatively minor medical problems. Figure 1 reflects the typical processes followed by patients arriving to an Emergency Room (Freedman (1994), McGuire (1994), and Centeno et. al, (1995). Upon arrival to the ER, patients are usually classified into two broad classes: critical and non-critical. Non-critical patients enter a reception area where a registration clerk takes his or her information, and writes down their arrival time. These patients then proceed to the triage nurse where the severity of their illness is determined. Whenever a bed becomes available in the ER, patients are called in for treatment. Critical patients, those having life-threatening injuries, do not go through the reception area. They proceed directly to the ER to receive treatment. For non-critical patients, this process sometimes takes an excessive amount of time that results in patient dissatisfaction and willingness to go somewhere else for treatment.

In the past 30 years, simulation modeling has been progressively used to support operational decision-making in the health care industry. Among the reasons for this usage are the stochastic nature of ER systems and the complexity of the interactions

between the different components of it. Existing analytical tools do not lend themselves for modeling scenarios where all these characteristics come together simultaneously. Additionally, through the use of simulation in health care, managers have been able to evaluate “what-if” scenarios without actually having to interrupt the daily operations of the facility.

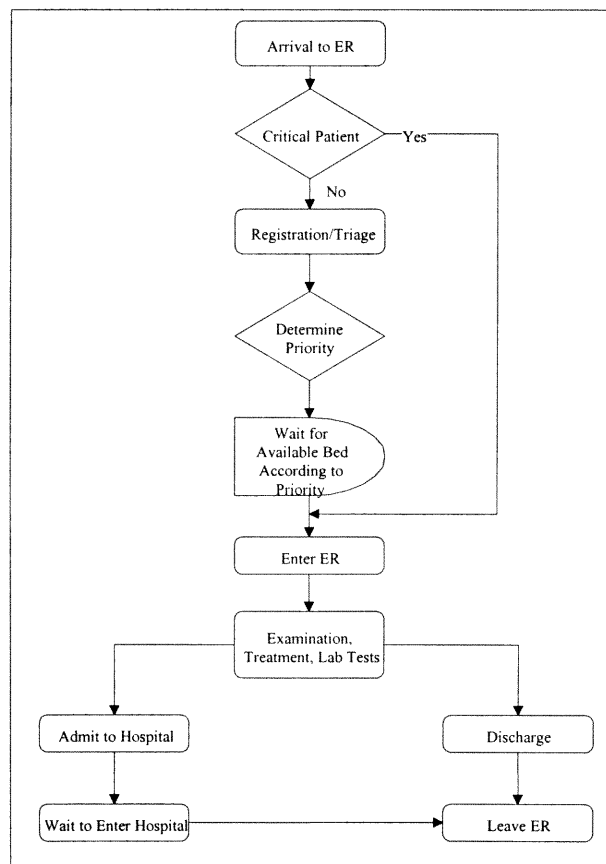


Figure 1: Patient Flow in an Emergency Room

Kirtland et al. (1995) conducted a study of the Peninsula Regional Medical



Center's (PRMC) Emergency Room in Salisbury, MD, where patients' visits decreased by 16% over a two-year period. During this time, the PRMC Emergency Room lost 8000 visits, denoting an opportunity loss of approximately 2 million dollars in revenue for the hospital.

McGuire (1994) describes how simulation was used to test alternatives to reduce patients' length of stay at the ER of a SunHealth Alliance Hospital. Similarly, Freedman (1994) used simulation to study how policy changes affected patients' length of stay in the ER at two different hospitals in the northeastern United States. Kittell and Pallin (1992) and Centeno et al. (1995) describe two different simulation projects of the operations at the ER of Mercy Hospital in Miami, FL. They identified the impact on patients' length of stay and savings in staff and resource allocation respectively if the ER were to implement a fast-track system.

From these efforts, one can conclude that there is a need to better manage the operations of an ER. The majority of these studies have focused only on evaluating changes to the number of available beds for patients and the addition or reduction of the number of resources. Another crucial characteristic present in these studies is the use of simulation as a **one-time project**. Whenever simulation has been employed by a hospital as a decision-support tool, it has been used only once to address a particular problem. Once the project concludes, the proposed changes are either implemented or disregarded, and the simulation model thrown away. Thus, these studies have failed to implement a tool that can be used on a continuous basis to evaluate a larger number of alternatives with multiple objectives. There still is a need for a decision support tool that can be used to explore many scenarios as time passes.

An important observation to make is that, despite the success of these projects in some hospitals and clinics, the use of simulation for decision making in health care systems has been the exemption rather than the rule. Hakes et al. (1994) expose several barriers present when implementing simulation in the health care industry. These barriers need to be taken into consideration when developing a Simulation-Based DSS (SB-DSS). Many managers in charge of ER at different hospitals do not have an engineering or technical background. Many are administrative or clinical type of employees who have been promoted to a management position. Therefore, they tend to evaluate policies and make decisions based on how things have always been made. They rely also on their intuition for their everyday decision making activities. This has been one of the main reasons, which account for the lack of use of simulation by ER managers.

Nonetheless, ER managers are faced with having to make decisions on different aspects of the operation of the facility. They are responsible for scheduling the necessary staff to account for the demand for services at different times of the day. They need to determine at what point to bypass the facility; i.e. when to advise local authorities to stop driving ambulances with life-threatening patients to the ER because the facility is full. Other types of decisions involve budgeting and cost control activities such as the forecasting and inventory control of medical supplies. Improper usage of supply ordering and distribution control strategies could result in high monetary losses to the facility because supplies may not be charged to patients.

Other types of decisions are those pertaining to changes in the operations of a specific area within the ER. Examples of these types of decisions are modifications to

the facility layout such as the addition of waiting rooms, augmenting the number of beds, and so forth.

## **1.2 Goal and Specific Objectives**

The main goal of this work was to design and prototype a Simulation-Based Decision Support System (SB-DSS) to be utilized by decision-makers for evaluating different operating policies of an Emergency Room. This work was developed based on a case study conducted at Mercy Hospital.

In order to achieve this goal, the most important types of decisions that ER managers need to make were identified. The different factors that need to be considered when evaluating these decision, as well as their expected outcomes, were also defined. One of these decisions was then selected for the design and prototype implementation of the DSS. Chapter three gives a thorough description of the methodology used in this study.

## CHAPTER 2:

### LITERATURE REVIEW

Computer simulation has been used in many ways within the health care industry. Most of the work reviewed in this study has concentrated on the use of simulation as a decision-making tool for health care managers and administrators. This chapter includes a brief overview of simulation modeling to familiarize the reader with some of the vocabulary used by simulation community. It then presents some of the applications of simulation as a decision-making tool in healthcare. The second part of this chapter is devoted to Decision Support Systems (DSS). We present an introduction to the basic concepts and terminology used in the DSS arena, and then proceed to review how DSS have been used in combination with simulation to support decision-making in health care.

For a detailed treatment of simulation modeling in general, see the works by Kelton and Law (1991), Shannon (1975), Banks, et al. (1996), and Pegden, et al. (1995).

#### **2.2 Simulation Modeling Overview**

Simulation modeling is a methodology that enables the building of a model to study the temporal behavior of a system and to evaluate different alternatives for its operation (Pegden et al. 1995). The Simulation Modeling Process (SMP) is divided in several stages, beginning with problem definition and ending with documentation and implementation. In *problem definition*, the problem that needs to be addressed by the

project is identified. Once that is achieved, a clear set of *goals and objectives* are identified. Having established the goal, a *model* is *conceptualized* and *formulated*. Simultaneously, the *data collection* phase is initiated to gather the data that will actually serve as inputs to the simulation model. When such data is ready, the conceptual model is built and transformed into a specific simulation package in the *model translation* stage. Once the model has been built, we need to ascertain that in fact it is, as we wanted it to be, and ascertain that the model is an accurate representation of the system being studied. This is called the *model verification and validation* stage. With a validated model, *experimentation* may begin to collect the desired information. After the model has been run, the outputs are *analyzed and interpreted* so as to make statistical inferences, and determine courses of action. The last stage in the simulation modeling methodology is *documentation and implementation* where a detailed report is written.

In simulation modeling, there is a clear distinction between a model and a system. Banks et al. define a model as a representation of a system which is created for the purpose of studying that system. A model is intended to be similar but not as complex as the real system. A system, on the other hand, is as a collection of objects or entities that interact towards the accomplishment of some goal Banks et al. (1996), and Pegden et al. (1995).

Banks et al. (1996) and Pegden et al. (1995) described some of the advantages of using simulation. With simulation, new policies can be investigated without interrupting the systems current operations, new hardware designs and physical layouts can be tested before they are bought or implemented, and “what if” scenarios can be

explored. This allows for the discovery of sensitive variables, those that influence the measures of performance the most. These advantages have persuaded managers in a wide variety of industries to use simulation as a decision-making and analytical tool.

As computer technologies evolve, researchers have continuously sought ways of overcoming some of the shortcomings of the simulation modeling methodology. Nowadays, the software is much more user friendly than in the past. Commercial packages include support for activities such as input and output analysis. Various prototypes addressing simulation optimization have also been developed. Nonetheless, simulation modeling is a powerful tool that has not yet reached all those communities who could benefit from its use because of the complexities of it. However, many researchers are working towards making it more accessible. This work seeks to make its contribution in reaching the health care community.

### **2.3 Simulation in the Healthcare Industry**

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, and their tendency to rely on traditional management tools. Analytical decision making tools have been preferred because they provide an "optimal" solution; however, these tools impose a series of constraints and assumptions that need to be made about the system. Simulation does not impose these constraints

(Hakes et al. 1994).

There has been a change in the past few years. Several simulation projects have taken way in hospitals around the world. Due to the success of these projects in terms of process improvement, costs reduction, better resource allocation and utilization, and improved customer satisfaction, many health care institutions have awakened and discovered the advantages of using simulation as their primary decision support tool.

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. Great attention has also been paid to the simulation of subsystems within a hospital such as the Emergency Room, Operating Room, ancillary department, and ambulance dispatching centers. Shuman et al. (1975) categorized simulation models of healthcare systems into two broad categories: *demand-oriented models* (admissions systems) and *resource-allocation systems*. The former focuses on the admission of patients to the various services within the hospital, whereas the latter deals mostly with resource, facility and equipment allocation. However, some simulation studies cannot be classified into a single category because they contain characteristics belonging to both domains.

The admissions system is the first point of contact between patients and the health care facility. Therefore, it is important for hospitals and clinics to have an efficient admissions system. In the case of outpatient clinics, patient admissions can be

controlled more effectively through the optimal scheduling of patients' appointments. The objective of the scheduling system then becomes the better utilization of human resources and equipment in the facility (physicians, nurses, beds, examination rooms, etc). As an example, Lowery (1996) developed a simulation model of patients' arrivals and discharges from a hospital. Kuljis and Paul (1995) have developed a visual simulation model of a generic outpatient clinic in the United Kingdom. The purpose of their work is to allow administrators to see how clinic appointments as well as operating policies affect patients' waiting times. Their work has led to the development of a visual simulation package called CLINSIM.

Resource-oriented models concentrate on the allocation of human resources, facilities and equipment. This area also includes studies on the evaluation of hospitals operating policies within hospital's subsystems such as operating rooms, ancillary tests laboratories, and emergency rooms. Pitt (1997) reports on the early stages of developing a generalized simulation system to support resource planning and allocation. This system called PRISM (Planning Resources using Interactive Simulation Modeling), incorporates a Visual Basic graphical user interface linked to WITNESS simulation software and serves as a tool for studying what if scenarios. Case studies have been developed using PRISM to test the simulated system's validity. In the case of allocation of non human resources such as hospitals beds, Dumas (1985) reports on the development of a simulation model to investigate bed utilization at the Mount Sinai Hospital in New York City. This simulation model, which was built using SimScript II.5, mirrors the actual bed configuration and it is used to evaluate the effect of proposed modification on bed allocation, and usage.



Recently, most simulation studies of ERs have focused on the evaluation or analysis of new policies with the objective of reducing the patient's average length of stay in the facility. 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.

There are other simulation studies related to the health care industry that do not necessarily fall into one of the categories mentioned above, rather they contain elements from both domains. For example, Lee and Liu (1988) simulated the emergency call system in Taipei to determine the number of ambulances needed. They also performed sensitivity analysis to determine which of the input factors (speed of ambulance and frequency of casualties) had the greatest influence on the system's performance (ambulance delay time). Management engineers at St John Hospital and Medical Center (Miller, 1997) used simulation to evaluate changes to its OR surgical environment. Most of these changes included expansions, renovations, and reduction in case schedule hours. Their purpose was to determine if the bed capacity suggested for the renovation would be sufficient to meet future demands. Using simulation, management was able to determine that the proposed renovation plans would not satisfy the demand; therefore, other recommendations were made and later implemented. 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).

## **2.4 Decision Support Systems Overview**

The literature gives several definitions and interpretations of a DSS and what its capabilities are. A general definition is the one provided by Mittra (1986), who defines a DSS as a computer-based information system, that aids managers in making efficient decisions. From a technical perspective, the main components of a DSS are a database, a DBMS, a set of mathematical models that contain procedures for analyzing data, and a user interface (Hammer et al. 1997). According to Silver (1991), DSS are systems that aid in the decision-making process in those situations where human knowledge is required but there is lots of information impossible for the human mind to process. These systems are intended to enhance human abilities but not to replace them. The main objective of a DSS is to help its users come to a decision.

DSS began to evolve in the early 1970's. They were developed in an effort to correct the failures of Management Information Systems (MIS) present at that time. Many a times, the information that management needed was not in the reports provided by traditional MIS. Moreover, these reports were so big that it was difficult for managers to locate the data they needed. In addition, managers also felt that the available systems were oriented towards managing large quantities of information, but not towards providing useful information (Silver, 1991). In other words, these systems

failed to provide information for supporting managers' decisions. DSS then came along to try to solve this problem through their capability of providing timely access to relevant data for specific managerial tasks.

Mittra (1986) suggests five phases in the development of a DSS, which are similar to the phases found in the simulation modeling process. These phases are problem definition, system analysis and model conceptualization, preliminary system design, detail system design, and implementation, maintenance and evaluation.

Silver (1991) describes the main capabilities of a DSS, which include selective data retrieval, data aggregation, and statistical data analysis. The system must be capable of presenting relevant information in some form of computerized graph or report. Another capability of a DSS is the use of simulation to predict consequences of the actions taken by decision-makers and evaluate different "what-if" scenarios. Lastly, the system needs the capability to allow its users to perform other types of statistical analysis such as regression analysis and optimization techniques such as linear programming.

During the past years, DSS have been accepted in a wide variety of industries. According to Sauter (1997), there are four main factors which have contributed to the acceptance and greater use of DSS. The first factor is the movement from mainframes to desktop computing which has made technology easier to use. The second factor is the development of user friendly packages, which do not require its users to be expert in computer languages to access the data they need. Both of these trends in the computer industry have led to the third factor: less fear from management to use computers. The last factor contributing to the growth in the usage of DSS is that managers have realized

the high cost of not using the available technology.

DSS have been developed for a different number of applications. They have been used in diverse fields such as marketing, medical, financial, retail, and manufacturing. The applications in these fields include strategic planning, production planning, inventory control, forecasting, scheduling, etc. Applegate and Pearlson (1995) describe how Mrs. Fields Inc. developed a sophisticated DSS that aids managers in several scheduling activities, marketing support, and hourly sale projections. The system works mainly by taking as input characteristics of the **workday**, which are gathered through a series of questions presented to the user, and then retrieving a specific mathematical model for computing the day's schedule. (Moynihan et al. 1995) developed a DSS, called SM<sup>2</sup>ILE (Simulation Model for Marketing and Integrated Logistics Environments), for logistic planning which aids in the design of optimal distribution and logistic networks. The system also allows financial assessment of changes in distribution policies and programs. Lursen and Yang (1996) developed a DSS for the truck dispatching industry, which aids dispatchers in making "optimal" driver to load assignments with the objective of minimizing total non-revenue miles. This system is composed of a database model made up of a geographical, a customer, and a driver database. In addition, there is a Decision Support unit, which uses various heuristics to solve the driver to load assignments. This unit also makes suggestions taking into account the given constraints.

As it can be seen, the use of DSS has been spread through various industries, and each one with a different purpose. Sauter (1997) states that although DSS do not solve all the problem of an organization it does solve some problems well. This is true

specially in those situation where the goals include generating better alternatives, considering more options for solving a problem, using multiple analyses in solving a problem, using more appropriate data, better utilizing models, and considering what-if analyses. Although the problems addressed by organizations vary, there are some common results in those organizations, mainly cost saving decisions. Therefore, the successes of these examples as well as others have prompted more and more organizations to develop and implement DSS for their everyday operations.

## **2.5 Decision Support Systems in the Healthcare Industry**

In healthcare, DSS have been used to predict the spread of diseases, as a training tool for both practicing physicians and students, and as a diagnostic tool. They have also been used by managers to aid them in making decisions with respect to the allocation of resources, patient and resource scheduling, facility layout, to assess the impact of acquiring new machinery, more ambulances, and altering the patient mix. Hammer et al. (1997) point out that DSS are one of management's key tools to help them improve their system's operation. This allows their institutions to reduce costs while at the same time they maintain high quality standards.

One common application of DSS in healthcare, however, has been in physician's support for medical diagnosis. It is in this area where DSS together with expert systems began to evolve. The MYCIN system, developed at Stanford University in the early 1970's, is said to be the first true knowledge-based DSS. This system was developed to aid physicians in the diagnosis and treatments for blood disorders. The system works by

taking as input several symptoms present in a patient, and it then recommends performing certain laboratory tests on the patient. After the tests have been completed, the results are fed to the system, which would then suggest an appropriate drug treatment for alleviating the disease (Dankel and Gonzalez, 1993).

Similar systems have been developed to aid in the diagnosis of diseases. The Laboratory of Computer Science at the Massachusetts General Hospital developed DXplain during the 1980's. This DSS contains a large database with approximately 5000 clinical manifestations (signs, symptoms, and laboratory data) and over 2000 different diseases. DXplain collects information on the different symptoms present in the patient. The system also suggests what other type of information should be collected to arrive to a more appropriate decision about the type of disease the patient has. With this information, the system suggests which is the best treatment to follow. DXplain also serves as an electronic medical reference system that can be accessed at any time during the diagnostic process to further assist the physician. Similarly, Aase et al. (1993) developed a DSS to be used by physicians in Emergency Rooms which estimates the risk of Acute Coronary Heart Disease (ACHD). The aim of this system is to try to improve the distinction between patients with or without ACHD and thus with a need to be admitted to the Coronary Care Unit (CCU). The system was developed using data from 38 case histories and from clinical variables collected from 918 patients with acute chest pain. A statistical procedure based on Baye's formula is used to estimate disease probabilities. Preliminary use of this system indicates that reclassification of patients into two groups, patients with or without ACHD, gives a diagnostic accuracy of 89%.

Ferri et al. (1995) discussed the development of an object-oriented DSS for

planning the geographical allocation of resources with the objective of optimizing the cost-benefit ratio. The DSS contains the necessary algorithms and strategies to obtain the best solution. It receives as input the goal and constraints of the problem being addressed to then determine the best solution while it optimize the cost-benefit ratio.

Forgionne and Kohli (1996) proposed the use of a DSS called the Hospital Management Support System (HMSS) which enables the application of concurrent engineering in hospitals. The HMSS delivers the necessary information and knowledge to support concurrent engineering decision-making. This system integrates available stand-alone systems by complementing a DSS with an executive information system and an expert system.

O'Connell et al. (1996) have developed a computer based system to support all phases in the in-patient discharge process with the objective of eliminating redundant paperwork, transcription errors and reducing the time a patient needs to wait for a discharge order. The system works by filling much of the information needed in a discharge order beforehand with information obtained gathered from different databases, by guiding the physician in choosing discharge medications and preparing general discharge and medications instructions for the patient.

McClean and Millard (1995) report on the development of a bed-occupancy management and planning DSS in the United Kingdom. This system works by analyzing patient data through the use of a mathematical model. With this model, the average number of short, medium, and long-stay patients and their length of stay are estimated. In addition, the model is utilized to assess the effect of policy changes on these measures of performance. The DSS consists of a combination of database



functions for inputting, processing and reporting and a modeling capability to carry out “what if” type of analysis.

## **2.6 DSS and Simulation**

Decision-makers have used a variety of methods and techniques embedded in DSS to aid them in their decision making process. Many of these systems rely on databases and model bases to perform the different types of analysis that its users need to make. Aggarwal (1990) points out that most of the model-based systems are limited to using modeling techniques such as linear programming and regression. This presents a problem for those environments where the assumptions of these techniques do not hold. For these environments, simulation models may be a better and useful to be embedded in the decision support tool.

Beard (1998) discusses a research effort to develop a computer model to assist in the scheduling and evaluation of equipment used for the maintenance of golf course facilities. The system, named TEMSE (Turfgrass Equipment Model for Scheduling and Evaluation), is composed of five databases, which contain relevant information on the calendar of events, course attributes, equipment information, equipment operators’ information, and weather history. The system either uses the information contained in these databases or allows the user to modify it. It then runs a simulation model for the specified period of time and yields as a result the number of operations completed and incomplete, and operating times and costs for each equipment, hole, region and course. Bradi et al. (1990) describe how simulation was incorporated to the model base of a

DSS to aid decision-makers of a carpet manufacturing company. Simulation was used to help managers in selecting the most appropriate organizational strategies in the marketing, manufacturing and distribution systems of the firm with the purpose of enhancing the company's competitive advantage. The simulation model of the DSS mimics the firm's ordering process to evaluate various combinations of processing strategies. Derks et al. (1996) report on the development of a DSS for capacity planning in clinical laboratories in the Netherlands. The proposed DSS aids management with the problem of determining rules for scheduling samples and assigning staff to different departments within the laboratory. The main components of the DSS are a simulation and an optimization module which address departmental operation questions. These questions include the best planning rule for a department with a specific demand pattern, and department's performance under varying demand patterns.

In another attempt to marry simulation with DSS, the Commission of the European Communities funded the development of a micro-simulation model for decision support in attempts to control sexually transmitted diseases (STDs) in developing countries (De Vlas et al. 1998). The main purpose of their model, called STDSIM, is to aid in evaluating the effects of different STD control strategies and policies on the spread of these diseases. The model accomplishes this by describing the transmission, consequences, and intervention possibilities for five STDs including HIV. Future enhancements to the model will include economic aspects of STDs interventions and infection to allow cost-effectiveness evaluations.

## **2.7 Summary**

This incorporation of simulation models in DSS is a trend gaining popularity as reported in the literature (see Section 2.5). Yet, there are many technical and managerial issues that have not been yet understood to make simulation models a natural part of a DSS. However, research efforts continue and their results should make that happen.

## **CHAPTER 3:**

### **DESIGNING A SB-DSS**

This chapter presents the various steps to address design and implementation issues to develop a Simulation-Based DSS (SB-DSS). To fully realize a SB-DSS two major issues had to be understood and defined: a) types of decisions made in the management of ER, and b) communication protocols needed to provide information support to the SB-DSS. The discussion begins with an overview of the framework for the SB-DSS. Because the framework requires the interaction of several heterogeneous packages, the discussion is then broken down according to the requirements of each capability of the SB-DSS. These requirements pertain to 1) data acquisition, 2) data processing and modeling engines, 3) simulation output analysis, and 4) communication protocols.

#### **3.1 A SB-DSS Framework**

A framework of the proposed system together with its components was conceptualized as in Figure 2. The proposed 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. In addition to the functional sections, the system's framework is complemented with an external section, which serves as a feeder to the inputs section.

Each one of the elements in the framework is discussed in later sections. First, it is necessary to clarify the objective that such framework is supposed to support: help in *ER decision making*.

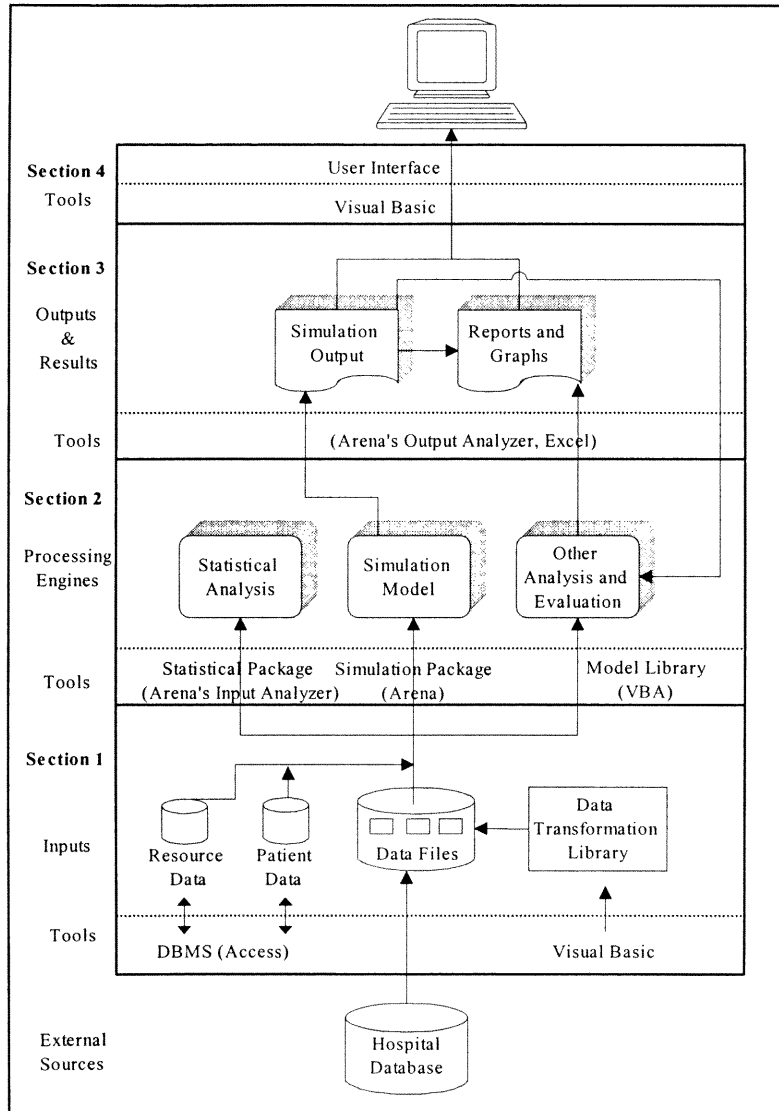


Figure 2 : Proposed SB-DSS Framework

To meet such an objective, it was necessary to become familiar with how

Emergency Rooms are operated, specially what are the different types of management decisions made in these systems. These decisions were identified in two phases. First, an in-depth review of the literature together with a review of health-care related web pages and interviews with ER managers at several hospitals in Miami-Dade County were made. Second, with the information gathered during the first phase, a survey was developed and distributed via e-mail among several ER managers/directors across hospitals in the United States.

From the initial review of the literature and health-care related web pages, it was possible to identify decision making issues for the ER and other departments in a hospital. These other departments include operating rooms, ancillary departments and outpatient clinics. This search also provided information regarding some of the problems these departments face and the approaches managers have taken to solve them. The most common issues in the management of the latter departments include scheduling (personnel, surgeries, and appointments), budget decisions (purchase of supplies and equipment), and decisions about facility renovations (mainly expansions). Not surprisingly, managers of ERs also have to deal with personnel scheduling, evaluations of changes in operating policies, physical changes to the facility, and controlling medical supplies.

With this information, a questionnaire was developed for the interviews with ER managers at Mercy Hospital, PanAmerican Hospital, South Kendall Regional Hospital, and South Miami Hospital in Miami-Dade County. During the interviews, it was found that these managers also deal with issues not mentioned in the literature such as temporarily closing the facility, handling insurance claims, meeting certain government

standards, and addressing patient complaints. After the interviews, a more detailed questionnaire was developed and distributed among several ER managers nationwide. The objective was to increase the sample size of surveyed hospitals. This questionnaire followed the same types of questions asked during the initial interviews. In addition, it emphasized questions regarding specific decisions to be developed for the prototype of the proposed SB-DSS. Appendix A gives a sample of the survey conducted through e-mails. A summary of the ER decisions identified in the interviews is given in Table 1, whereas a summary of the ER decisions identified through e-mail surveys is given in Table 2. A complete summary of the interviews and e-mail surveys is given in Appendix C.

The answers obtained from the surveys resulted in the classification of ER decisions into three major areas or categories as shown in Figure 3. The areas identified are *human resources*, *facility operation policies*, and *cost and inventory control* activities. Decisions under the Human Resources category include management of physicians, nurses, technicians, and other staff. Examples of specific decisions under this category include staff scheduling, staff evaluation, and hiring of new personnel. The category of Facility Operation Policies includes determining when to temporarily close the facility, implementing changes to the operation of the ER, assessing the feasibility of adopting a new operating policy, and modifications to the physical structure of the facility. The category of Cost and Inventory Control includes purchasing equipment, purchase and control of medical supplies, and preparing the ER budget.

	H1	H2	H3	H4
<b>Areas:</b>				
<b>Human Resources</b>				
Staff Scheduling	✓	✓	✓	✓
Hiring new personnel	✓	✓	✓	✓
Evaluation of personnel	✓			
Control labor over time				✓
<b>Facility Operation</b>				
Temporarily close ER	✓	✓	✓	✓
Resolve patients complaints		✓	✓	
Adoption of new policies			✓	✓
Analysis on ER renovation			✓	
<b>Cost and Inventory</b>				
Inventory Control of medical supplies	✓			✓
Prepare budget	✓			
Equipment Purchase	✓	✓		
Purchase Office supplies/medical supplies				✓
Charging supplies to patients				✓
Medication distribution process				✓
<b>Other</b>				
Identify and address problems area: bottlenecks, procedures			✓	

Table 1: Interviews Summary of ER Decisions



	H5	H6	H7	H8	H9	H10
<b>Areas:</b>						
<b>Human Resources</b>						
Staff Scheduling	✓	✓	✓	✓	✓	✓
Hiring new personnel	✓		✓			
Evaluation of personnel	✓			✓		
Control labor over time						
<b>Facility Operation</b>						
Temporarily close ER	✓	✓		✓	✓	
Resolve patients complaints						
Adoption of new policies	✓	✓	✓	✓	✓	✓
Analysis on ER renovation	✓	✓	✓	✓	✓	✓
<b>Cost and Inventory</b>						
Inventory Control of medical supplies						
Prepare budget	✓	✓	✓	✓		✓
Equipment Purchase	✓		✓	✓		
Purchase Office supplies/medical supplies	✓	✓	✓	✓		
Charging supplies to patients						
Medication distribution process						

Table 2: E-mail Survey Summary of ER Decisions

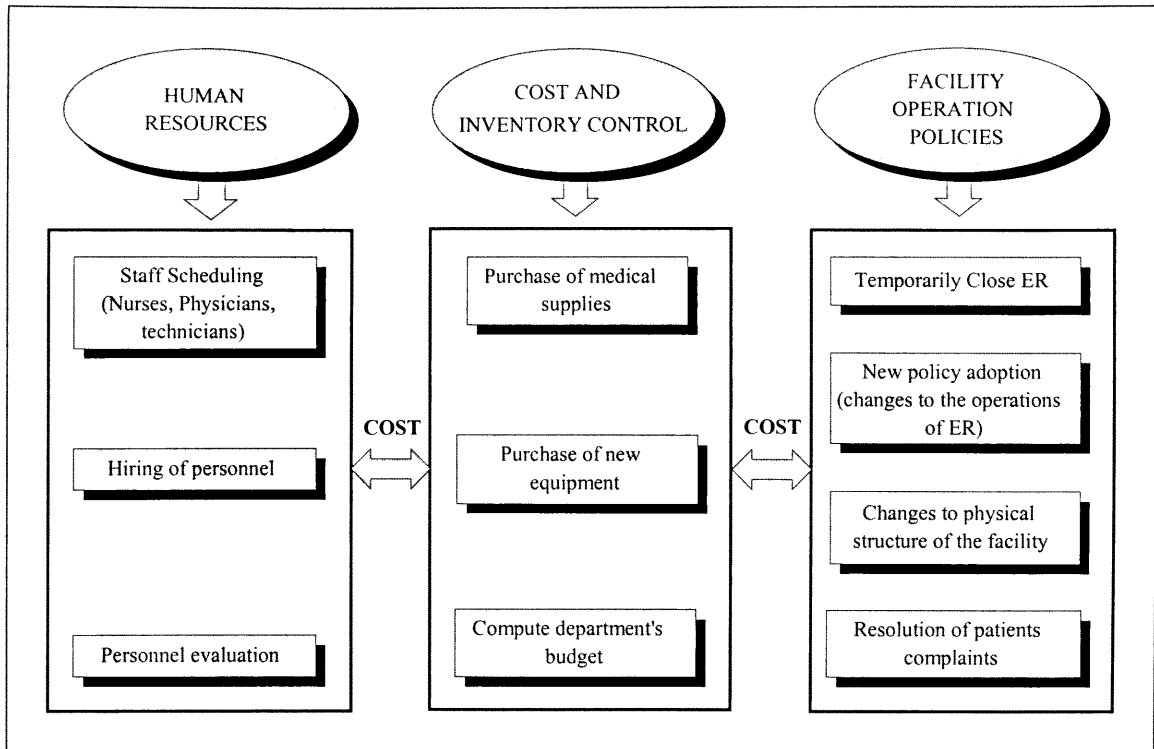


Figure 3: Classification of ER Decisions

These categories cannot be considered independent from one another. One of the common factors between them is cost. For example, the hiring of new personnel is dependent on the appropriated budget, and it will affect the schedule of resources. Other examples include the purchase of new equipment which influences the ER budget, and adopting a new policy. For the latter, let us say that a manager needs to make a decision on the adoption of a fast-track system. To evaluate this decision several factors must be considered such as the personnel needed to cover the hours of operation, the physical location of the fast-track within the facility, and the equipment needed.

A full scale DSS would provide support for all these types of decisions. However, to demonstrate the feasibility of the proposed system, it is not necessary to

implement them all. In fact, there are some decisions that belong to different categories; yet, they could be implemented simultaneously. In other words, a single routine could implement them. For example, the input data for decisions regarding hiring of personnel, implementing a fast-track, and adding capacity to the waiting area could be collected in a form similar to Figure 9 (in Chapter4).

Other decisions do require that they be implemented independently. To show how to implement one of such decisions a category was selected. The selected category was Facility Operation Policies and within this category, the decision of temporarily closing the facility was chosen. To fully integrate all of these categories, the following issues must be resolved for each one: 1) inputs and outputs required, 2) data sources, and 3) communication protocols.

Each category requires a different set of *inputs* to evaluate the set of decisions belonging to that category. Similarly, the *outputs* of each category will also be different; i.e. the decision-maker requires a different set of measures of performance to reach a decision. For example, in the category of human resources the SB-DSS requires that the schedules of the nurses, doctors and technicians be available; however, for the category of inventory, these schedules have no relevance whatsoever. On the contrary, this category requires information regarding departmental budgeting. These inputs will lead to different simulation outputs. For instance, schedule of nurses will lead to the gathering of resource utilization and the impact on flow time that the schedule has, whereas department budget will lead to the gathering of an upper bound on the number of resources for the facility.

Another issue is to clearly establish the *origin of the data* that will be used as

inputs. Because an ER facility is typically part of a hospital, 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 data extraction from the hospital's database, the SB-DSS own database, and from the user. They will also include off-line data extraction procedures from hospital's or external databases.

Table 3 to Table 5 list the inputs required for each decision category as well as the results expected in each category. In the case of Human Resources, simulation is used to evaluate how the management of human resources affects different operational parameters in the ER. It includes analyzing the impact of changes to the schedules of doctors, nurses, ER technicians, and triage and registration personnel on parameters such as average patient length of stay and resource utilization. Therefore, the system should be integrated with the scheduling/attendance system in the ER or be fed the appropriate schedules on-line. The system can also be used to observe the effects of staff not coming to work on a particular day given the current conditions. It would then suggest actions for management to take in order to maintain the systems level of performance or even improve it. The system can also be used to assess the impact of hiring new personnel on these same measures of performance.

In the case of Inventory Control, simulation can be used to control medical supplies inventory and to evaluate the effects of purchasing new equipment on systems performance. The system can be used to assess how changes in the way medical supplies

are ordered and distributed can reduce cost or if, on the contrary, will increase it. In many hospitals, a materials management department does the actual supply ordering. If this is the case, then the system should have either on-line or off-line access to data from the materials management system.

<b>HUMAN RESOURCES</b>	
<b>INPUTS</b>	<b>OUTPUTS</b>
<ul style="list-style-type: none"> <li>◆ Types of resources: nurses, doctors, technicians</li> <li>◆ Number of resources</li> <li>◆ Schedule of resources</li> <li>◆ Probability of failure for resource (sick calls, late arrivals)</li> <li>◆ Salaries of resource</li> <li>◆ Demand: patient arrivals</li> </ul>	<ul style="list-style-type: none"> <li>◆ Patient average length of stay</li> <li>◆ Resource utilization</li> <li>◆ Payroll cost</li> <li>◆ No. of patients attended/day</li> </ul>

Table 3: Human Resources: Inputs and Outputs

<b>INVENTORY CONTROL</b>	
<b>INPUTS</b>	<b>OUTPUTS</b>
<ul style="list-style-type: none"> <li>◆ Types of supplies</li> <li>◆ Cost of supplies</li> <li>◆ Demand for supplies</li> <li>◆ Number and type of medical equipment</li> <li>◆ Cost of medical equipment</li> <li>◆ Probability of equipment failure</li> </ul>	<ul style="list-style-type: none"> <li>◆ Cost of ordering policy</li> <li>◆ No. of patients attended/day</li> <li>◆ Patient average length of stay</li> </ul>

Table 4: Inventory Control: Inputs and Outputs

<b>FACILITY OPERATION POLICIES</b>	
<b>INPUTS</b>	<b>OUTPUTS</b>
<ul style="list-style-type: none"> <li>◆ Configuration of ER system</li> <li>◆ Current ER policies</li> <li>◆ Patient data</li> <li>◆ Patient Flow</li> <li>◆ Resource Data</li> <li>◆ Resource Flow</li> </ul>	<ul style="list-style-type: none"> <li>◆ No. of patients attended/day</li> <li>◆ Patient average length of stay</li> <li>◆ Patients not serviced</li> <li>◆ Resources utilization</li> </ul>

Table 5: Facility Operation Policies: Inputs and Outputs

The inputs required to decide on the control of medical supplies inventory include the types of supplies together with their cost and demand. Additionally, the system should have the capability to track down how these medical supplies are distributed among patients. It should keep track of the quantity and to whom these supplies are administered, so that the facility does not incur in losses due to supplies not being properly charged to patients. In this category the system can also be used to determine how purchasing new medical equipment, beds, monitors, etc can improve or deteriorate systems performance before actually buying it.

In the case of Facility Operation Policies the objectives of using simulation is to evaluate the effects of changes to the operation of the facility. Therefore, the system needs to hold information, either on the database or the model itself to represent the current operating policies of the facility. Examples of these policies include registration policies as well as admissions and transferring policies of patient from ER to the hospital. It needs to know the configuration of the ER system (number of resources, regular beds, fast-track or hallway beds, classification of patient types). It needs to know how patients flow through ER upon arrival, how resources work; i.e. how they are assigned to patients, the possible actions that can be taken by the manager when a problem arises (need to temporarily close facility, patients waiting in ER for times longer than usual). By using simulation, the system can also be used to identify problem areas such as long registration process, delays waiting to be admitted to the hospital, delays waiting for laboratory results, etc.

Advances in simulation technology enable the modification of the simulation model at run time. Hence, modifying the flow of patients is feasible. In summary, the

design of the framework is modular and inclusive to allow support of these decision categories, as well as the integration of existing technologies. If there is a need for adding a new category of decisions, some common issues need to address to include it once a full-scale SB-DSS has been developed (Figure 4). First, the input factors that go into evaluating that particular decision as well as the measures of performance desired by the user are to be established. Similar to how it is done with the initial set of decisions, the data acquisition and the communications protocols for each new decision are also to be established. If the need is to add only one decision then the same process should be followed but starting from point A in Figure 4, but with  $n = 1$ . Then, the type of simulation model to be used, in the case the SB-DSS has a model library, is to be identified. If such model does not exist then it needs to be built.

It's important to note that there are certain decisions that are more difficult to implement than others, for example any decision that will change the flow of operations in the ER will require new simulation components. Hence, these decisions require significant simulation savvy.

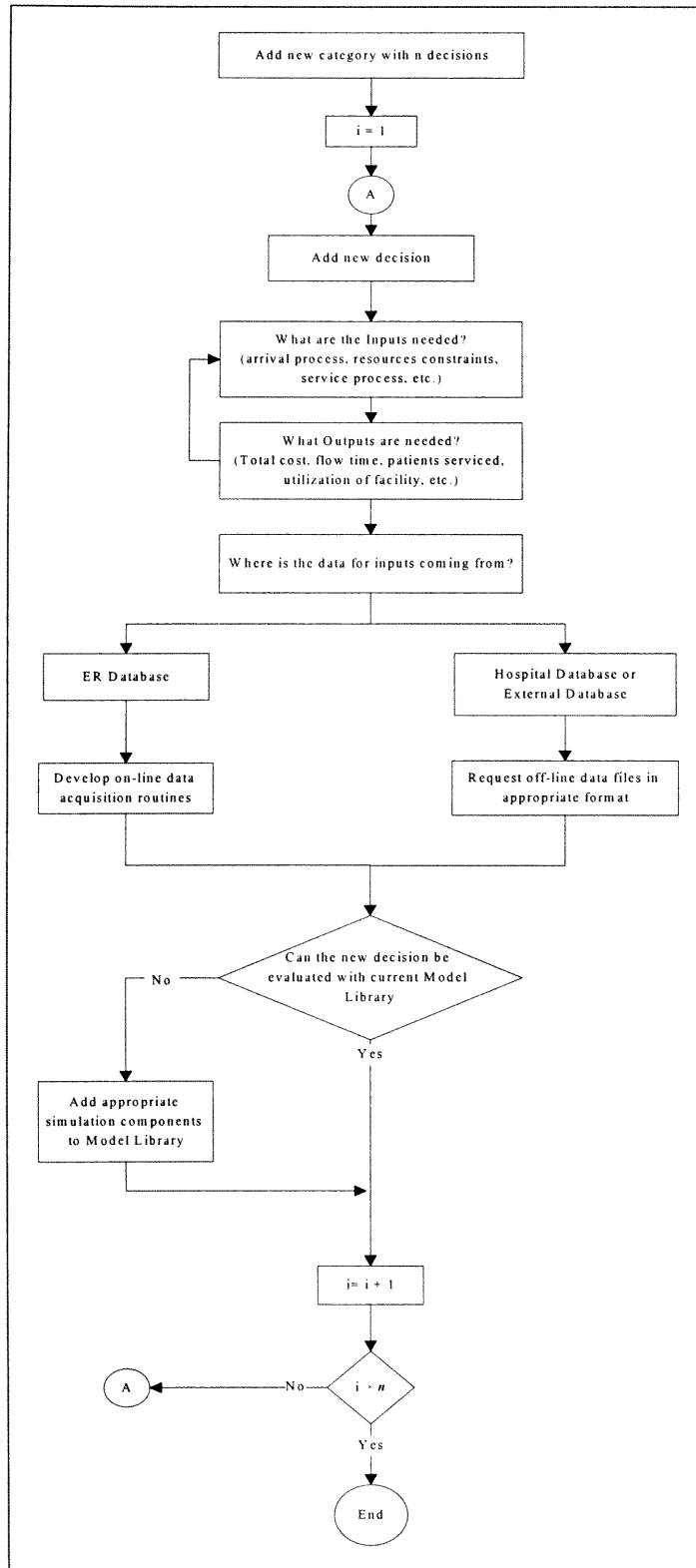


Figure 4: Process of adding a new category/decision to SB-DSS



### 3.2 Data Acquisition

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

As mentioned before, each category of decisions requires a different set of inputs. 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. ER databases typically hold patient charts, payroll data, and staff schedules. Hospital's databases contain more clinical and administrative data. These databases are usually located at separate places and are stored on different hardware and software platforms (Forgionne and Kohli, 1996). This fact was confirmed during the surveying process because it was found that in many instances, the ER computer system is not integrated with the one for rest of hospital's. Consequently, the data must be transformed into a proper format through routines compiled in a *data transformation library*. Most simulation packages are capable of reading data from ASCII files, spreadsheets, and relational databases through SQL. Thus, in some instances a direct link between the simulation model and the database may be established. In any case, implementing a full scale SB-DSS will require a library of routines written in a programming language such as Visual Basic. These subroutines must be capable of transforming the extracted data from the database and the user into the proper format required by the system.

The routines should generate a collection of data files that contain values such as time between arrivals, type of patients, service times, and so forth. These values are to be

used by the SB-DSS to estimate a probability function for the arrival process, the service process and so forth. These files must be processed using a statistical package or using any support tool that comes with the simulation package. Software technology has evolved to a point in which programming statistical routines from scratch is a waste of time. Most statistical packages can be accessed by simulation packages; some may be even embedded. Hence, processing the data to establish the inputs for the model should not be difficult.

Another input required by the simulation model is the configuration of the facility. The configuration of the facility consists of the number of resources and schedules. This information must be given to the SB-DSS once, and then changed only when needed. There are two ways of achieving this: 1) to create a configuration file (ASCII), or 2) to modify the appropriate objects in the simulation model. For the first case, any text editor can be used to create the file, whereas for the second case an interface may be developed to collect the data and modify the simulation model. In the latter case, the user will interact with such interface on a *as-needed* basis.

### **3.3 Processing and Modeling Engines**

The data processing and modeling engines section has three modules: a simulation model, a statistical analysis tool, and a library of subroutines to perform other analysis and evaluations.

A valid and general simulation model of an ER must be built and placed at the heart of the SB-DSS. When developing the simulation model, two approaches can be

followed: 1) built a single model that represents all the operations of an ER, or 2) have several smaller models to be used according to the decision being evaluated. The first requires defining multiple entities (patients, medical supplies, etc), multiple resources (staff, beds, medical equipment, patients, etc.) and the interactions between these to build a single model. The latter approach consists of building different smaller and simpler models that would be called upon and executed according to the decision being evaluated. These models are stored in a model library. The advantage of using the second approach lies in the reduction in execution time.

When using simulation as an analysis tool, it is of utmost importance to utilize valid input data in order to obtain valid results. Similarly, the outputs from the simulation modeling need to be statistically analyzed in order to draw valid conclusions. Therefore, it is important to have tools capable of statistically analyzing both the input and output data. In the case of the input data, a tool would be responsible for determining distribution for time between arrivals, service times, etc. For the case of simulation output data, a tool would be in charge of constructing confidence intervals, and allowing the user to perform hypothesis testing, ANOVA analysis, correlation analysis etc. A separate statistical analysis packages such as Statistica, SPSS and ExpertFit may be integrated into the system to accomplish this task. In any case, the analysis of the data should be as automated as possible. Any of these programs may be called from within the SB-DSS transparently to the user. Only relevant results should be passed or transferred on to the model. For example the best fit data distribution of data representing the time between patient arrivals can be found by ExperFit. Only the expression of this distribution should be passed to the model.

Another alternative is to use the input analysis capabilities that come within many of the commercial simulation software packages. ARENA for example comes with an Input Analyzer, a tool which allows to perform best fit distribution analysis, and that is also able to build histograms with the same data. The advantage of using this approach is that it facilitates the integration between the system components due to these programs being developed with the same language. The user interface together with the routines in the data transformation library should be enough to collect information with respect to the types of analysis desired by the user and required by the system.

The function of the module labeled other analysis and evaluations is to execute a set of algorithms that will allow carrying out other types of analysis such as post-simulation optimization and cost-benefit analysis. The processing of these algorithms can be accomplished through the use of available commercial software or a model library. A model library consists of a series of routines of mathematical models and algorithms, which are stored in subroutines within the system code. These subroutines are responsible for performing calculations from the fields of economic analysis, financial decisions, and mathematical programming. Examples of packages are CPLEX, MATHEMATICA, etc.

### **3.4 Outputs and Results**

This section presents the results from the simulation experimentation, statistical analysis, and other types of analysis and evaluations. Results are presented to the user via a customized user interface. Specific graphs, tables and charts will vary depending on the

type of decision being evaluated. External applications, such as Excel, may be launched by the SB-DSS to create charts and graphs from the simulation output data. An application to generate reports may also be included.

### **3.5 Communication Protocols**

For the system to properly work, there needs to exist smooth communication between its components. The ease of communication will depend on the programming tools selected to develop each functional section of the SB-DSS. The tools for manipulating, retrieving, and analyzing data, as well as for simulating, displaying results and developing the user interface need to be set. Two main approaches can be followed to accomplish this task. Different and separate applications may be chosen to perform the different functions in each section. For example, ExperFit or Statistica can be used for data analyses, ARENA/SIMAN or ProModel for simulation, and Excel for creating graphical displays. Then, a programming language can be used to integrate all of these components and to develop the user interface. The second approach consists of using a programming language embedded in the applications to develop both the user interface and the communication protocols between systems components. The latter approach was used in this effort.

Visual Basic for Applications (VBA), which comes embedded in ARENA/SIMAN was used to develop the user interface and integrate the systems components. This package presents several advantages, which makes it suitable for developing the prototype. 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. The events triggered by ARENA include RunBegin, RunEnd, RunEndReplication, RunEndSimulation, and RunPause.

The operation of analyzing the input data can be achieved using any available software. To be consistent with the programming tools used and facilitate the communication and transferring of data, the Input Analyzer tool from ARENA was chosen for this task. The Input Analyzer application is launched by a VBA routine when necessary. Once the proper analysis is done, without the user having to work directly in this application, relevant results are collected back by a routine. These results are then transferred to the simulation model, used for other analysis or they are presented to the user via the user interface.

In summary, close attention must be paid to how the various tools will interact. All chosen tools must support DLL, DOE, or similar data exchange facility.

## CHAPTER 4:

### A PROTOTYPE SB-DSS

A prototype of the system was developed to demonstrate the feasibility of using simulation as an integral part of a SB-DSS in the context of healthcare. This chapter describes the specifications of the prototype. A description of the overall specifications is given, followed by descriptions of each functional section. The last section in the chapter provides two sample scenarios.

#### 4.1 General Specifications

The first thing that was decided was the category of decisions. Given the availability of data the *Facility Operating Policies* category was chosen. Within this category, the decision of *Temporarily Closing* the ER was selected because managers must evaluate constantly if the ER should be temporarily closed when it is almost full. That is, managers must decide if they continue or stop accepting patients with life threatening conditions that are brought to the facility by ambulance. The criteria for temporarily closing or bypassing the facility varies among hospitals. In general, the ER is considered to be full when almost all beds are occupied, there are no more beds with a monitor, or the waiting areas are full. Other criteria include high volume of high acuity patients, not enough resources to safely take care of patients, the total ER volume exceeds a specified number or waiting times exceed a specific amount of time. Moreover, if these conditions are suspected to extend over a significant period of time, the manager is forced

to bypass the facility in order 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. The facility stays officially on bypass for a specific period of time. Once this time has elapsed, the manager needs to advise the authorities if the ER is still closed or if it has re-opened. In Miami-Dade County, this time is four hours. If the authorities are not advised of the status of the ER four hours after closing, they assume that the facility is no longer on bypass.

The actual number of times or total hours the ER is closed also varies among hospitals. From the surveyed hospitals, it was found that certain hospitals close their ER from eight to ten times a month. In one case, the facility was closed, at least once a day, twenty days in a month. A factor that influences the amount of time the ER is closed is the location of the hospital, i.e. a big city versus a small town. Even though this variation exists, managers still need to evaluate this situation on a daily basis. Therefore, it is important for them to assess when to close the facility and what actions to take. Although the ER itself does not occur into any kind of penalties for closing the facility, there is an opportunity cost to the institution as well as to the doctors. This prototype allows managers to predict if the ER needs to be closed at a given time if the facility's current conditions are expected to stay constant.

The prototype was developed using VBA embedded in the simulation package ARENA/SIMAN. This selection was based on the fact that VBA looks at the simulation model as an object which experiences events. Examples include *Run\_Begin* and *End\_Replication*. The object *This Document* represents the ARENA model. Through the events experienced by *This Document*, it is possible to collect the configuration of the ER



and the current conditions of it.

The user interface is the only means of communication between the user and the system. The user interface was built using VBA. Using VBA facilitates the acquisition of information from the user through a user-friendly interface. The function of these screens is to collect data from the user for the SB-DSS to work. In VBA, the object *form* is equivalent to the screen or window seen by the user. In this document, the words form and screen will be used interchangeably to refer to the screens seen by the user. In this section the word system is also used to refer to two types of systems: the SB-DSS and the ER system. The decision support tool (SB-DSS) will be referred to simply as the system, whereas the latter one will be referred to as the ER system.

## **4.2 Inputs Section**

The ideal system consists of an ER database developed in a DBMS such as ACCESS or of a series of spreadsheet files. For the prototype, Excel tables were used to store the data required by the system. These tables are designed so that information can be gathered in almost real time on each patient arriving to the facility. They are also used to store resource data such as service times, schedules, and actual number of hours worked. The information is then fed to the simulation model via the Data Transformation Library routines. The objective of using such a database is to allow ER staff to input the times of arrivals of the patients, acuity level, how long they spend with the nurse or physician, and the time of departure from the facility. This data is then saved into a file so that the data can be read and manipulated by the SB-DSS.

In a previous effort, Reyes (1998) collected and analyzed data required to run a simulation model of the ER at Mercy Hospital in Miami. The collected data is equivalent to the data from patient charts for one-week period (Monday through Sunday). This data was used to test the model of the prototype. Other input data such as the number of human resources and their schedules, number of beds, queue capacity etc was gathered through the surveying process. In summary, the inputs section of the prototype is used to collect the time for evaluating decision, the type of data to be used by the system (historical versus same day data), and the outputs of interest to the user.

The data acquisition process is given in Figure 5. After pressing the run begin button, the first form (*Welcome* form) is loaded into the screen. Here, the user is given the option of reconfiguring the ER system, or continuing to the next form with the current configuration. If the user chooses to configure the system then the *Configuration* form is shown. Upon loading this form, the user is presented with the current input parameters of the ER system such as number of beds, nurses, doctors, and Waiting Room capacity. The user is allowed to make changes to these parameters, which are then reflected in the simulation model itself. On the other hand, if the user does not choose to configure the system, the model will run with the parameters last used.

Afterwards, the *Main Menu* form of the DSS is shown. The main menu presents three categories of ER decisions for the user to choose from staff management (human resources), facility operation (facility operation policies) and inventory control. Within each category, the user can select among different decisions. The prototype was developed for the case when the decision of closing the ER under the Facility Operation category is chosen. Once the user selects this option the *Close ER* form is shown.

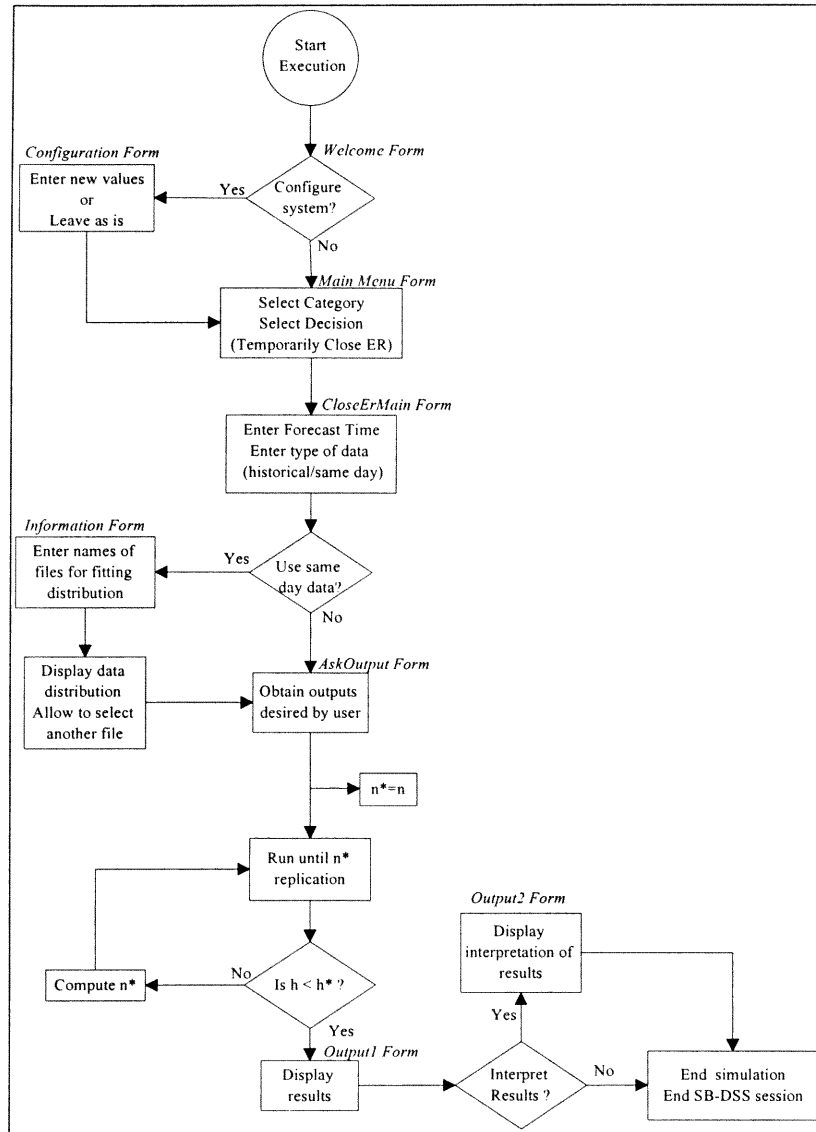


Figure 5: SB-DSS User Interaction

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. The simulation length (SL) is calculated as follows

$$\begin{aligned}
 &= \text{Warm-up time} + \text{Simulation time} + \text{safety factor} \\
 &= [60(T_{CH} - T_{DH}) + (T_{CM} - T_{DM})] + [60(T_{FH} - T_{CH}) + (T_{FM} - T_{CM})] + W_B
 \end{aligned}$$

where,

$T_{CH}$  = Current time when decision is being evaluated (hour)

$T_{CM}$  = Current time when decision is being evaluated (minutes)

$T_{FH}$  = Time at which the decision of closing the ER is being considered (hour)

$T_{FM}$  = Time at which the decision of closing the ER is being considered (minutes)

$T_{DH}$  = Time from when data was collected (hour)

$T_{DM}$  = Time from when data was collected (minutes)

$W_B$  = Safety factor

The times are first collected in military times and are then converted into minutes. In addition the simulation length also contains a few extra minutes that serve as a warm up period. This period is taken into consideration because it cannot be assumed that the status of an ER is empty at the beginning of the simulation.

For the case that the user selects to use historical data, the system will take him/her into the *AskOutput* form. 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 and the contents of it. 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 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.

### **4.3 Processing and Modeling Engines**

A crucial step in developing the prototype of the SB-DSS was to build a simulation model of the operation of an ER facility. The flow of patients through the ER of the prototype is presented in Figure 6.

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 patient, 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. Table 6 presents a description of patients priorities. 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 hours a day, in other cases they operate under a fixed schedule. For the prototype, 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. The tool used for building and operating such a model is SIMAN/ARENA. Appendix B presents the model and experimental frames of the simulation model of the SB-DSS.

One of the characteristics of the model built for the prototype is that it is 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.

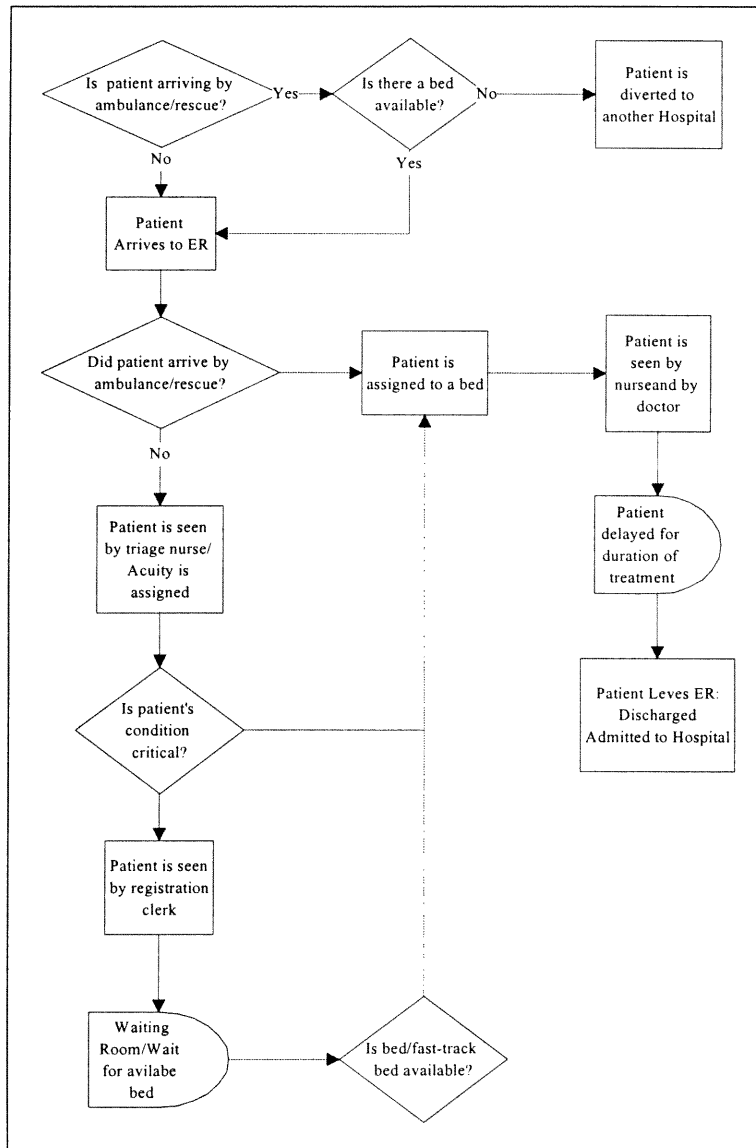


Figure 6: Patient Flow through ER

Priority	Description
1	Life threatening condition arriving by ambulance/rescue car
2	Life threatening condition arriving by car
3	Patients who can go without treatment for 1 hour
4	Patients who can wait 4-6 hours without being treated
5	Patients who if not treated can leave and come back later

Table 6: Patient Categories

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  $(1-\alpha)\%$  confidence interval, around the mean of the measures of interest, is computed and compared to the half width desired by the user ( $h^*$ ). 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  $h^*$  are calculated ( $n^*$ ) and the model is run once again for that many replications. However, if  $n^*$  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  $h^*$ , but it



would require a higher execution time, which is of utmost importance when deciding what actions to take in the immediate future.

#### **4.4 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, by executing VBA routine. 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, there are no resources to safely care for additional patients, patients waiting times exceeds a fixed amount of hours, and there are no more beds left unoccupied.

#### **4.5 Testing and Validation**

To assess the validity of the prototype, its components had to be tested. The simulation model as well as the data acquisition capabilities, data transformation routines, and input and output analysis routines were validated, by separating each module from the SB-DSS and treating it as a stand-alone tool. For example, the simulation model was run in "design mode" from within ARENA. This is equivalent to running the model detached from the SB-DSS by disabling the VBA integration. The results obtained from this run were compared to the results obtained from running the SB-DSS under the same experimental conditions, i.e. same number of resources, same simulation length, same number of replications. Table 7 shows the results from this validation by comparing the outputs yielded by both approaches on some common measures of performance. The statistical analysis component was tested in a similar way. Using Arena's Input Analyzer, a best fit was performed on the data file called "arrivals.txt". The resulting distribution was **1 + ERLA(1.43, 3)**. Using the SB-DSS to open the same file and fit a distribution to it, the result is a **1 + ERLA(1.43, 3)**. This test also shows the validity of the communication protocols. Information collected through the user interface is properly communicated from one tool to the other. Hence, the proper file is opened and the proper

operation is performed: a distribution is fitted to it.

Measure of Performance	Simulation by itself	Simulation within SB-DSS
No. of beds busy	8.35	8.35
Beds not being occupied	5.65	5.65
Bed Utilization	.596	.5964
Total No. of patients in ER	11	10.95
Total No. of patients in WR	1.71	1.71

Table 7: Simulation Results

#### 4.6 Sample Scenarios

It is 4:00 P. M. It is been a busy day since early morning at the ER of the XYZ Hospital. The ER is becoming full and these conditions (patient arrival and high acuity levels) are expected to stay constant. The manager of this facility would like to evaluate if the facility should go on bypass in two hours. The manager needs to make the decision based on the number of beds not being occupied.

Two scenarios illustrating how the prototype of the SB-DSS works are described in this section. In the first scenario, the user reconfigures the system because he/she wants to evaluate the levels of the ER if it were to have 1 additional regular bed. Historical data will be used to make this evaluation. In the second scenario, the user does not reconfigure the system, but this time he/she wants to use the same day data for arrivals between patients, to make the evaluation.

For the first scenario, the *WelcomeForm* appears (Figure 7). The user needs to click on the **Continue** button in order to continue to the main menu. In the *MainMenu*

*Form* (Figure 8), the user must select the *Configure ER* option to change the number of beds. Once in the *Configuration Form* (Figure 9), the user needs to enter the new number of beds (14) in the box labeled **Number of beds**. Then, the **Continue** button must be clicked to go back to the *MainMenu Form* (Figure 8).

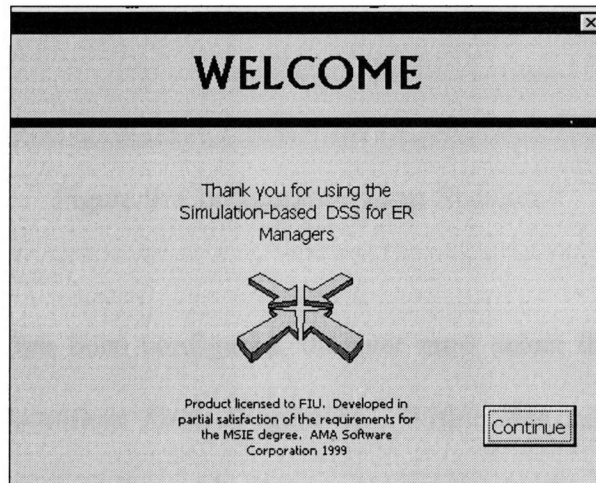


Figure 7: Welcome form-Scenario1

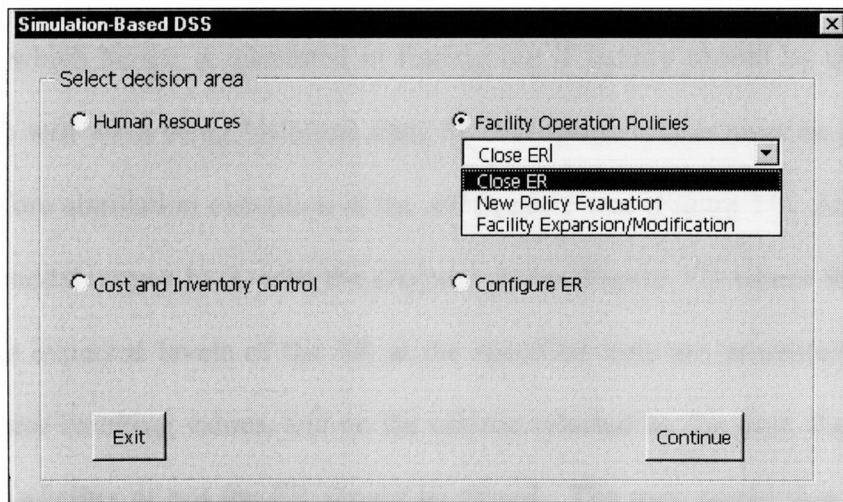


Figure 8: MainMenu form-Scenario 1

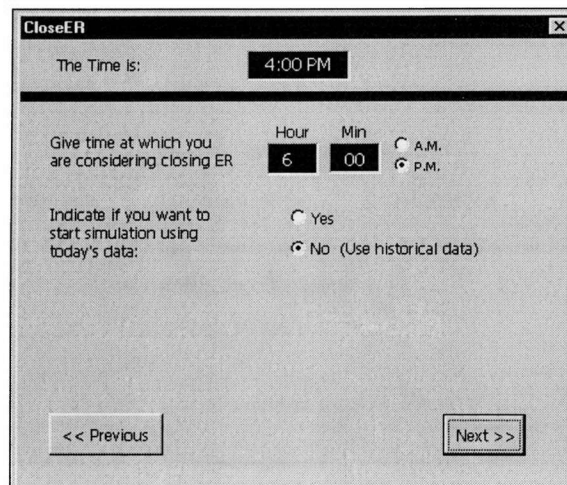
The following information is required if you are using this package for the first time or if you want to make changes to the current configuration of your system

Number of Registration Clerks	<input type="text" value="2"/>	Number of beds	<input type="text" value="14"/>
Number of Triage Nurses	<input type="text" value="1"/>	Waiting Room Capacity	<input type="text" value="30"/>
Number of Nurses	<input type="text" value="6"/>	Is there a fast-track	<input checked="" type="radio"/> Yes <input type="radio"/> No
Number of doctors	<input type="text" value="2"/>	How many fast-track beds?	<input type="text" value="3"/>

Figure 9: Configuration form-Scenario1

Once the ER has been configured, the user must select the **Facility Operation** category from the *MainMenu Form* (Figure 8). Within this category, the **Close ER** decision must be selected. Afterwards, the user must click in the **Continue** button. The next form to appear is the *CloseER Form* (Figure 10). The user must specify the time of the day for which he/she is interested in finding out if facility should be closed. Since this scenario will work using historical data, the **No** option button must be clicked. The last step before simulation execution is the *AskOuput Form* (Figure 11). At the end, the simulation model comes back with the *Output1 Form* (Figure 12) where the results are shown. The expected levels of the ER at the specified time are presented to the user. Based on these resulting values, and on the criteria selected by the user, the system will recommend whether or not the ER should be closed. The user would then click on the **Continue** button to further analyze the response. At this point the *Output2 Form* Figure 13 appears and the user only needs to click on the **The End** button to finish the SB-DSS

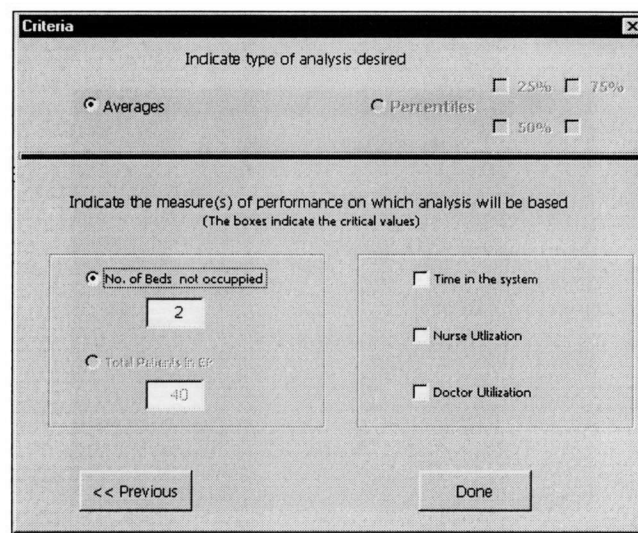
session



The CloseER form is a window titled "CloseER" with a close button (X) in the top right corner. It contains the following elements:

- A label "The Time is:" followed by a text box containing "4:00 PM".
- A section titled "Give time at which you are considering closing ER" with two spinners for "Hour" (set to 6) and "Min" (set to 00). To the right are radio buttons for "A.M." and "P.M.", with "P.M." selected.
- A section titled "Indicate if you want to start simulation using today's data:" with radio buttons for "Yes" and "No (Use historical data)". The "No" option is selected.
- At the bottom, there are two buttons: "<< Previous" on the left and "Next >>" on the right.

Figure 10: Close ER form-Scenario1



The Criteria form is a window titled "Criteria" with a close button (X) in the top right corner. It contains the following elements:

- A section titled "Indicate type of analysis desired" with two radio buttons: "Averages" (selected) and "Percentiles". To the right of the "Percentiles" radio button are four checkboxes: "25%", "75%", "50%", and an unlabeled one.
- A section titled "Indicate the measure(s) of performance on which analysis will be based (The boxes indicate the critical values)". This section contains two groups of options:
  - On the left, a radio button for "No. of Beds not occupied" (selected) with a spinner box set to "2", and a radio button for "Total Patients in ER" with a spinner box set to "40".
  - On the right, three checkboxes: "Time in the system", "Nurse Utilization", and "Doctor Utilization", all of which are currently unchecked.
- At the bottom, there are two buttons: "<< Previous" on the left and "Done" on the right.

Figure 11: AskOutput form-Scenario1

**Predictions**

THESE ARE THE PREDICTIONS FOR THE ER AT TIME

All predictions are given as averages

Number of Beds Busy	8.35	Beds not being occupied	5.65	Number of FT Beds Busy	0.90
Total Patients in ER	10.95	Waiting Room Level	1.71		

Utilizations are given as percentages

Bed Utilization	59.64	Nurse Utilization	19.68	Doctor Utilization	16.84
-----------------	-------	-------------------	-------	--------------------	-------

---

IF CURRENT CONDITIONS REMAIN CONSTANT

**ER SHOULD NOT BE CLOSED**

*ER SHOULD BE CLOSED*

Click here for further analysis on response

[Continue](#)

Figure 12: Output1 form-Scenario

**Output Analysis**

95% Confidence Interval on the mean.

( 9.04, 12.86)	Number of Patients in ER
(4.86, 6.44)	Empty Beds

[The End :-\)](#)

Figure 13: Ouput2 form-Scenario 1



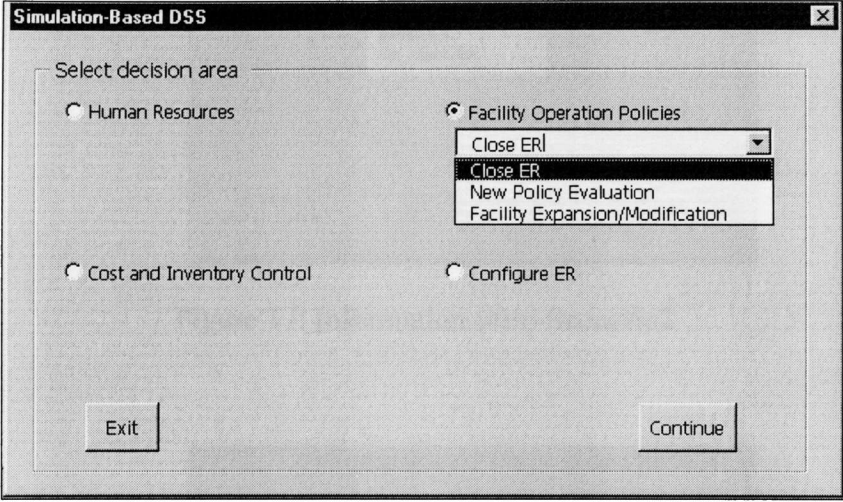
Figure 14: Welcome form-Scenario2

For the second scenario, the first screen to appear is the *WelcomeForm*. In this case, the user must click the **Continue** button since he/she is not interested in reconfiguring the system Figure 14. In the *MainMenu Form* the user must select the **Facility Operation** category . Within this category, the **Close ER** decision must be selected. Afterwards, the user must click on the **Continue** button Figure 15. The next form to appear is the *CloseER Form* Figure 16. The user must specify the time of the day for which he/she is interested in finding out if facility should be closed. Since this scenario will work using the same day data, the **Yes** option button must be clicked. In addition, the time from when data is to be collected for that particular day must be specified. The *Information Form* appears next if all the sections have been properly completed in the previous form Figure 17. Here, the user needs to specify **arrivals** as the name of the file to be analyzed. Upon clicking on the **Next 1** button, the distribution of the data in this file will be determined. Internally, the Input Analyzer application is launched and the expression of the distribution is obtained. The *Information Form*



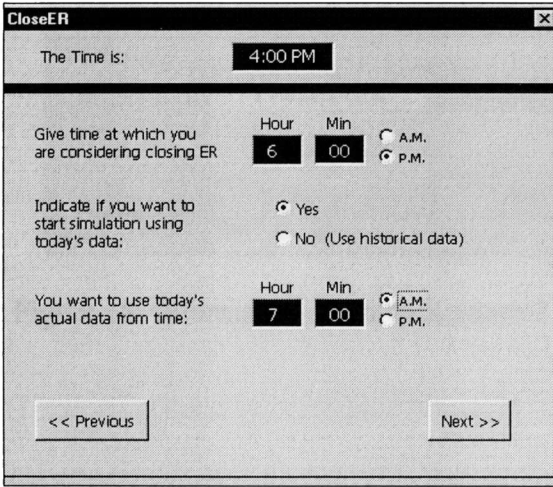
appears one more time Figure 18. The expression of the distribution is shown to the user.

The user must then click on the **Next 2** button.



The image shows a window titled "Simulation-Based DSS" with a close button in the top right corner. Inside the window, there is a section labeled "Select decision area" containing four radio buttons: "Human Resources", "Facility Operation Policies", "Cost and Inventory Control", and "Configure ER". The "Facility Operation Policies" radio button is selected. Below it, a dropdown menu is open, showing four options: "Close ER", "Close ER", "New Policy Evaluation", and "Facility Expansion/Modification". At the bottom of the window, there are two buttons: "Exit" on the left and "Continue" on the right.

Figure 15: Main menu form-Scenario2



The image shows a window titled "CloseER" with a close button in the top right corner. Inside the window, there is a section labeled "The Time is:" with a text box containing "4:00 PM". Below this, there is a section labeled "Give time at which you are considering closing ER" with two spin boxes for "Hour" (6) and "Min" (00), and two radio buttons for "A.M." and "P.M." (P.M. is selected). Below this, there is a section labeled "Indicate if you want to start simulation using today's data:" with two radio buttons: "Yes" (selected) and "No (Use historical data)". Below this, there is a section labeled "You want to use today's actual data from time:" with two spin boxes for "Hour" (7) and "Min" (00), and two radio buttons for "A.M." and "P.M." (P.M. is selected). At the bottom of the window, there are two buttons: "<< Previous" on the left and "Next >>" on the right.

Figure 16: Close ER form-Scenario2

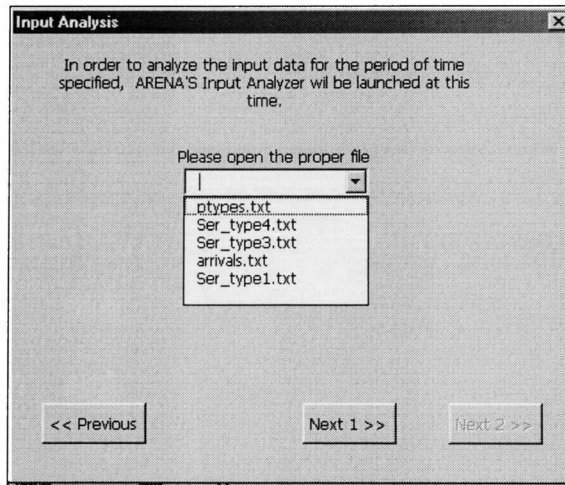


Figure 17: Information form-Scenario2

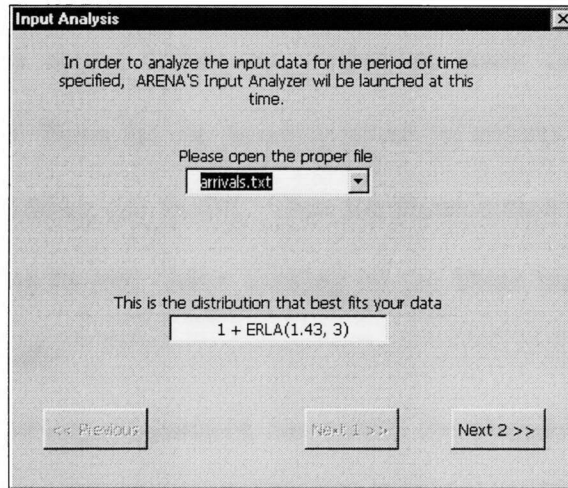


Figure 18: Information form(b)-Scenario2

Figure 19: AskOutput form-Scenario2

The last screen to appear before the evaluation starts to take place is the *AskOutput Form* Figure 19. Here, the user needs to select the criteria in which the system will base the decision of closing the facility. Then the **Done** button must be pressed for the SB-DSS to begin doing its job. After clicking on the **Done** button, the simulation model begins to run internally.

Once the simulation experimentation has ended, the *Output1 Form* is displayed Figure 20. The expected levels of the ER at the specified time are presented to the user. Based on these resulting values, and on the criteria selected by the user, the system will recommend whether or not ER should be closed. The user would then click on the **Continue** button to further analyze the response. At this point, the *Output2 Form* appears Figure 21. The user only needs to click on the **The End** button to finish the SB-DSS session.

**Predictions**

THESE ARE THE PREDICTIONS FOR THE ER AT TIME

All predictions are given as averages

Number of Beds Busy	12.34	Beds not being occupied	1.66	Number of FT Beds Busy	2.20
Total Patients in ER	29.15	Waiting Room Level	14.63		

Utilizations are given as percentages

Bed Utilization	88.14	Nurse Utilization	37.34	Doctor Utilization	35.18
-----------------	-------	-------------------	-------	--------------------	-------

---

IF CURRENT CONDITIONS REMAIN CONSTANT

**ER SHOULD NOT BE CLOSED**

*ER SHOULD BE CLOSED*

Click here for further analysis on response

[Continue](#)

Figure 20: Output1 form-Scenario2

**Output Analysis**

95% Confidence Interval on the mean.

( 28.03, 30.28)	Number of Patients in ER
(1.45, 1.87)	Empty Beds

[The End :-\)](#)

Figure 21: Output 2 form-Scenario2

## **CHAPTER 5:**

### **CONCLUSIONS AND FURTHER WORK**

The objective of this effort was to design a DSS centered on a simulation model. A major concern of such approach was the execution of the simulation model. Other issues regarding integration of heterogeneous tools as well as regarding decision making had to be understood.

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 require to run and evaluate that particular decision together with the simulation outputs that would aid the managers in making the decision.

One of the stated disadvantages of using simulation in healthcare has been that execution time of these types of models takes too long for them to be useful. This effort demonstrated that simulation can indeed be used in a timely fashion to evaluate decisions in the ER environment. An algorithm to automatically calculate the number of replications for the simulation model has been implemented. The significance of it is that the DSS can achieve statistical significance on its own, in a timely fashion.

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.

This effort has shown it is possible to acquire data from the user and transfer it to other components such as the simulation model and statistical analysis tool. This is a major contribution because it demonstrates that simulation models can be modified at run time. Other efforts have played with changing input parameters at run time, so does this. However, the innovative contribution of this effort is that elements of the model, not only parameters, can also be added or deleted as needed at run time.

In conclusion, the findings of this research may be summarized as follows:

- The various type of decisions that occur in ER systems have been identified and classified.
- Development of a DSS centered around a simulation model is feasible for ER systems.
- Statistically significant results can be obtained in a timely basis.
- The interaction between the simulation model and an ER's information system is possible, and it can be on-line and fully automated.
- Development of a flexible simulation model is feasible because the model can be modified at run time without its user knowing anything about simulation.

Despite the success of this effort, there are still several actions that need to be

done. For instance, for the prototype, only one decision was selected. Other types of decisions need to be developed and integrated into the SB-DSS. A more extensive survey of decisions made in ER systems needs to be made to refine the classification given here. The survey conducted for this study was limited in the number of responses due to time constraints and to the suspicious attitude of some of the hospitals contacted. A more extensive study of ER systems configuration is needed to clarify whether generic simulation models can be developed. A study on the various types of statistical support tools to establish a standard communication protocol may be developed.

Another interesting area for further research is the inclusion of neural nets into the framework of the SB-DSS. By using neural nets, the system can be trained to identify those scenarios it has evaluated previously. This potentially offers the ability of adding a great variety of scenarios and configurations, which the system could analyze without needing new programming. The key issue here is the ability of neural nets to be trained. Several questions may need to be answered. For example, how would the neural net be trained? Will it be trained with real world data? or Could it be trained with a simulation model? What would the interaction between the neural net and the simulation model be? Answers to these questions would add significant flexibility and reliability to a full scale SB-DSS.

In summary, further research areas of this effort include:

- A more extensive survey process of decisions made in ERs.
- Development and integration of other decisions into the SB-DSS.
- Extensive study of ER configurations in trying to develop a generic model.
- Inclusion of neural nets in the framework of the SB-DSS.

## REFERENCES

- Aggarwal, A. K. I. 1990. "Simulation-Based DSS." *Simulation Series*. 21(4). 143-146.
- Aasen, O., K. Liestol, A. Rollag, and J. Erikssen. 1993. "Decision Support by Computer Analysis of Selected Case History Variables in the Emergency Room Among Patients with Acute Chest Pain." *European Heart Journal*. 14. 433-440.
- Applegate, L. M. and K. Pearlson. 1995 "Mrs. Fields, Inc., 1977-1987.
- Badri, M., D. L. Davis, and D. F. Davis. 1990. "Simulation Modeling for Decision Support: A Model Base Development for a Carpet Manufacturing Company." *Simulation Series*. 21(4). 55-59.
- Banks, J., J. Carson, and B. Nelson. 1996. *Discrete Event System Simulation*. Upper Saddle River, NJ: Prentice-Hall, Inc.
- Beard, F. R. 1998. "Decision Support System for Scheduling and Evaluation of Turfgrass Equipment." *Applied Engineering in Agriculture*. 14(2). 201-207.
- 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.
- Dankel, A. and A. Gonzalez. 1993. *The Engineering of Knowledge-based Systems: Theory and Practice*. Englewood Cliffs, NJ: Prentice Hall, Inc.
- De Vlas, S., L. Fransen, J. K. Habbema, J. O. Ndinya-Achola, C.P. Van der Ploeg, G.J.



- Van Oortmarssen, and C. Van Vliet. 1998. "STDSIM: A Microsimulation Model for Decision Support in STD Control." *Interfaces*. 28(3). 84-100.
- Derks, J., H.M.J. Goldschmidt, A. Hasman, B. Schoenmaker, and G.G. van Merode. "Advanced Management Facilities for Clinical Laboratories." *Computer Methods and Programs in Biomedicine*. 50. 195-205
- Dumas, M. B. 1985. "Hospital Bed Utilization: An Implemented Simulation Approach to Adjusting and Maintaining Appropriate Levels." *Health Services Research*. 20. 42-61.
- Ferri, F., R. Maceratini, M. Rafanelli, and G. Sindoni. 1995. "An Object Oriented Decision Support System for the Planning of Health Resource Allocation." *Computer Methods and Programs in Biomedicine*. 48(1-2). 163-168.
- 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.
- Hakes. B., L. Keller, W. Lilegdon, J. Lowery, and K. Mabrouk. "Barriers to Implementing Simulation in Health Care." *Proceedings of the 1994 Winter Simulation Conference*, ed. J. D. Tew, S. Manivannan, D. A. Sadowski, A. F. Seila, 868-875.
- Hammer, L., H. W. Ittmann, and Y.E. Walus. 1997. "Decision Support Systems in

- Health Care.” *Methods of Information in Medicine*. 36. 82-91.
- Kelton, W. D. and A. M. Law. 1991. *Simulation Modeling and Analysis*. New York, NY: McGraw-Hill, Inc.
- 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.
- Kuljis, J. and R. Paul. 1995. “A Generic Simulation Package for Organizing Outpatient Clinics.” *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, D. Goldsman, 1043-1047
- Lange, V. 1997 “Blueprints Designed to Improve Quality of Care Using simulation Tools.” Success Stories. <http://www.medmodel.com/stories.html>. (09 Sep. 1998).
- Lee, J. and M. Liu. 1988. “A Simulation of a Hospital Emergency Call System Using SLAM II.” *Simulation*. 51(6). 216-221.
- Lowder, B. “Durham Regional Saves \$150,000 Annually Using Simulation Tools.” Success Stories. 1997. <http://www.medmodel.com/stories.html>. (09 Sep. 1998).
- Lowery, J. 1999. “Design of Hospital Admissions Scheduling System Using Simulation.” *Proceedings of the 1996 Winter Simulation Conference*, ed. J. M.

- Charnes, D. J. Morrice, D. T. Brunner, and J.J. Swain, 1199-1204.
- Lursen, S., and J. Yang. 1996. "An Application of DSS in Truck Dispatching." *1996 Proceedings: Decision Sciences Institute*. 2. 556-558.
- McClean, S., and P. Millard. 1995. "A Decision Support System for Bed-Occupancy Management and Planning Hospitals." *IMA Journal of Mathematics Applied in Medicine and Biology*. 12. 249-257.
- 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, D. A. Sadowski, A. F. Seila, 861-867.
- Miller, L. 1997. "Pre-Op and Post-Op Planned Expansion Using Simulation." Success Stories. <http://www.medmodel.com/stories.html>. (09 Sep. 1998).
- Mittra, S. S. 1986. *Decision Support Systems: Tools and Techniques*, New York, NY: John Wiley and Sons, Inc.
- Moynihan, G., W. Nichols, P. Raj, and J. Sterling. 1995. "Decision Support System for Strategic Logistics Planning." *Computers in Industry*. 26. 75-84.
- O'Connell. E. M., L. A. Pedraza., J. M. Teich., and D. Thomas. 1996 "A Comprehensive Inpatient Discharge System." *Journal of the American medical Informatics Association*. Fall Symposium. 699-703
- Pegden, C. D., R. E. Shannon, and R. P. Sadowski. 1995. *Introduction to Simulation Using Siman*. Blacklick, OH: McGraw-Hill, Inc.

- Pitt, M. 1997. "A Generalised Simulation System to Support Strategic Resource Planning In Health Care." *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1155-1162.
- Reyes, F., 1998, *A Heuristic For On-Line Assessment Of Simulation Output For Goal Driven Simulation*, Master Thesis, Department of Industrial Engineering, Florida International University, Miami, FL 33199.
- Sauter, V.L. 1997. *Decision Support Systems: An Applied Managerial Approach*. New York, NY: John Wiley.
- Shannon, R.E. 1975. *Systems Simulation: the Art and Science*. Englewood Cliffs, New Jersey: Prentice Hall.
- Shuman, L., R. Dixon, and J. P. Young. 1975. *Operations Research in Health Care: A Critical Analysis*, Baltimore, MD: The Johns Hopkins University Press.
- Silver, M. 1991. *Systems that Support Decision-Makers: Description and Analysis*. West Sussex, England: John Wiley and Sons Ltd.

## APPENDIX A

1. The following questions are intended to gather information about the decisions taken by ER managers and the tools used to evaluate them

- From the list below select the most important decisions you need to make

- ☐Scheduling      Please specify:   ☐Nurses      ☐Physicians      ☐ER technician  
☐Prepare Budget(s)  
☐Equipment purchase  
☐Temporarily close ER  
☐Implementation of a new policy  
☐Modifications/Renovation of physical structure of facility  
☐Purchase medical supplies  
☐Other: Please name other decisions your make not included above

- Fill in the following information.

What tools do you use to aid you in making decisions in the following areas? (Name software packages or other techniques utilized)

➤ Scheduling	➤ Prepare Budget(s)	➤ Purchase equipment
➤ Purchase medical supplies	➤ Renovate facility	➤ Other:
➤ Handle insurance claims		

- Are you involved in deciding on renovations to facility (expanding ER, augmenting number of rooms)  
☐Yes   ☐No

If, Yes how often is your decision or recommendation on proposed changes actually implemented?

- ☐All the time   ☐Most of the time   ☐Once in a while   ☐They have never been implemented

If No, who is responsible?

- Please rank in order of importance (1 being most important) the measures of performance or statistics which are relevant to your ER

<input type="checkbox"/> Average waiting time	<input type="checkbox"/> Average time spend being treated
<input type="checkbox"/> Average time nurses are busy	<input type="checkbox"/> Average time doctors are busy
<input type="checkbox"/> Bed utilization	<input type="checkbox"/> Fast-track system utilization
<input type="checkbox"/> Number of patients leaving without being treated	<input type="checkbox"/> Number of patients refused treatment due to ER being closed

<input type="checkbox"/> Number of hours ER is closed per period of time	<input type="checkbox"/> ER Revenue
<input type="checkbox"/> Other	

2. The next questions are concerned with the computer systems used in your facility.

- Which operating system is used in your facility?  
☐ Win3.1    ☐ Win 95    ☐ Mac    ☐ Unix-based    ☐ Other:  
☐ Do not know
- Does your computer system(s) interface with the hospital's system?  
☐ Yes    ☐ No    ☐ Do not know
- Is patient data stored in a database? ☐ Yes    ☐ No    ☐ Do not know  
If yes, what database system do you use? ☐ Do not know
- Is nurse data stored in a database? (Schedules, salaries, hours worked in a week)  
☐ Yes    ☐ No    ☐ Do not know  
If yes, what database system do you use? ☐ Do not know
- Is cost data stored in a database (cost of medical supplies, equipment cost, etc.)  
☐ Yes    ☐ No    ☐ Do not know  
If yes, what database system do you use? ☐ Do not know
- From the list below check what you consider to be the most important capabilities or characteristics of an ideal decision support system to aid you in the decision-making process.  
☐ Retrieve relevant data from a database  
☐ Allow to make calculations with the data  
☐ Analyze effects on measures of performance by evaluating different scenarios (i.e. increase/decrease resources, of purchasing new equipment, change facility layout)  
☐ Analyze: ☐ Worst case scenario    ☐ On the average    ☐ Best case scenario  
☐ Display data: ☐ graphs    ☐ tables    ☐ other forms of displaying data  
☐ All off the above  
☐ Other: Please name what other capabilities you would like to see in a support system and which were not mentioned

3. The following questions seek to find information about how the decision of temporarily closing the ER is determined and evaluated.

- Do you ever have to temporarily close your ER? ☐ Yes    ☐ No  
If you answered yes to the previous question please continue with this section. Otherwise, skip to section 4
- How often do you have to evaluate if ER needs to be closed? (Average number of times per day)\_\_\_\_\_
- On the average, how many hours per month is your facility actually closed?  
☐ 0-10 hours    ☐ 10-20 hours    ☐ 20-30hours    ☐ 30-40 hours

- ☐ More than 40 hours
- For the following questions please check all options that apply  
When do you start debating that ER needs to be closed?  
☐ 0 beds unoccupied      ☐ 1-2 beds unoccupied  
☐ 3 or more beds unoccupied
  - At what point do you definitely close ER?  
☐ 0 beds unoccupied      ☐ 1-2 beds unoccupied  
☐ 3 or more beds unoccupied
  - What do you do to try to get patients through so that ER can be reopened?  
☐ Move less critical patients from beds to hallway units  
☐ Extend resources shifts (nurses)  
☐ Get less critical patients discharged  
☐ Get less critical patients admitted to hospital  
☐ Other: (please give examples of other actions taken to reduce time ER remains closed)

## APPENDIX B

### Model Frame

0\$	CREATE,	1,0.1:expo(1/Maxrate):MARK(Tin_Sys):NEXT(thinning);
thinning	BRANCH,	1: With,PeriodRate(period)/Maxrate,assignments,Yes: Else,33\$,Yes;
assignments	ASSIGN:	ptype=discrete(.08177,1,.11112,2,.51782,3,.96646,4,1,5): drttime=ED(ptype): nursetime=ED(ptype+8);
1\$	BRANCH,	1: If,ptype.eq.1.or.ptype.eq.2,60\$,Yes: Else,Totriage,Yes;
60\$	BRANCH,	1: If,NR(3).EQ.MR(3),critical_not serviced,Yes: Else,dummy1,Yes;
critical_not serviced	COUNT:	Critical_patients_not serviced,1;
22\$	DISPOSE;	
dummy1	QUEUE,	Dummyq1:MARK(Tin_dummy1);
23\$	SEIZE,	1: beds,1;
25\$	TALLY:	ptype+14,interval(Tin_Dummy1),1;
34\$	QUEUE,	DoctorQ;
35\$	SEIZE,	1: Doctors,1;
36\$	DELAY:	drttime;
37\$	RELEASE:	Doctors,1;
38\$	QUEUE,	NurseQ;
39\$	SEIZE,	1: nurses,1;
40\$	DELAY:	nursetime;
41\$	RELEASE:	nurses,1;
42\$	DELAY:	ED(ptype+13);
28\$	RELEASE:	beds,1:NEXT(LeaveER);
LeaveER	TALLY:	ptype,interval(tin_sys),1;
21\$	DISPOSE;	
Totriage	STATION,	Triage:MARK(Tin_Triage);
9\$	QUEUE,	TriageQ;
10\$	SEIZE,	1: Triage_Nurse,1;
12\$	DELAY:	uniform(5,10);
13\$	RELEASE:	Triage_Nurse,1;
11\$	TALLY:	ptype+6,interval(Tin_triage),1;
14\$	ROUTE:	0.0,Registration;
33\$	DISPOSE;	
2\$	STATION,	Registration:MARK(Tin_Reg);
3\$	QUEUE,	RegistrationQ;
4\$	SEIZE,	1: Registration_Clerk,1;
6\$	DELAY:	expo(10);
7\$	RELEASE:	Registration_Clerk,1;
5\$	TALLY:	ptype+3,interval(Tin_Reg),1;
8\$	ROUTE:	2.0,EDdepartment;



15\$	STATION,	EDdepartment:MARK(Tin_WR);
16\$	BRANCH,	1:
		If,NR(3).EQ.MR(3),askftbed,Yes:
		Else,EnterWR,Yes;
askftbed	BRANCH,	1:
		If,NR(4).EQ.MR(4),EnterWR,Yes:
		Else,dummy2,Yes;
EnterWR	STATION,	WaitingRoom;
getbed	QUEUE,	WR_Q,30,WROUT;
17\$	SEIZE,	1:
		beds,1;
18\$	TALLY:	ptype+9,interval(Tin_WR),1;
58\$	QUEUE,	DoctorQ;
59\$	SEIZE,	1:
		Doctors,1;
19\$	DELAY:	drttime;
51\$	RELEASE:	Doctors,1;
52\$	QUEUE,	NurseQ;
53\$	SEIZE,	1:
		nurses,1;
54\$	DELAY:	nursetime;
55\$	RELEASE:	nurses,1;
56\$	DELAY:	ED(ptype+13);
20\$	RELEASE:	beds,1:NEXT(LeaveER);
WRout	COUNT:	Patients_balking_WR,1;
57\$	DISPOSE;	
dummy2	QUEUE,	Dummyq2;
24\$	SEIZE,	1:
		ftbeds,1;
26\$	TALLY:	ptype+9,interval(Tin_WR),1;
43\$	QUEUE,	DoctorQ;
44\$	SEIZE,	1:
		Doctors,1;
27\$	DELAY:	drttime;
45\$	RELEASE:	Doctors,1;
46\$	QUEUE,	NurseQ;
47\$	SEIZE,	1:
		nurses,1;
48\$	DELAY:	nursetime;
49\$	RELEASE:	nurses,1;
50\$	DELAY:	ED(ptype+13);
29\$	RELEASE:	ftbeds,1:NEXT(LeaveER);
30\$	CREATE,	1,0.0:,1;
31\$	ASSIGN:	period=period+1;
32\$	BRANCH,	1:
		If,period.lt.4,lb1,Yes:
		Else,lb2,Yes;
lb1	DELAY:	ED(5+period):NEXT(31\$);
lb2	ASSIGN:	period=0:NEXT(31\$);

## Experimental Frame

ATTRIBUTES: 1,Ptype:  
2,Tin\_Reg:  
3,Tin\_Triage:  
4,Tin\_WR:  
5,Tin\_Sys:  
6,drtime:  
7,index:  
8,nursetime:  
9,drindex,1:  
10,nurseindex,1:  
11,Tin\_dummy1;

SCHEDULES: Reg\_Clerk,1\*420,2\*960,1\*60:  
Nurse,4\*180,3\*240,4\*240,6\*720,4\*60:  
DrShift,1\*630,2\*720,1\*90;

VARIABLES: 1,tba,:  
2,total\_beds\_busy,0:  
3,bedindex,1:  
4,period,:  
5,MaxRate,.056:  
6,Adriana,3:  
7,PeriodRate(3),.020,.056,.020;

QUEUES: 1,RegistrationQ,FirstInFirstOut:  
2,TriageQ,FirstInFirstOut:  
3,WR\_Q,LowValueFirst(Ptype):  
4,Dummyq1,FirstInFirstOut:  
5,Dummyq2,FirstInFirstOut:  
6,NurseQ,FirstInFirstOut(),SHARED:  
7,DoctorQ,FirstInFirstOut(),SHARED;

RESOURCES: 1,Registration\_Clerk,Schedule(Reg\_Clerk,Ignore),-,Stationary:  
2,Triage\_Nurse,Capacity(1),-,Stationary:  
3,beds,Capacity(14),-,Stationary:  
4,ftbeds,Capacity(3),-,Stationary:  
5,ERtechnician,Capacity(1),-,Stationary:  
6,nurses,Schedule(Nurse,Ignore),-,Stationary:  
7,Doctors,Schedule(DrShift,Ignore),-,Stationary;

STATIONS: 1,Registration:  
2,Triage:  
3,EDdepartment:  
4,WaitingRoom;

COUNTERS: 1,Critical\_patients\_not serviced,,Replicate:  
2,Patients\_balking\_WR,,Replicate;

TALLIES: 1,Time\_in\_systemP1:  
2,Time\_in\_systemP2:  
3,Time\_in\_systemP3:  
4,Time\_in\_systemP4:  
5,Time\_in\_systemP5:  
6,Time\_RegQ\_P3:  
7,Time\_RegQ\_P4:  
8,Time\_RegQ\_P5:  
9,Time\_TriageQ\_P3:  
10,Time\_TriageQ\_P4:  
11,Time\_TriageQ\_P5:  
12,Time\_WR\_P3:  
13,Time\_WR\_P4:  
14,Time\_WR\_P5:  
15,Time\_Dummyq1\_p1:  
16,Time\_Dummyq1\_p2;

DSTATS: 1,NR(1)/MR(1),Reg\_clerk\_util:

```

2,NR(2)/MR(2),Triage_Nurse_Util:
3,NQ(1),Avg_no_regQ:
4,NQ(2),Avg_no_triageQ:
5,NQ(3),Avg_no_WR:
6,NQ(4),No_dummy1:
7,NQ(5),No_dummy2:
8,NR(3),No_Beds_Busy:
9,NR(4),No_FT_Busy:
10,NR(6)*100/MR(6),Nurse_Utilization:
11,NR(7)*100/MR(7),Doctor_Utilization:
12,MR(3)-NR(3),Available_beds:
13,NQ(3)+NR(3)+NR(4),ERTotal:
14,NR(3)*100/MR(3),Bed_Utilization;

OUTPUTS: 1,DAVG(8),"c:\students\adriana\outputs\bedbusy.dat",Beds_Busy:
2,DAVG(12),"c:\students\adriana\outputs\unccbeds.dat",Unoccupied_Beds:
3,DAVG(13),"c:\students\adriana\outputs\ertot.dat",Total_in_ER:
4,DAVG(9),"c:\students\adriana\outputs\ftbusy.dat",Busy_FT:
5,DAVG(14),"c:\students\adriana\outputs\bedut.dat",Bed_Utilization:
6,DAVG(5),"c:\students\adriana\outputs\wrtot.dat",NoWR:
7,DAVG(10),"c:\students\adriana\outputs\nurseut.dat",Nurse_Utilz:
8,DAVG(11),"c:\students\adriana\outputs\drutil.dat",Dr_Utilz;

REPLICATE, 13,0.0,1261,Yes,Yes,0.0;

EXPRESSIONS: 1,drtime_p1,uniform(10.2,19.8):
2,drtime_p2,uniform(10.2,19.8):
3,drtime_p3,uniform(10,15):
4,drtime_p4,uniform(1,5):
5,drtime_p5,uniform(1,5):
6,Espera1,480:
7,Espera2,840:
8,Espera3,120:
9,nursetime_p1,uniform(30,60):
10,nursetime_p2,uniform(30,60):
11,nursetime_p3,uniform(30,40):
12,nursetime_p4,uniform(20,40):
13,nursetime_p5,uniform(0,30):
14,maxstay_p1,max(0,erla(10.2,4)*60-nursetime-drtime):
15,maxstay_p2,max(0,max(norm(230.4,123),107.4)-nursetime-drtime):
16,maxstay_p3,max(0,gamm(84,90)-nursetime-drtime):
17,maxstay_p4,max(0,max(norm(33.12,17.7),15.42)-nursetime-drtime):
18,maxstay_p5,max(0,max(norm(33.12,17.7),15.42)-nursetime-drtime);

```

## APPENDIX C

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
<b>Decision Areas:</b>										
<b>Human Resources</b>										
Staff Scheduling	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hiring new personnel	✓	✓	✓	✓	✓		✓			
Evaluation of personnel	✓				✓			✓		
Control labor over time				✓						
<b>Facility Operation</b>										
Temporarily close ER	✓	✓	✓	✓	✓	✓		✓	✓	
Resolve patients complaints		✓	✓	✓	✓	✓	✓	✓	✓	✓
Adoption of new policies			✓		✓	✓	✓	✓	✓	✓
Analysis on ER renovation										
<b>Cost and Inventory</b>										
Inventory Control of medical supplies	✓			✓						
Prepare budget	✓				✓	✓	✓	✓		✓
Equipment Purchase	✓	✓			✓		✓	✓		
Purchase Office supplies/medical supplies				✓	✓	✓	✓	✓		
Charging supplies to patients				✓						
Medication distribution process				✓						
<b>Other:</b>										
Identify and address Problems area: bottlenecks, procedures	✓		✓							

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
<b>Most Relevant Measures of Performance</b>										
Patient length of stay										✓
Patient waiting times										
Service times	✓	✓	✓	✓	✓	✓	✓			
No. pat. not serviced		✓	✓	✓	✓		✓	✓	✓	✓
No. pat. Discharged		✓				✓	✓		✓	
No. pat refused treatment due to ER being closed	✓				✓	✓	✓	✓		✓
Number of patients/day					✓	✓		✓	✓	
Time to administer Thrombolytics	✓			✓						
Revenue	✓	✓			✓	✓		✓		✓
Customer satisfaction		✓								
Resource utilization					✓				✓	
Bed/Fast-track utilization					✓		✓	✓	✓	✓
Ambulance arrivals/hour					✓					

	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
<b>Decision:</b>										
<b>Temporarily Close ER</b>										
Does ER temporarily close?	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Time ER is closed (per month)	N/A	14	40	N/A	10-20	0-10	N/A	0-10	0-10	10-20
<b>Criteria for closing ER</b>							N/A	N/A		
All regular beds are full	✓	✓	✓	✓	✓	✓		✓	✓	✓
High volume of high acuity patients	✓	✓		✓	✓			✓	✓	
No resources to take care of additional patients			✓		✓	✓		✓		
Patient safety is at risk				✓	✓					
Total patients in ER exceeds a fixed number								✓		
Long patient waiting times									✓	
<b>Actions taken to reopen ER</b>			N/A				N/A	N/A		
Discharge less critical patients	✓	✓		✓	✓	✓			✓	✓
Reorganize patients and accommodate less critical ones in auxiliary units						✓			✓	✓
Extend nurses shifts					✓					
Extend doctors shifts					✓					
Pull other clinical and ancillary staff to assist in expediting patient flow and patient care						✓				✓
MD and RN on call									✓	✓
Encourage direct admission										✓
Improve turnaround times on dirty beds										

## APPENDIX D

### VBA Code

#### FORMS

##### AskOutput Form

```
Private Sub CheckBox1_Click()  
    Call Checkboxes(AskOutputNbr, AskOutput, 1)  
End Sub  
  
Private Sub CheckBox2_Click()  
    Call Checkboxes(AskOutputNbr, AskOutput, 2)  
End Sub  
  
Private Sub CheckBox3_Click()  
    Call Checkboxes(AskOutputNbr, AskOutput, 3)  
End Sub  
  
Private Sub CheckBox4_Click()  
    Call Checkboxes(AskOutputNbr, AskOutput, 4)  
End Sub  
  
Private Sub CheckBox5_Click()  
    Call Checkboxes(AskOutputNbr, AskOutput, 5)  
End Sub  
  
Private Sub CommandButton5_Click()  
    AskOutput.Hide  
End Sub  
  
Private Sub PreviousBtn_Click()  
    AskOutput.PreviousBtn.Cancel = True  
    AskOutput.Hide  
End Sub  
  
Private Sub UserForm_Initialize()  
    Call LoadingForms(AskOutputNbr, AskOutput)  
End Sub
```

##### CloseERMain Form

```
Private Sub ContinueBtn_Click()  
    flag = ContinueButtons(CloseERmainNbr, CloseERmain)  
    If flag <> 1 Then CloseERmain.Hide  
End Sub  
Private Sub OptionButton1_Click()  
    Call OptionButtons(CloseERmainNbr, CloseERmain, -1, 1)  
End Sub  
Private Sub OptionButton2_Click()  
    Call OptionButtons(CloseERmainNbr, CloseERmain, 0, 2)  
End Sub  
  
Private Sub PreviousBtn_Click()  
    CloseERmain.PreviousBtn.Cancel = True  
    CloseERmain.Hide  
End Sub
```

```

Private Sub TextBox2_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(CloseERmainNbr, CloseERmain, KeyAscii, 2)
End Sub

Private Sub TextBox3_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(CloseERmainNbr, CloseERmain, KeyAscii, 3)
End Sub

Private Sub TextBox4_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(CloseERmainNbr, CloseERmain, KeyAscii, 4)
End Sub

Private Sub TextBox5_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(CloseERmainNbr, CloseERmain, KeyAscii, 5)
End Sub

Private Sub UserForm_Initialize()
    Call LoadingForms(CloseERmainNbr, CloseERmain)
End Sub

```

## Configuration Form

```

Private Sub doctors_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(ConfigurationNbr, Configuration, KeyAscii, 2)
End Sub

Private Sub fasttrackbeds_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(ConfigurationNbr, Configuration, KeyAscii, 7)
End Sub

Private Sub nurses_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(ConfigurationNbr, Configuration, KeyAscii, 3)
End Sub

Private Sub OptionButton1_Click()
    Configuration.Label8.Enabled = True
    Configuration.fasttrackbeds.Enabled = True
End Sub

Private Sub OptionButton2_Click()
    Configuration.Label8.Enabled = False
    Configuration.fasttrackbeds.Text = 0
    Configuration.fasttrackbeds.Enabled = False
End Sub

Private Sub Registration_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(ConfigurationNbr, Configuration, KeyAscii, 5)
End Sub

Private Sub Triage_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    Call KeyPressCode(ConfigurationNbr, Configuration, KeyAscii, 6)
End Sub

Private Sub UserForm_Initialize()
    Call Beginning
End Sub

Private Sub UserForm_Terminate()
    End
End Sub

```

## Information Form

```

Private Sub Button1_Click()
    flag = ContinueButtons(InformationNbr, information)
    If flag <> 1 Then information.Hide

```



```

End Sub

Private Sub Button2_Click()
    information.Hide
End Sub

Private Sub PreviousBtn_Click()
    information.PreviousBtn.Cancel = True
    information.Hide
End Sub

Private Sub UserForm_Initialize()
    Call LoadingForms(InformationNbr, information)
End Sub

```

## Inventory Form

```

Private Sub CommandButton2_Click()
    Call Previous(InventoryNbr, inventory)
End Sub
Private Sub ContinueBtn_Click()
    Call NextForm(inventory)
End Sub

```

## MainMenu Form

```

Private Sub CommandButton1_Click()
    MainMenu.CommandButton1.Cancel = True
    MainMenu.Hide
End Sub

Private Sub UserForm_Initialize()
    Call LoadingForms(MainMenuNbr, MainMenu)
End Sub

Private Sub ContinueBtn_Click()
    flag = ContinueButtons(MainMenuNbr, MainMenu)
    If flag <> 1 Then MainMenu.Hide
End Sub
Private Sub OptionButton1_Click()
    Call OptionButtons(MainMenuNbr, MainMenu, 0, 1)
End Sub
Private Sub OptionButton2_Click()
    Call OptionButtons(MainMenuNbr, MainMenu, 0, 2)
End Sub

Private Sub OptionButton3_Click()
    Call OptionButtons(MainMenuNbr, MainMenu, 0, 3)
End Sub

```

## Modification Form

```

Private Sub CommandButton2_Click()
    Call Previous(ModificationNbr, modification)
End Sub

Private Sub ContinueBtn_Click()
    Call NextForm(modification)
End Sub

```

## Output1 Form

```
Private Sub CommandButton1_Click()  
    End  
End Sub  
  
Private Sub ContinueBtn_Click()  
    Unload output1  
End Sub  
  
Private Sub UserForm_Initialize()  
    Call ShowFinal(Output1Nbr, output1)  
End Sub
```

## Output2 Form

```
Private Sub CommandButton1_Click()  
    Unload Output2  
    modelo.Application.Quit  
    modelo.End  
End Sub  
  
Private Sub UserForm_Initialize()  
    Call CI  
End Sub  
  
Private Sub UserForm_Terminate()  
    End  
End Sub
```

## StaffManagement Form

```
Private Sub ContinueBtn_Click()  
    Call NextForm(StaffManagement)  
End Sub  
  
Private Sub PreviousBtn_Click()  
    StaffManagement.PreviousBtn.Cancel = True  
    StaffManagement.Hide  
End Sub
```

## Welcome Form

```
Private Sub CommandButton1_Click()  
    Call ToConfiguration  
End Sub  
  
Private Sub ContinueBtn_Click()  
    Welcomeform.Hide  
End Sub  
  
Private Sub Image1_Click()  
    Welcomeform.Hide  
End Sub  
  
Private Sub Label1_Click()  
    Welcomeform.Hide  
End Sub
```

```
Private Sub Label2_Click()  
    Welcomeform.Hide  
End Sub  
  
Private Sub Label3_Click()  
    Welcomeform.Hide  
End Sub  
  
Private Sub UserForm_Click()  
    Welcomeform.Hide  
End Sub
```

## MODULES

### Global Variables Module

```
Public Const ConfigurationNbr = 1
Public Const WelcomeformNbr = 2
Public Const MainMenuNbr = 3
Public Const CloseERmainNbr = 4
Public Const InformationNbr = 5
Public Const AskOutputNbr = 6
Public Const InventoryNbr = 7
Public Const ModificationNbr = 8
Public Const StaffManagementNbr = 9
Public Const Output1Nbr = 10
Public Const InformationtwoNbr = 11
Public Const Output2Nbr = 12
Public Const output3Nbr = 13
Public ftvar As Integer
Public flag As Integer
Public corrida As Integer
Public repli1 As Integer
Public simtime As Double
Public forecastfinal As Double
Public Timetoclose As String
Public MyInit1 As Integer
Public myinit2 As Integer
Public modelo As Arena.Model, out As Arena.SIMAN
Public resourcesmodule, schedmodule, passignmodule, bedsbranchmodule, ftbedsbranchmodule As
Arena.Module
Public bedsbranch2module, replicatemodule, createmodule, createthinmodule, wrqueuemodule As
Arena.Module
Public variablemodule As Arena.Module
```

### Codigo Module

#### 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
        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
```

### **Sub Exit\_Appl()**

```
Dim repl As Integer
    MsgBox "closed form"
    repl = modelo.Modules.Find(smFindTag, "replicatetag")
    If repl > 0 Then
        Set replicatemodule = modelo.Modules.Item(repl)
    Else
        MsgBox "Did not find replicate module"
    End If
    replicatemodule.data("Length") = 0.1
    replicatemodule.data("NumReps") = 1
    replicatemodule.UpdateShapes

End Sub
```

### **Sub NextForm(formname As UserForm)**

```
    '-----
    'Message for decisions not implemented in this DSS
    '-----
    MsgBox "This option is not currently available"
End Sub
Sub ToConfiguration()
    '-----
    'Show Configuration menu
    '-----
    Welcomeform.Hide
    Configuration.Show
End Sub
```

### Sub CloseERlogica()

```
Dim Dataone, Forecastone, texto2, texto3, texto4, texto5 As String
Dim Datatime3, Forecasttime2, CurrentTime As Double
Dim flag1 As Integer
Dim Mytime, MyDate, MyStr
Dim restag, schedtag, repl, inter, resft, thin, AdrianaVar, Minuto, Hora As Integer
Dim texto5Min, texto4Min As Integer
Dim interdist As String
'-----
'This logic will change the interarrival time of patients based on whether user
'selected using historical or today's data. (use interval opened in create block)
'-----
inter = modelo.Modules.Find(smFindTag, "createtag")
thin = modelo.Modules.Find(smFindTag, "createthintag")
If inter > 0 Then
    Set createmodule = modelo.Modules.Item(inter)
Else
    MsgBox "Did not find create module"
End If
If thin > 0 Then
    Set createthinmodule = modelo.Modules.Item(thin)
Else
    MsgBox "Did not find create thin module"
End If

'-----
'Converting current time into numeric value for calculations by
'transforming standard p.m times into military time.
'-----
Mytime = Time
MyStr = Format$(Mytime, "hmm AMPM")
Hora = Hour(Mytime)
Minuto = Minute(Mytime)
Stop
If MyStr > TimeValue("01:00 PM") Then
    CurrentTime = Val(MyStr) + 1200
ElseIf TimeValue("12:00 AM") < MyStr < TimeValue("12:59 PM") Then
    CurrentTime = Val(MyStr)
Else
    CurrentTime = Val(MyStr)
End If

'-----
'Converting time when user is considering closing ER into proper
'form for calculation of simulation length
'-----
texto2 = Format$(CloseERmain.TextBox2.Text, "00")
texto4 = Format$(CloseERmain.TextBox4.Text, "00")
texto4Min = Val(texto4)
Forecastone = texto2 + texto4

If CloseERmain.OptionButton4.Value = True Then
    Forecasttime2 = Val(texto2) + 12
    ' Timetoclose = texto2 + ":" + texto4 + " " + "P.M"
ElseIf CloseERmain.OptionButton3.Value = True Then
    Forecasttime2 = Val(texto2)
    ' Timetoclose = texto2 + ":" + texto4 + " " + "A.M"
End If

'-----
'Converting time of data collection into proper form for calculations
'-----
texto3 = Format$(CloseERmain.TextBox3.Text, "00")
texto5 = Format$(CloseERmain.TextBox5.Text, "00")
texto5Min = Val(texto5)
'Dataone = texto3 + texto5
If CloseERmain.OptionButton1.Value = True And CloseERmain.OptionButton6.Value = True Then
    Datatime3 = Val(texto3) + 12
```



```

ElseIf CloseERmain.OptionButton1.Value = True And CloseERmain.OptionButton5.Value = True Then
    Datatime3 = Val(texto3)
ElseIf CloseERmain.OptionButton2.Value = True Then
    Datatime3 = Val("00")
End If

'simtime = Int(((CurrentTime - Datatime) / 100) * 60 + ((Forecasttime - CurrentTime) / 100) * 60
+ 120)
simtime = Int((60 * (Hora - Datatime3) + (Minuto - texto5Min)) + (60 * (Forecasttime2 - Hora) +
(texto4Min - Minuto)) + 120)
Stop
'-----
'Change value of simulation length in Replicate Module
'Set n (#of replications)= 10 during preliminary experimental run
'-----

repl = modelo.Modules.Find(smFindTag, "replicatetag")
If repl > 0 Then
    Set replicatemodule = modelo.Modules.Item(repl)
Else
    MsgBox "Did not find replicate module"
End If

replicatemodule.data("Length") = simtime
replicatemodule.data("NumReps") = MyInit1
replicatemodule.UpdateShapes
'-----
' Change val of var adriana in siman model, so that ADRIANA always has the
' value of the initial number of replications
'-----
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

variablemodule.data("value(1,6)") = MyInit1
variablemodule.UpdateShapes
'-----
'If user selects to use HISTORICAL data then use historical distribution and
'do thinning process
'-----
If CloseERmain.OptionButton2.Value = True Then
    createmodule.data("nextlabel") = "thinning"
    createmodule.data("interval") = "expo(1/Maxrate)"
    createthinmodule.data("offset") = "0.0"
    createmodule.UpdateShapes
    createthinmodule.UpdateShapes
End If

'-----
'If user selects to use TODAY'S data then do not do thinning process and just use
'one distribution (the value for interarrival times comes from data obtained from
'best fit in "Information" form)
'-----
If CloseERmain.OptionButton1.Value = True Then
    createmodule.data("nextlabel") = "assignments"
    createthinmodule.data("offset") = "15000"
    createmodule.UpdateShapes
    createthinmodule.UpdateShapes
End If

End Sub

```

## Function ContinueButtons(formnumber As Integer, formname As UserForm) As Integer

```

'-----
'Declaring all variables that will be used in the ContinueButtons Subroutines
'-----
Dim restag, schedtag, repl, inter, resft, thin As Integer
Dim interdist As String
'-----
'This logic will change the interarrival time of patients based on wether user
'selected using historical or today's data. (use interval opeand in create block)
'-----
inter = modelo.Modules.Find(smFindTag, "createtag")
thin = modelo.Modules.Find(smFindTag, "createthintag")
If inter > 0 Then
    Set createmodule = modelo.Modules.Item(inter)
Else
    MsgBox "Did not find create module"
End If
If thin > 0 Then
    Set createthinmodule = modelo.Modules.Item(thin)
Else
    MsgBox "Did not find create thin module"
End If

ContinueButtons = -1
Select Case formnumber
    Case ConfigurationNbr
        '-----
        'Making sure that all information in Configuration form has been entered by user
        '-----
        If formname.OptionButton1.Value = False And formname.OptionButton2.Value = False Then
            MsgBox "Please specify if your ER has a fast-track system"
        ElseIf formname.OptionButton1.Value = True And formname.fasttrackbeds.Text = "" Then
            MsgBox "Please indicate number of fast-track beds"
        Else
            '-----
            'Change resource capacities according to user's input
            '-----
            restag = modelo.Modules.Find(smFindTag, "resourcestag")'look for block tag named

setstag
            Dim wr As Integer
            wr = modelo.Modules.Find(smFindTag, "WRtag")
            If wr > 0 Then
                Set wrqueuemodule = modelo.Modules.Item(wr)
            Else
                MsgBox "Did not find wr queue module"
            End If

            If restag > 0 Then
                Set resourcesmodule = modelo.Modules.Item(restag)
            Else
                MsgBox "Did not find resources module" 'if module is not found display

message to user
            End If

            repl = modelo.Modules.Find(smFindTag, "replicatetag")
            If repl > 0 Then
                Set replicatemodule = modelo.Modules.Item(repl)
            Else
                MsgBox "Did not find replicate module"
            End If
            resourcesmodule.data("CapExp(3)") = Configuration.beds.Text 'change
            resourcesmodule.data("CapExp(4)") = Configuration.fasttrackbeds.Text 'change
            resourcesmodule.data("CapExp(2)") = Configuration.Triage.Text 'change
            wrqueuemodule.data("Capacity") = Configuration.wrcapacity.Text
            If formname.OptionButton2.Value = True Then

```

```

        resourcesmodule.data("CapExp(4)") = 0
        End If
        resourcesmodule.UpdateShapes
        replicatemodule.UpdateShapes
        Configuration.Hide
        Load MainMenu
        MainMenu.Show
    End If
Case MainMenuNbr
    -----
    'Making sure that user made a selection before continuing
    -----
    If formname.OptionButton1.Value = False And formname.OptionButton2.Value = False And
formname.OptionButton3.Value = False Then
        MsgBox "You need to specify the type of decision to be evaluated in order to
continue"
        ContinueButtons = 1
    End If
Case CloseERmainNbr
    -----
    'Code to be executed when all data is entered into the CloseERMain form
    'Making sure that user filled in all information required for evaluation
    -----
    If formname.TextBox2.Text = "" Or formname.TextBox4.Text = "" Then
        MsgBox "Please indicate the time of day for which you want to make evaluation"
        ContinueButtons = 1
    ElseIf Val(formname.TextBox2.Text) < 0 Or Val(formname.TextBox2.Text) > 12 Then
        MsgBox "The hour must be a number between 0 and 12"
        ContinueButtons = 1
    ElseIf Val(formname.TextBox4.Text) < 0 Or Val(formname.TextBox4.Text) > 59 Then
        MsgBox "The minutes must be a number between 0 and 60"
        ContinueButtons = 1
    ElseIf formname.OptionButton3.Value = False And formname.OptionButton4.Value = False
Then
        MsgBox "You forgot to indicate if A.M or P.M"
        ContinueButtons = 1
    ElseIf formname.OptionButton1.Value = False And formname.OptionButton2 = False Then
        MsgBox "You need to specify if you want to use today's data"
        ContinueButtons = 1
    ElseIf formname.OptionButton1.Value = True And (formname.TextBox3.Text = "" Or
formname.TextBox5.Text = "") Then
        MsgBox "Please indicate the time of the day from when you want to use data"
        ContinueButtons = 1
    ElseIf Val(formname.TextBox3.Text) < 0 Or Val(formname.TextBox3.Text) > 12 Then
        MsgBox "The hour must be a number between 0 and 12"
        ContinueButtons = 1
    ElseIf Val(formname.TextBox5.Text) < 0 Or Val(formname.TextBox5.Text) > 59 Then
        MsgBox "The minutes must be a number between 0 and 60"
        ContinueButtons = 1
    ElseIf formname.OptionButton1.Value = True And formname.OptionButton5.Value = False And
formname.OptionButton6.Value = False Then
        MsgBox "You forgot to indicate if A.M or P.M"
        ContinueButtons = 1
    End If
Case InformationNbr
    -----
    'Check if user made appropriate selections
    -----
    If information.Fileslist.Value = "" Then
        MsgBox "Please select file to be analyzed"
        ContinueButtons = 1
    Else
        Dim MyAppl, Pathfile
        Pathfile = "P:\HOME\Centeno\adriana\dss\" & information.Fileslist.Value

        -----
        'Show message if no text files can be found in specified path
        -----

```

```
        If Dir$("p:\home\centeno\adriana\dss\*.txt") = "" Then
            MsgBox "No.txt files could be found"
            ContinueButtons = 1
        End If
    End If
End Select
End Function
```

## Sub OpenInput()

-----  
'Open Input Analyzer so that input data can be analyzed and distribution for  
'interarrival times can be determined  
-----

'Run Input Analyzer

```
Dim MyAppl, Pathfile
Pathfile = "P:\HOME\Centeno\adriana\dss\" & 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 file
SendKeys Pathfile          'Copy name of file to be opened
SendKeys "%{O}", True      'Open the file to be analyzed
SendKeys "%{T 2}{ENTER}", True 'Show Fit menu
SendKeys "A", True         'Fit all distributions
SendKeys "%{E}", True      'Show edit menu
SendKeys "E", True         'Copy distribution expression
SendKeys "%{F4}", True     'Close input 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

### Sub LoadingForms(formnumber As Integer, formname As UserForm)

```
Dim Mytime, MyStr

Select Case formnumber
    Case MainMenuNbr
        MainMenu.CommandButton1.Cancel = False
        '-----
        'Add the different decision on each area upon loading form
        '-----
        formname.ComboBox1.AddItem "Staff Information"
        formname.ComboBox1.AddItem "Staff Scheduling"
        formname.ComboBox2.AddItem "Close ER"
        formname.ComboBox2.AddItem "Facility Expansion/Modification"
        formname.ComboBox2.AddItem "New Policy Evaluation"
        formname.ComboBox3.AddItem "Budget"
        formname.ComboBox3.AddItem "Equipment Purchase"
        formname.ComboBox3.AddItem "Medical Supplies" 'Inventory, purchase, ordering
    Case InformationNbr
        '-----
        'Display different files that could be opened to determine their distribution
        '-----
        Dim Filespath, FileName
        Filespath = "P:\HOME\Centeno\adriana\dss\*.txt" 'setting the path
        FileName = Dir(Filespath) ' Retrieve the first entry.
        Do While FileName <> "" ' Start the loop.
            FileName = Dir ' Get next entry.
            information.Fileslist.AddItem FileName 'Adding names of files to combobox
        Loop
    Case CloseERmainNbr
        '-----
        'Display current time
        '-----
        Mytime = Time
        MyStr = Format(Mytime, "h:mm AMPM")
        CloseERmain.Label2.Caption = MyStr
        formname.PreviousBtn.Cancel = False
    Case Else
        formname.PreviousBtn.Cancel = False
End Select
End Sub
```

### Sub Checkboxes(formnumber As Integer, formname As UserForm, Numero As Integer)

```
Select Case formnumber
    Case AskOutputNbr
        Select Case Numero
            Case 1
                formname.TextBox1.Visible = True
                If formname.CheckBox1.Value = False Then formname.TextBox1.Visible = False
            Case 2
                formname.TextBox2.Visible = True
                If formname.CheckBox2.Value = False Then formname.TextBox2.Visible = False
        End Select
    End Select
End Sub
```

**Sub OptionButtons(formnumber As Integer, formname As UserForm, Valor As Integer, Numero As Integer)**

```
Select Case formnumber
    Case CloseERmainNbr
        '-----
        'List the different types of decisions under each category
        '-----
        formname.Label5.Visible = Valor
        formname.Label8.Visible = Valor
        formname.Label9.Visible = Valor
        formname.TextBox3.Visible = Valor
        formname.TextBox5.Visible = Valor
        formname.OptionButton5.Visible = Valor
        formname.OptionButton6.Visible = Valor
    Case MainMenuNbr
        Select Case Numero
            Case 1
                formname.ComboBox1.Visible = True
                formname.ComboBox2.Visible = False
                formname.ComboBox3.Visible = False
            Case 2
                formname.ComboBox2.Visible = True
                formname.ComboBox1.Visible = False
                formname.ComboBox3.Visible = False
            Case 3
                formname.ComboBox3.Visible = True
                formname.ComboBox1.Visible = False
                formname.ComboBox2.Visible = False
        End Select
    End Select
End Select
End Sub
```

# **Sub KeyPressCode(formnumber As Integer, formname As UserForm, Tecla As MSForms.ReturnInteger, Numero As Integer)**

-----  
 'Verifying that user input a numeric value in the textboxes where  
 'required in all forms  
 -----

```

Select Case formnumber
    Case CloseERmainNbr
        Select Case Numero
            Case 2
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.TextBox2.Text = ""
                End If
            Case 3
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.TextBox3.Text = ""
                End If
            Case 4
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.TextBox4.Text = ""
                End If
            Case 5
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.TextBox5.Text = ""
                End If
        End Select
    Case ConfigurationNbr
        Select Case Numero
            Case 1
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.beds.Text = ""
                End If
            Case 2
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.doctors.Text = ""
                End If
            Case 3
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.nurses.Text = ""
                End If
            Case 5
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.Registration.Text = ""
                End If
            Case 6
                If Tecla < 48 Or Tecla > 57 Then
                    MsgBox "Please enter a numeric value"
                    Tecla = 0
                    formname.Triage.Text = ""
                End If
            Case 7
                If Tecla < 48 Or Tecla > 57 Then

```



```
                MsgBox "Please enter a numeric value"
                Tecla = 0
                formname.fasttrackbeds.Text = ""
            End If
        End Select
    End Select
End Sub
```

## Sub Beginning()

'This code shows the current values for number of beds, doctors, nurses, f-t beds etc that the model  
'has as defaults according to time of the day and that could be changed by user when necessary

```
Dim restag, schedtag, pattag, wr As Integer
restag = modelo.Modules.Find(smFindTag, "resourcestag") 'look for block with tag resourcestag
schedtag = modelo.Modules.Find(smFindTag, "schedtag")
pattag = modelo.Modules.Find(smFindTag, "patientstag")
wr = modelo.Modules.Find(smFindTag, "WRtag")
If wr > 0 Then
    Set wrqueuemodule = modelo.Modules.Item(wr)
Else
    MsgBox "Did not find wr queue module"
End If
If restag > 0 Then
    Set resourcesmodule = modelo.Modules.Item(restag)
Else
    MsgBox "Did not find resources module"
End If
If schedtag > 0 Then
    Set schedmodule = modelo.Modules.Item(schedtag)
Else
    MsgBox "Did not find schedules module"
End If
If pattag > 0 Then
    Set passignmodule = modelo.Modules.Item(pattag)
Else
    MsgBox "Did not find assign module"
End If
'-----
'Depending on time of day system will know the number of drs,reg clerks and nurses
'that should be working
'-----
If Time > TimeValue("0:00") And Time < TimeValue("10:29") Then
    Configuration.doctors = Val(schedmodule.data("Capacity(1,3)"))
ElseIf Time > TimeValue("10:30") And Time < TimeValue("22:29") Then
    Configuration.doctors = Val(schedmodule.data("Capacity(2,3)"))
ElseIf Time > TimeValue("22:30") And Time < TimeValue("23:59") Then
    Configuration.doctors = Val(schedmodule.data("Capacity(3,3)"))
End If
If Time > TimeValue("0:00") And Time < TimeValue("06:59") Then
    Configuration.Registration = Val(schedmodule.data("Capacity(1,1)"))
ElseIf Time > TimeValue("7:00") And Time < TimeValue("22:59") Then
    Configuration.Registration = Val(schedmodule.data("Capacity(2,1)"))
ElseIf Time > TimeValue("23:00") And Time < TimeValue("23:59") Then
    Configuration.Registration = Val(schedmodule.data("Capacity(3,1)"))
End If
If Time > TimeValue("0:00") And Time < TimeValue("02:59") Then
    Configuration.nurses = Val(schedmodule.data("Capacity(1,2)"))
ElseIf Time > TimeValue("3:00") And Time < TimeValue("06:59") Then
    Configuration.nurses = Val(schedmodule.data("Capacity(2,2)"))
ElseIf Time > TimeValue("07:00") And Time < TimeValue("10:59") Then
    Configuration.nurses = Val(schedmodule.data("Capacity(3,2)"))
ElseIf Time > TimeValue("11:00") And Time < TimeValue("22:59") Then
    Configuration.nurses = Val(schedmodule.data("Capacity(4,2)"))
ElseIf Time > TimeValue("23:00") And Time < TimeValue("23:59") Then
    Configuration.nurses = Val(schedmodule.data("Capacity(5,2)"))
End If

With Configuration
    .beds = resourcesmodule.data("CapExp(3)")
    .fasttrackbeds = resourcesmodule.data("CapExp(4)")
    .Triage = resourcesmodule.data("CapExp(2)")
    .wrcapacity = wrqueuemodule.data("Capacity")
End With
End Sub
```

### Sub ShowFinal(formnumber As Integer, formname As UserForm)

```
'-----
'Display form with output results from summary report
'at the end of the simulation
'-----
    Dim Temporal, Temporal1, Temporal2, newtemp, newtemp1, newtemp2 As Double
    Dim otro, otro1
    Dim restag, ft As Integer
    restag = modelo.Modules.Find(smFindTag, "resourcestag")
    If restag > 0 Then
        Set resourcesmodule = modelo.Modules.Item(restag)
    Else
        MsgBox "Did not find resources module"
    End If
    ft = resourcesmodule.data("CapExp(4)")

    Select Case formnumber

Case Output1Nbr
        Temporal = modelo.SIMAN.OutputAverageAcrossReplications(1)
        Temporal1 = modelo.SIMAN.OutputAverageAcrossReplications(2)
        Temporal2 = modelo.SIMAN.OutputAverageAcrossReplications(3)
        otro = simtime

        With output1
            .busybeds.Caption = Format$(out.OutputAverageAcrossReplications(1), "##0.00")
            .Utilldr.Caption = Format$(out.OutputAverageAcrossReplications(8), "##0.00")
            .Utilnurse.Caption = Format$(out.OutputAverageAcrossReplications(7), "##0.00")
            .wrlevel.Caption = Format$(out.OutputAverageAcrossReplications(6), "##0.00")
            .Label36.Caption = Format$(out.OutputAverageAcrossReplications(2), "##0.00")
            .Label38.Caption = Format$(out.OutputAverageAcrossReplications(3), "##0.00")
            .utilbed.Caption = Format$(out.OutputAverageAcrossReplications(5), "##0.00")
            .busyft.Caption = Format$(out.OutputAverageAcrossReplications(4), "##0.00")
            .displayt.Caption = Format$(Timetoclose, "#0:00")
        End With

        If ft > 0 Then
            output1.Label26.Enabled = True
            output1.busyft.Enabled = True
            output1.busyft.Caption = Format$(out.DStatAverage(9), "##,##0.00")
        End If
    End Select
End Sub
```

### Sub Replications1()

```
Set modelo = ThisDocument.Model
Set out = modelo.SIMAN
Dim h, hstar As Double
Dim repl, i, nstar As Integer

'-----
'Formula to find the required number of replications
'to obtain the optimal value of h(hstar)
'-----
h = modelo.SIMAN.OutputHalfWidthAcrossReplications(3)
hstar = 2

repl = modelo.Modules.Find(smFindTag, "replicatetag")
If repl > 0 Then
    Set replicatemodule = modelo.Modules.Item(repl)
Else
    MsgBox "Did not find replicate module"
End If
Stop
MsgBox hstar
MsgBox Format(h, "###0.0##")
```

```

If h <= hstar Then                                '10 replications are enough
    MsgBox "h less than hstar"
    Load output1
    output1.Show
    Load Output2
    Output2.Show
ElseIf h > hstar Then
    nstar = Int(10 * ((h / hstar) ^ 2)) 'formula to calculate nstar
    If nstar > 100 Then nstar = 100
    replicatemodule.data("NumReps") = nstar
    myinit2 = nstar
    MyInit1 = 550
    repli1 = 0
    replicatemodule.UpdateShapes
    modelo.StartOver
    modelo.Go
    modelo.Application.refresh

```

```

    End If

```

```

End Sub

```

### **Sub Replications2()**

```

    Load output1
    output1.Show
    Load Output2
    Output2.Show
    modelo.End
End Sub

```

### Sub CI()

```
'-----  
'routine to construct confidence interval.  
'-----  
  
Dim halfw1, halfw2, halfw3, xavg1, xavg2, xavg3, lowerlimit1, lowerlimit2, lowerlimit3 As Double  
Dim upperlimit1, upperlimit2, upperlimit3 As Double  
xavg1 = modelo.SIMAN.OutputAverageAcrossReplications(1)  
xavg2 = modelo.SIMAN.OutputAverageAcrossReplications(2)  
xavg3 = modelo.SIMAN.OutputAverageAcrossReplications(3)  
halfw1 = modelo.SIMAN.OutputHalfWidthAcrossReplications(1)  
halfw2 = modelo.SIMAN.OutputHalfWidthAcrossReplications(2)  
halfw3 = modelo.SIMAN.OutputHalfWidthAcrossReplications(3)  
lowerlimit1 = xavg1 - halfw1  
upperlimit1 = xavg1 + halfw1  
lowerlimit2 = xavg2 - halfw2  
upperlimit2 = xavg2 + halfw2  
lowerlimit3 = xavg3 - halfw3  
upperlimit3 = xavg3 + halfw3  
lowerlimit1 = Format(lowerlimit1, "###.##")  
upperlimit1 = Format(upperlimit1, "###.##")  
lowerlimit2 = Format(lowerlimit2, "###.##")  
upperlimit2 = Format(upperlimit2, "###.##")  
lowerlimit3 = Format(lowerlimit3, "###.##")  
upperlimit3 = Format(upperlimit3, "###.##")  
With Output2  
    .Label5.Caption = "(" + lowerlimit2 + "," + Str$(upperlimit2) + ")"  
    .Label2.Caption = "(" + Str$(lowerlimit3) + "," + Str$(upperlimit3) + ")"  
End With  
End Sub
```

## This Document

### Sub ModelLogic\_RunBegin()

```
    repli1 = 0
    MyInit1 = 10
    Call Abrir
End Sub
```

### 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)"))
    '-----
    ' Setting MyInit2
    '-----
    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
```