MENU-DRIVEN MANUFACTURING SYSTEM SIMULATOR

BY

IMED H. JAMOUSSI

Bachelor of Science in

Industrial Engineering

and Management

Oklahoma State University

Stillwater, Oklahoma

1985

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 1987

MENU-DRIVEN MANUFACTURING

SYSTEM SIMULATOR

Thesis Approved:

Joe H. Mize
Thesis Adviser

Carl B. Estes

John W. Nazemetz

Norman N. Durham
Dean of the Graduate College

PREFACE

A new software has been created in this project: a Menu-Driven Manufacturing System Simulator (MDMSS). The software is designed so that even if the user does not know much about simulation, he(she) is still able to simulate systems. The users are anticipated to be related to manufacturing as well as the systems to be simulated. The software has been tested. Some of the examples are included in this thesis. This first version of MDMSS is found to satisfy all the objectives of this project.

As I conclude this research work, I look back to the past just to see many great people and organizations without whom I would have never been able to accomplish this achievement. Therefore, I believe that the least I can do is to give these people the credit they deserve.

I would like to express my sincere appreciation and gratitude to the following people and organizations that made this study possible with their support, influence, and assistance:

- Dr. Joe H. Mize, my major adviser, for his intelligent guidance, concern, and invaluable help.
- Dr. Carl B. Estes, and Dr. John W. Nazemetz for accepting to be my committee members.

- The School Of Industrial Engineering and Management at Oklahoma State University for the knowledge that helped me carry out this research.
- AT&T foundation for sponsoring the CIMS research center at Oklahoma State University. Without this massive research going on at this center, probably the subject of my work would have been different.
- Mr. Pablo Nuno, a PHD candidate, for his advice and continuous encouragement.
- Selma for a great job in editing this work.
- The Tunisian Government for sponsoring me financially all along my Studies in the United States.
- The Scientific Mission of Tunisia as well as the Tunisian Embassy in Washington, DC. for their concern and valuable services.
- The Tau Beta Pi Association for including me in their exciting experience.
- The Alpha Pi Mu Association for inviting me and accepting me as one of their members.
- The University Computer Center staff for their help in solving some of the problems I faced when working on this project.
- All my friends here for sharing with me both the good and bad times. Without the moral support of some of them, it would have been a lot harder to survive some of the difficulties I faced throughout my studies.
- A Great father, Hassouna, a Super mother, Naziha, for

their unending support and belief in my capabilities.

- A caring Abdelhamid, who had a lot of influence on me since my childhood.
- My two brothers, Adnen and Aref, and my sister, Aida, for their encouragements and services.
- Finally, my beloved Hajeur for her positive influence, encouragement, and understanding.

Last but not least, I would like to dedicate my modest work to the memory of all the loving persons I left home and I never had the chance to see again.

TABLE OF CONTENTS

Chapter Page

Chapter Page

Chapter Page

Chapter | Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

For a long time, agriculture has been the dominant activity of Mankind. It was sufficient to provide Man with the most important need: Survival. As Man went to explore the surroundings, the need to convert raw materials to finished goods arose. Probably the first Man- made products were the weapons made out of stone to protect him from wild animals. A new phenomenon was born then: Production. Production is the process of transforming raw materials to finished goods. To produce goods, a combination of manual labor, special tools, and some kind of energy is essential. Later on, through the centuries, Man needed more and more goods, and the products made became more and more complex. Consequently, production tools and processes had to evolve to satisfy the market. Agriculture was losing its standing for the benefit of manufacturing. Nowadays, developing countries, as well as the developed ones, see manufacturing as the predominant tool to assure a good standard of living. Once, when life was in its primitive stages, agriculture was sufficient to assure the survival of Man; nowadays, survival of nations is contingent to its ability to produce goods. This is the manufacturing Era.

Manufacturing started with Man using only his hands and some basic tools to convert raw material. Step by step, the manual skills of Man were used less and less, more tools and new machines had to be introduced to help in the production process. Competition between industries and also between nations led manufacturing to its latest level: Automation.

Production of a certain item is a parameter of the supply and demand. Figure 1 shows the overall manufacturing cycle. This cycle includes a sequence of events that are necessary to follow in order to permit goods to be available to people. The main events are pictured in Figure 2. This figure shows that the manufacturing system is a closed loop process whose elements need to be controlled and engineered first as identical units and then as a total system.

Manufacturing, Computers, and Simulation

With the fast evolution in manufacturing processes, needs, and objectives, the integration of the discovery of the century in today's industrial environment emerged smoothly and rapidly. The computer had to be exploited to the maximum. It was found to be the savior of the modern manufacturing. In fact, it has been used in a variety of tasks. From data storage to fully automated process monitor, the computer showed an extraordinary flexibility in solving many engineering problems and concerns. Computer integrated manufacturing systems (CIMS), although still conceptual, seem to be the next stage in the manufacturing evolution.

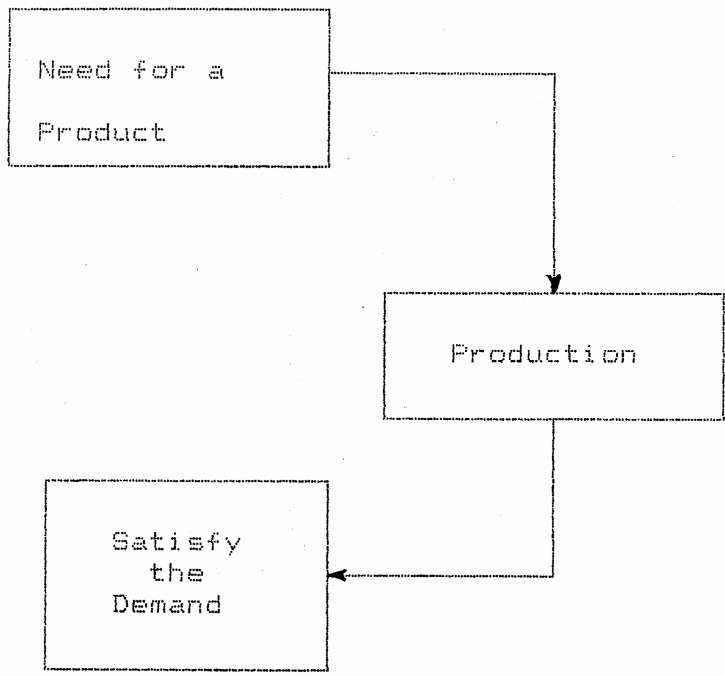


Figure 1. Overall Manufacturing
Cycle

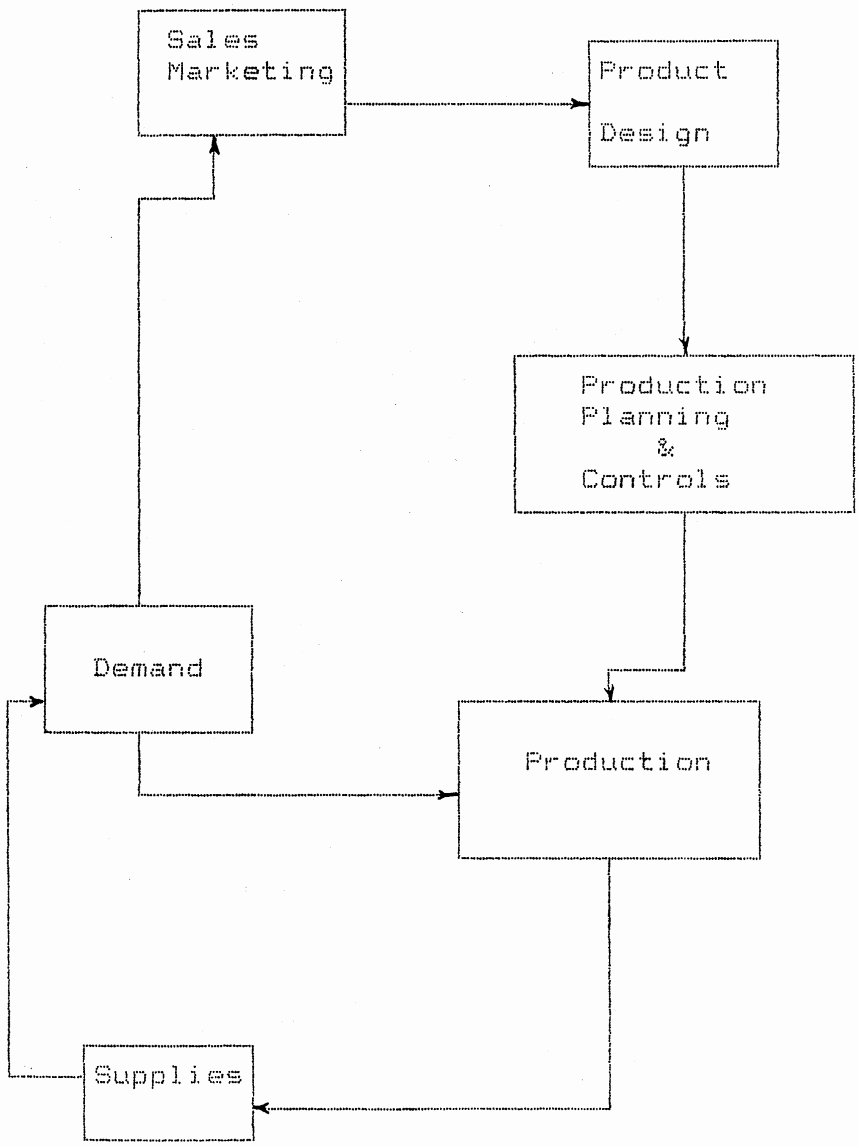


Figure 2. Sequence of Manufacturing Cycles

Ideally, once such a system has been designed, implemented, and put to work, a distributed processing computer has to control the different elements of the manufacturing cycle. More important, the software in CIMS should be able to simulate the actual manufacturing system when analysis or decision making is needed.

Pritsker defined computer simulation as the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer[31]. Another definition of simulation by Mize and Cox (as referenced in Turner, Mize, and Case[38]) states that, simulation is the process of conducting experiments on a model of a system in lieu of, either (1) direct experimentation with the system itself or, (2) direct analytical solution of some problem associated with the system. Both definitions agree that simulation allows drawing conclusions about the system, without building it, disturbing it, or destroying it. Thus, a simulation model is very useful in both design and analysis.

Simulation models are built primarily as aids for decision making. For this reason, measures of performance are to be determined for the performance assessment.

A simulation of a manufacturing system, or even one part of it, could be a very challenging and complex task. In fact, the external factors that could influence such a system, and consequently any contingent decision, are enormous. What makes the simulation even harder is the fact that some of these factors cannot be formulated as explicit mathematical equations or variables mainly because these factors might be uncertain and unpredicted. A fire, an earthquake, and a strike are just a few examples. Figure 3 is an illustration of a manufacturing system and its standing in the environment. Theoretically, such a system can never be simulated exactly; for this reason any model describing the system is just an approximation of the real system.

Model building is both an art and a science. It is a very critical step in the simulation process. Any mistake or misinterpretation in designing the model could jeopardize the total simulation. The model building approach could be compared to a human eye that is able to reproduce a scene, but can never recognize and include all the details of the scene (although most people think that they do !). Figure 4 shows the model building approach and the mentioned analogy at the same time. The point is that simulation is just another tool that helps decision making, but is by no mean exact.

Problem Statement

What makes a comprehensive simulation of manufacturing systems almost impossible is the wide variety of such systems. To prove this, just consider the measures of performance that could be used to evaluate an armament industry compared to the ones for a food industry! More,

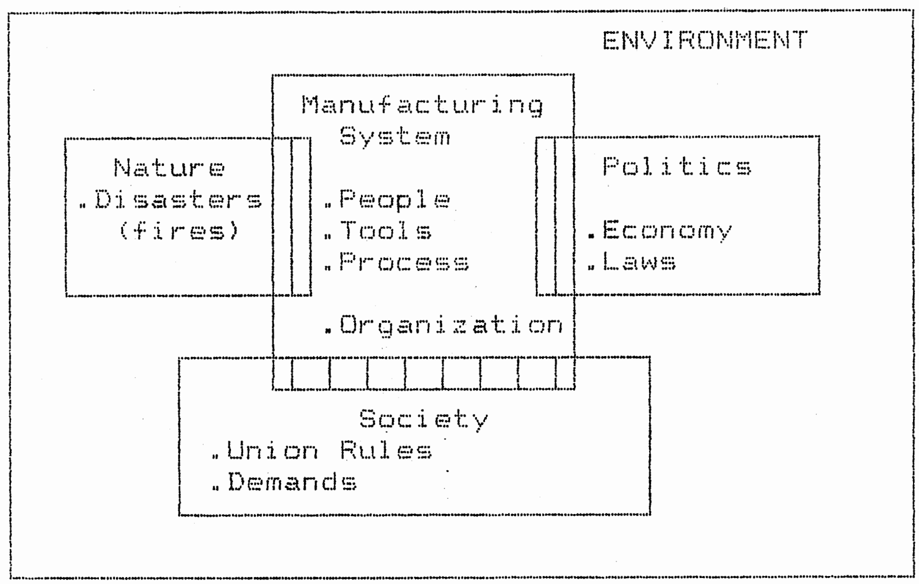


Figure 3. A Manufacturing System

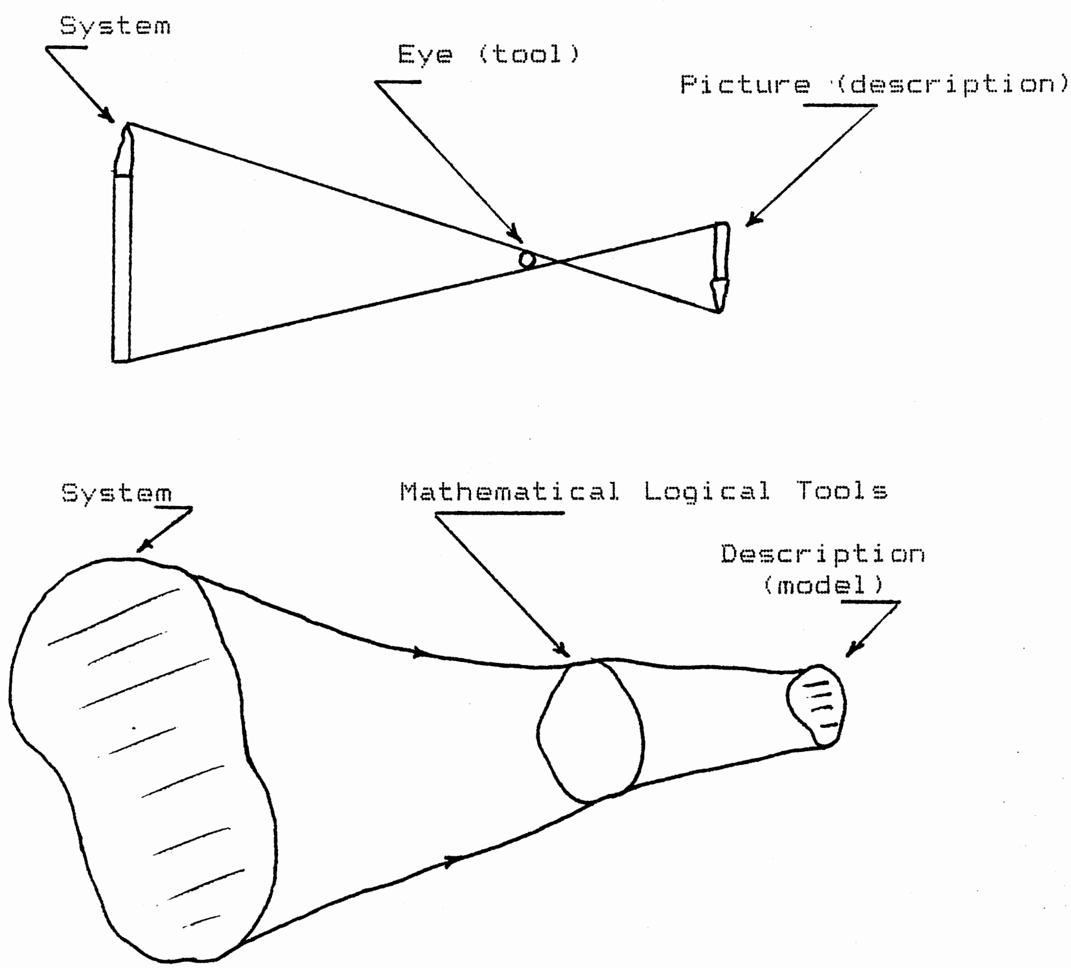


Figure 4. Model Building Approach Analogy

even simulating just one module of a single manufacturing system could be very challenging; still too many parameters exist. In fact, trying to simulate the production module in a manufacturing system, by itself, could cause many problems. The cause is that production plants can vary according to (1) the production quantities, (2) the type of plant layout, and (3) the production operations. These three classifications are described below:

A.1 Production Quantities

1.1 Job Shop Production

It is categorized by a low volume, flexible equipment, and a high skill level among the workers.

1.2 Batch Production

Lots are manufactured at predetermined interval time. The equipment is for general purpose.

1.3 Mass Production

The same product is processed at high production rates, the equipment is dedicated, and the skill of the worker is low.

A.2 Types of Plant Layout

1.1 Fixed-Position

The process remains fixed for a certain period, and the work stations are brought to it. Plane building is an example. Figure 5 shows a fixed position layout.

1.2 Process Layout

The main feature of such a process is that the machines are arranged according to their purpose. Figure 6 illustrates this process.

1.3 Product Flow Layout

It is categorized by production of either, one product, or one class of products. The work stations are arranged to satisfy the route the product needs to undertake. Figure 7 is an illustration of this process.

A.3 Production Operations

1.1 Processing Operations

It includes activities that change the state of a part. No material or components are added during this transformation (energy is needed, an example is forging).

1.2 Assembly Operations

Two or more components are added to make a finished product.

Due to the increasing competition, industries often face the urge to change, modify, or even replace some of their processes of productions. One of the manufacturing engineering functions is process planning and evaluation.

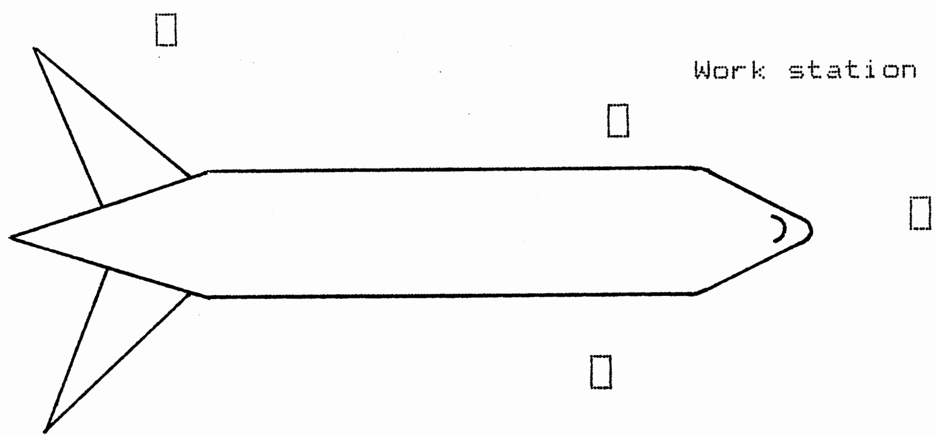


Figure 5. Fixed Position Layout

Assembly	Finishing	Die Casting	Recieving
Milling	Painting	Grinding	Shipping

Figure 6. Process Layout

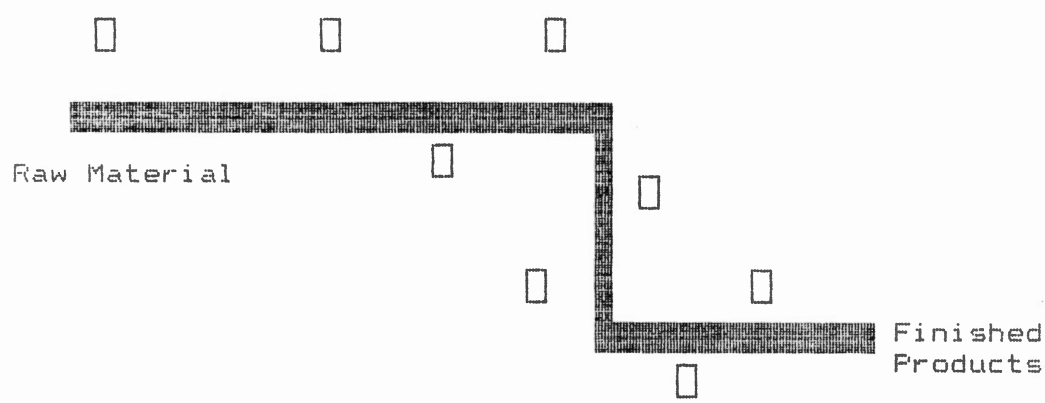


Figure 7. Product Flow Layout

Process planning is directly related to the production floor layout and functions. For planning, the engineer could use the real system (the production floor) to implement and test his design. This might be very costly; in fact, just moving one machine (a numerical control machine for example) could cost thousands of dollars! This cannot be justified economically especially when the outcome of the change is not known. Also, such redesign could interfere with the natural process of the system resulting in decreasing output, and again loss of money. For these reasons, a general purpose simulation model of the actual system could be the ideal tool to the engineer in order to design, implement, or evaluate new ideas. As it was mentioned before, such a simulation is just a tool for educated approximations. Although it might be the best tool available now, it still faces some uncertainties that need to be taken into consideration before the final decision is pronounced.

CHAPTER II

OBJECTIVES

Lately, manufacturing companies are facing a challenging phenomenon: Competition. Competition is found at both the national and international levels. Success of any company is very dependent on its products' quality, and its costs! Engineers have to work harder and smarter to fulfill these requirements. Faced with this burden, engineers turned to computers to save time and money. Today's computers with their high processing speeds, low cost memory, had to be exploited to the maximum in all areas where they are applicable, especially in the decision making process. As a result, the computer deeply affected the engineers' practice. This new design tool (the computer) became a necessity in some areas such as literature search, data manipulation, mathematical operations, optimization, and simulation.

This project is an attempt to provide engineers with a new tool that would help them in making decisions concerning manufacturing floor controls. As a sequence, the project consists of developing a simulation model that should be able to fit different configurations on the shop floor. A detailed explanation of the model will follow in other chapters. To use this tool, the engineer needs to be familiar only with the process to be simulated. The model is designed so that only very basic computer skills are required from the user; the operator does not need to know anything about simulation modeling or simulation languages. As a matter of fact, the user could be totally ignorant on how to develop software and still should be able to use the model.

This program is not an iterative optimizer though. After running the program, the operator obtains some statistics that evaluate some predetermined measures of performance. From here, it is the job of the engineer to analyze the given data. This is done by organizing it (the data) to information that could be used later in making the final decision.

So, the primary objective of this project is to develop new software that engineers could use when it comes to making decisions concerning manufacturing production floor implementations. To use this tool, no expertise in either simulation or any other programming language is needed. The model is used to predict a set of measures of performance. The program has to accommodate future implementations as they become needed. Also, the program is to be user friendly, menu driven, and well documented internally and externally. Finally, when designing this software, thoughts about the potential use of this program in a future computer integrated manufacturing system (CIMS) had to be taken into consideration.

To summarize, the main goal of this project is to create a new, efficient, simulation tool that could help engineers, especially manufacturing engineers, to obtain useful predictions of performance for certain situations when decisions have to be made. This has to be possible whether or not the operator has any expertise about computer programming.

Situations Where the Program Could be Used

In order to define, or rather make the use of this software easier, some anticipated situations where the program could be used are discussed below:

A.1 Situation 1

Consider the shop floor showed in Figure 8, also suppose that the engineer wants to know if adding a new machine is justifiable or not (see Figure 9). In this case, a simple simulation should provide the necessary parameters for the decision.

A.2 Situation 2

Now consider the process shown in Figure 10. Initially, only one product was made. The question is the possibility of the integration of another product using the same machines (see Figure 11). Once again, the program should be able to help make such a decision.

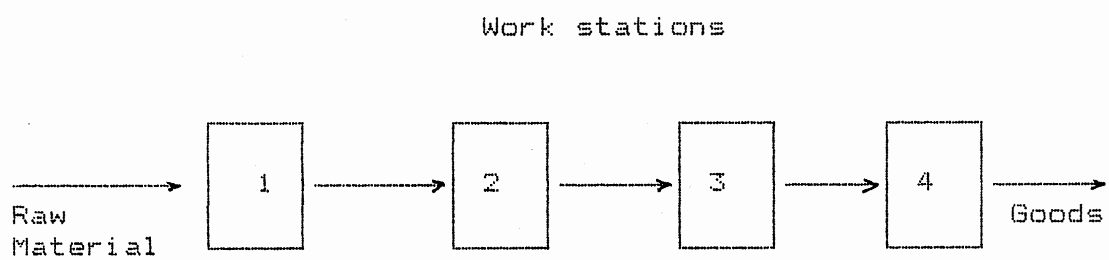


Figure 8. Present System

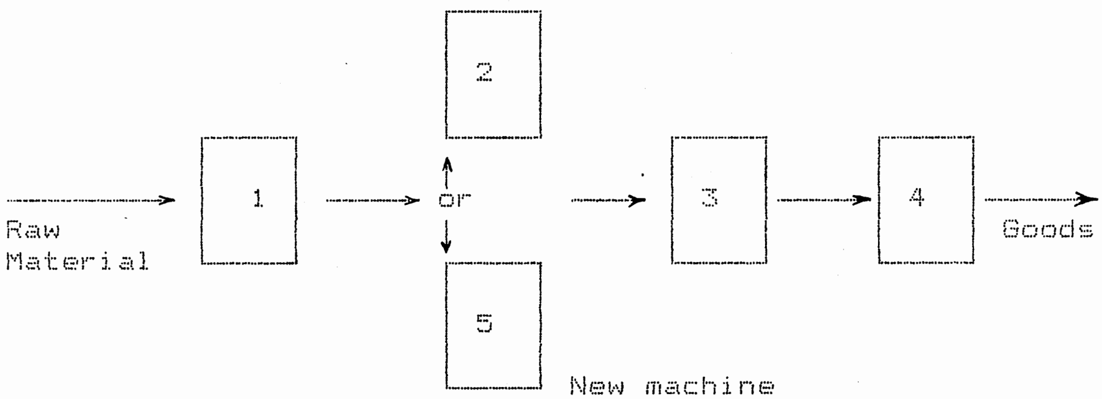


Figure 9. New System

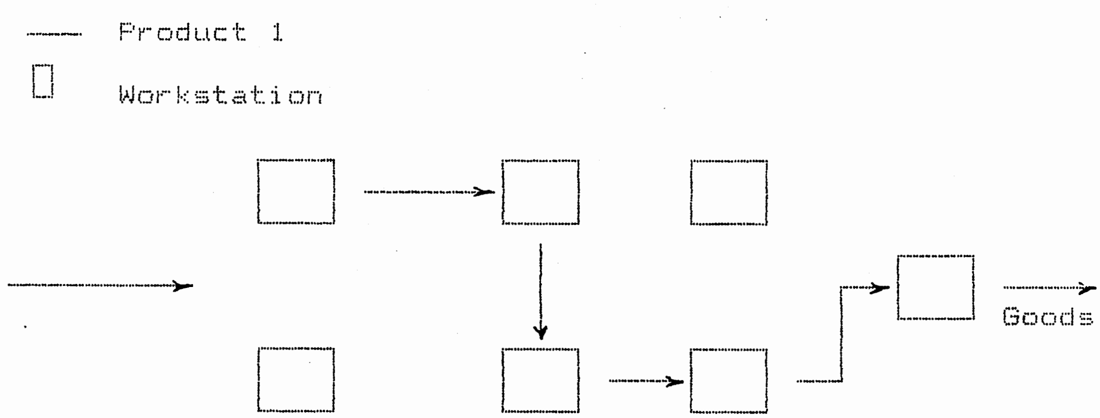


Figure 10. Actual Process

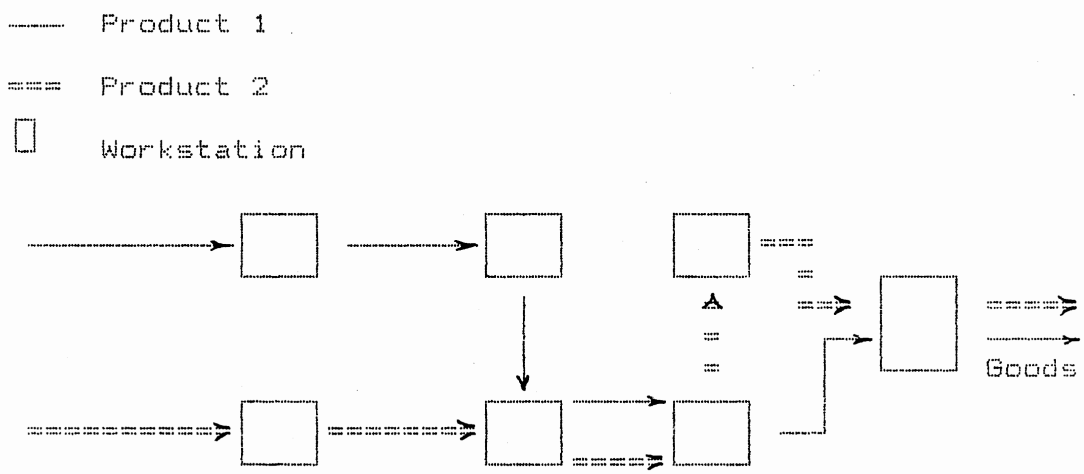


Figure 11. Proposed Process

The previous situations are just examples, and they are, by no means, the only applications of this model. A more detailed discussion of the software and its applications will follow in separate chapters.

CHAPTER III

LITERATURE REVIEW

Simulation is being used in diverse fields and different applications. In fact, it has been used in military, medical research, government projects, and biology, just to name a few areas. To illustrate the variety in simulation applications, just consider the two following examples: in the first one, SLAM II (a simulation language) was used to obtain the values of six parameters that affect the state of a lake. Continuous modeling was used to monitor these variables as time elapses. The second application is a military one. It is a massive simulation of both the NATO and Soviet sides in a conventional war in Europe. This simulation required the development and use of a 40,000 lines of SIMSCRIPT 11.5 program named TAC THUNDER. In both applications, simulation was found to be the ultimate tool and technique to help in the decision making process.

Surveys indicate that simulation and statistics are widely used in various areas. Industry is one of the areas where simulation became a necessity. In manufacturing, for example, simulation is considered to be the most powerful tool in dealing with the operations of flow lines, whether they are automated or not.

Before computers reached their present state of development, simulation models were limited to mostly developing heuristic algorithms. The breakthrough in computer technology lead to a new Era of simulation: the Computer Simulation. The simulation software was found to be fast, economically justifiable, and able to accommodate most problems.

The manufacturing areas that were considered in simulation are different. The models developed differ in goals. Some studies were concerned with flow lines [17], [18], [28], [29]; others were developed to solve productivity and production problems [12], [17]; some others were used in sequencing and scheduling [1]. Few of the most recent studies were a result of new concepts in manufacturing. Simulation of flexible manufacturing systems is an example [11]; the design of automated factories is another one [35].

Probably one of the newest concepts in the manufacturing area is CIMS: Computer Integrated Manufacturing Systems. To put this concept to work, simulation had to be used extensively. As its name states, CIMS integrates all different parts of a manufacturing system. This integration is not possible without the use of a massive computer. One of the many modules that need to be included in CIMS is the simulation model. Simulation is to be performed at different levels. This Thesis project offers to implement a new software. This software can be best described as a Menu-

Driven Manufacturing System Simulator. At a later stage this simulator is hoped to be one part of the Simulation model in a Computer Integrated Manufacturing System.

CHAPTER IV

METHODS OF ANALYSIS

As was mentioned in earlier chapters, the main objective of this project is to provide engineers, more specifically manufacturing engineers, a tool that they could rely upon when it is time to make decisions about shop floor layout, processes, or products routing. The model to be designed, built, and documented has to be as comprehensive as possible. A trade off emerges at this point: if a general purpose model is to be implemented, many details have to be excluded from it.

Also, in the previous sections of this report, it was mentioned that the technique to be used in creating this model is simulation. Two questions could be asked:

1- Why simulation?

2- Is there any other technique that could be considered?

Answering the second question first, yes, other techniques are available. In fact, since the 1960's many analytical methods were developed. But, with the evolution of the manufacturing industry and its continuous complexity, heuristic algorithms became harder to create, and developing such models became very complex and even impossible in some situations. Such difficulties obliged researchers to turn to the computer just to discover that simulation is the ultimate technique to use when one needs to experiment with real-life shop control. Consequently, iterative simulation ended up to be the most popular way to evaluate different processes when decision making is on the line.

The answer to the second question proposed earlier, falls straight forward: simulation is to be used in this project because it (simulation) is probably the most powerful tool that is able to help in the analysis of a given situation. A second reason for using simulation in this study, is to provide a tool that does not require the user to have major expertise in building models. When the model is designed properly, by just answering a set of questions, the user leads the program to picture and simulate a real life situation. A third factor that favors simulation over the other techniques is the flexibility of simulation in adjusting from one situation to a similar one. This advantage rarely exists when using theoretical modeling. In fact, for example, when a mathematical model is developed it serves a very precise situation; a single change in the input usually requires the total recreation of the model. On the other hand, when simulation is used and the model is well designed, switching from one situation to another should not cause any major problems; if the model was not designed to adjust automatically, only minor changes in the code of the program should be sufficient to accommodate the new situation.

The software to be created for this project should serve different situations without requiring any changes in the code. Of course, totally comprehensive software may never be available. This study is just another attempt to implement a simulation program that is not comprehensive, but rather, a general purpose model that could be used for different situations in manufacturing processes decisions.

Assuming that using simulation was agreed to be a good decision, one could ask why start from scratch and not take one of the many existing software packages and just modify it? The answer to this question might not be convincing at first, but when looked at closely should be satisfactory: if the reader is familiar with programming in general, or simulation in particular, he(she) should agree that a model is judged to be good, only if it reflects the real world as closely as possible. When one takes an already written program and tries to modify it to accommodate new situations, most of the time the results become somewhat misleading. The reason for that could be explained rather easily: when a program is changed extensively, its structure and logical flow become very poor. Consequently, debugging the software for verification is very difficult. For those reasons, in this project, before the design of the model took place, the general scope of the study had to be determined. Which means that most of (if not all) the situations that the software has to accommodate were anticipated. Then and only then, the general purpose model was designed.

Up to now, only good things were mentioned about simulation. The simulation technique has also some disadvantages. Although they are few, they could be deterministic in not using this powerful tool. The cost of the implementation of the simulation model is the first disadvantage. In fact, for some situations, the model costs a lot of money. Also, because of the nature of the results given by simulation, validation and verification are vital. This could be a big challenge, essentially when the model is huge. Last but not least, simulations provide the user with only statistics. The obvious next step is to process the given data into relevant information that is easy to use when the decision time comes. Manipulating such statistics could require advanced expertise in this field.

To summarize, the urge to develop a general purpose simulation model for manufacturing systems, and the need to provide a tool that, almost, does not require any expertise in programming inspired this study.

CHAPTER V

LANGUAGE TO BE USED

As of October 1986, the catalog of simulation software shows fifty three different simulation languages or packages that are available commercially. Appendix A is a list of the available software. The reason for the coexistence of all these languages results from the widespread use of simulation as both an analysis and a decision making tool. Some of these languages are for general purpose simulations, others are designed for very specific applications. Banks and Carson [3] state that simulation languages are constructed generally from three frameworks: process-interaction, event-scheduling, and continuous process.

The process-interaction perspective uses statements that define the flow of the entities within or through a network. In continuous simulation, some dependent variables in the model change continuously over the simulated time. In the event-scheduling framework, systems are modeled by defining the events that are able to change the state of the system and the logic associated with each kind of event.

To decide upon a particular simulation language to use to model a situation, many factors have to be considered. First, the selection of the programming language is a function of whether or not it is well championed by the designer. The capabilities of the language to fit the situation to be modeled are also a determining factor. A third factor concerns the measures of performance sought from the simulation: a given language could be able to simulate a situation, but unable to provide the operator with the needed statistics necessary for the decision making or analysis process. Last, some firms are sometimes concerned with the availability of a language that could be used on a time sharing network.

For this project, after a preliminary educated elimination, four languages were found to be prospects to be used. They are: SIMLIB, GPSS, SLAM II, and SIMNET.

SIMNET is a new language developed by Dr. Hamdy A. Taha which has not yet been commercialized. Due to the possibility of the existence of some unsolved problems in the software, and especially because this language does not provide any form of interactive mode up to this date, this software was excluded.

GPSS (General Purpose Simulation System) is a process-oriented simulation language for modeling discrete systems. It was first developed in 1960. Because this language is limited in computing power and lacks a capability for floating point or real arithmetic, it was also excluded from this study.

SIMLIB is a package formed by a set of FORTRAN subroutines that files entities, processes the event calendar, and calculates the time dependent statistics based on observations. For the magnitude of this project, SIMLIB seemed rather a low level simulation software compared to the others.

SLAM (Simulation Language for Alternative Modeling) is a FORTRAN based simulation language. It was first introduced in 1979. SLAM II, the latest version of SLAM, allows the use of the three different modeling viewpoints in just one integrated framework. In fact, process-interaction, event-scheduling, and continuous modeling could be used separately or in any combination which makes the language very powerful. Also, the language allows the integration of independent FORTRAN 77 subroutines to the model. This makes the interactive option of the program possible and feasible.

Although the language to be used in a certain simulation is just a tool, the choice of the most appropriate one could make a considerable difference in debugging, providing a better, more efficient model, and producing more satisfactory results. In this project, SLAM II was found to be the most appropriate language.

CHAPTER VI

MODEL DESCRIPTION

System Description

It was emphasized in the previous chapters that the primarly purpose of this project is to create and implement a tool that manufacturing engineers primarily, and other engineers in general, could use to evaluate some given or proposed manufacturing situations and processes. It is then only logical to describe or rather define the system or systems that could be subject to simulation using the proposed model.

Manufacturing environments differ from one organization to another and even from one application to another. For this reason, to fit a general purpose, a simulation model requires the study to originate from the most general configuration available in a manufacturing situation.

In general, a manufacturing or production floor in the manufacturing cycle is formed of the following components:

1- A storage area.

2- A fabrication area.

3- An assembly area.

Figure 12 is an illustration of a general production floor. The arrows show the traditional flow of materials and

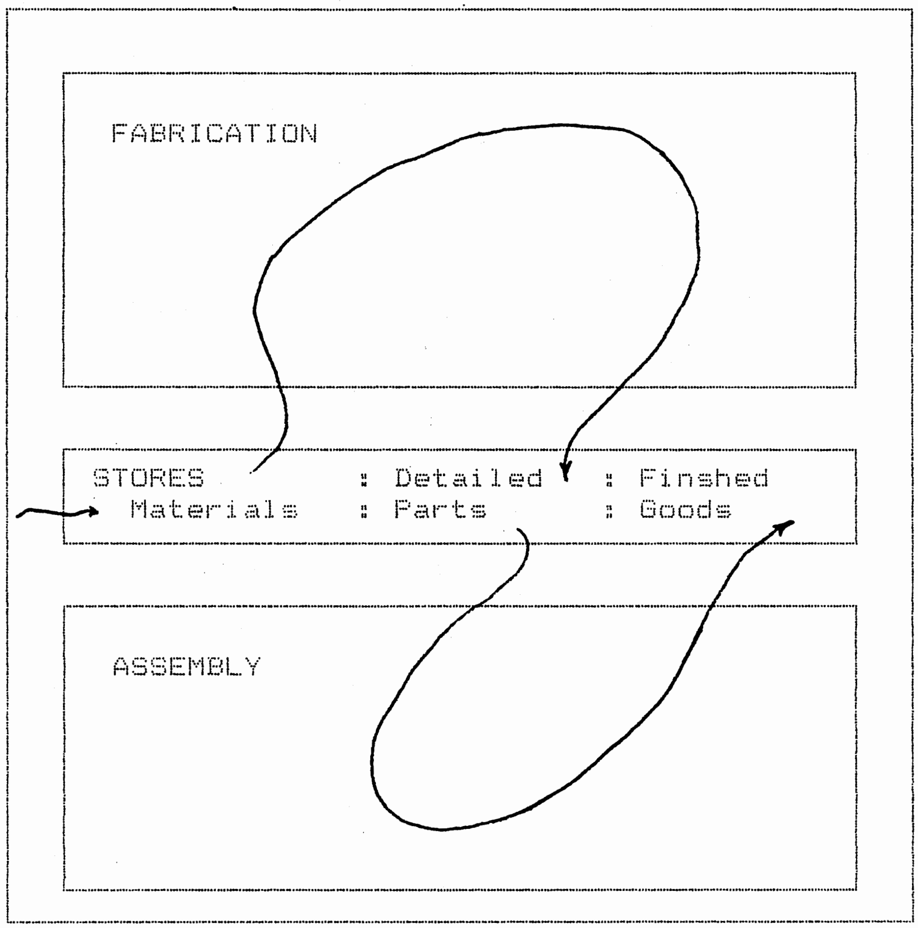


Figure 12. General Production Floor

products on the floor. In the following paragraphs a separate discussion of each area is presented.

1- The storage area

Depending on the activities performed in the organization, this area could be divided in to three different but interrelated sections. In general, the organization gets raw materials in order to transform them into finished goods. Finished goods for an industry could be considered as detailed parts in other industries; the definition is relative to the mission and kind of organization. Figure 13 shows the storage module in a production floor.

2- The fabrication area

Both the complexity of this area and its floor space vary with the organization. Where a fabrication area exists, it is usually organized into departments. Cutting, milling, heat treating, and drilling are just a few operations that are usually performed in these departments. Figure 14 is an example of a fabrication area. The arrows on this figure illustrate different paths for different products. The figure shows also an inspection station for all products that are presumed to be finished. The inputs to the fabrication area usually consist of some kind of raw material. At the other end, either detailed parts or finished goods or both are considered to be the output.

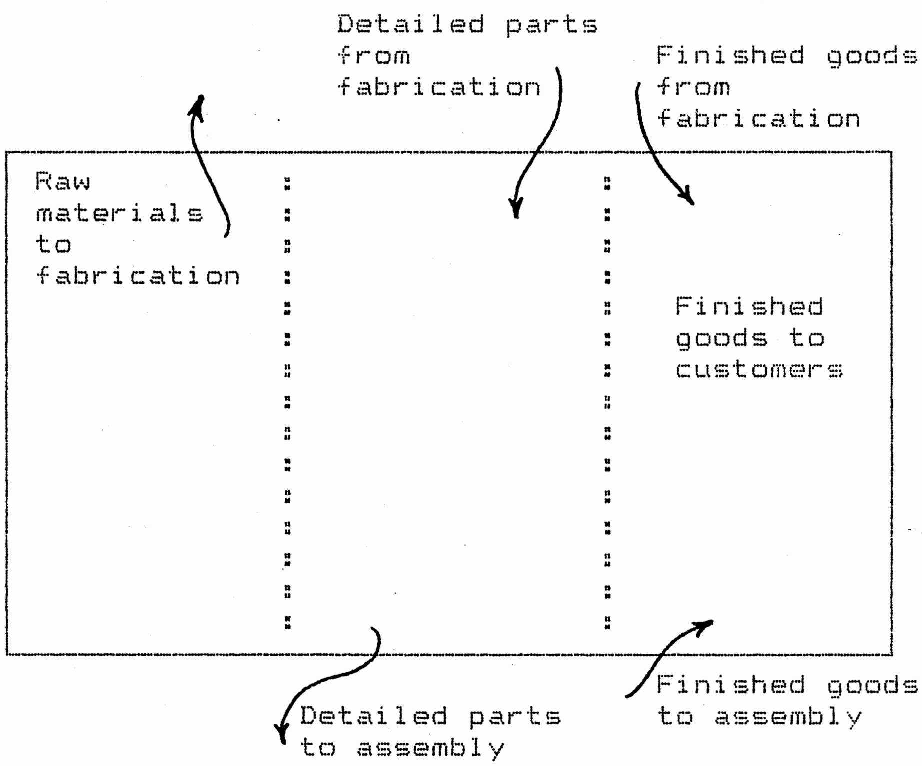


Figure 13. Storage Area and its Inputs and Outputs

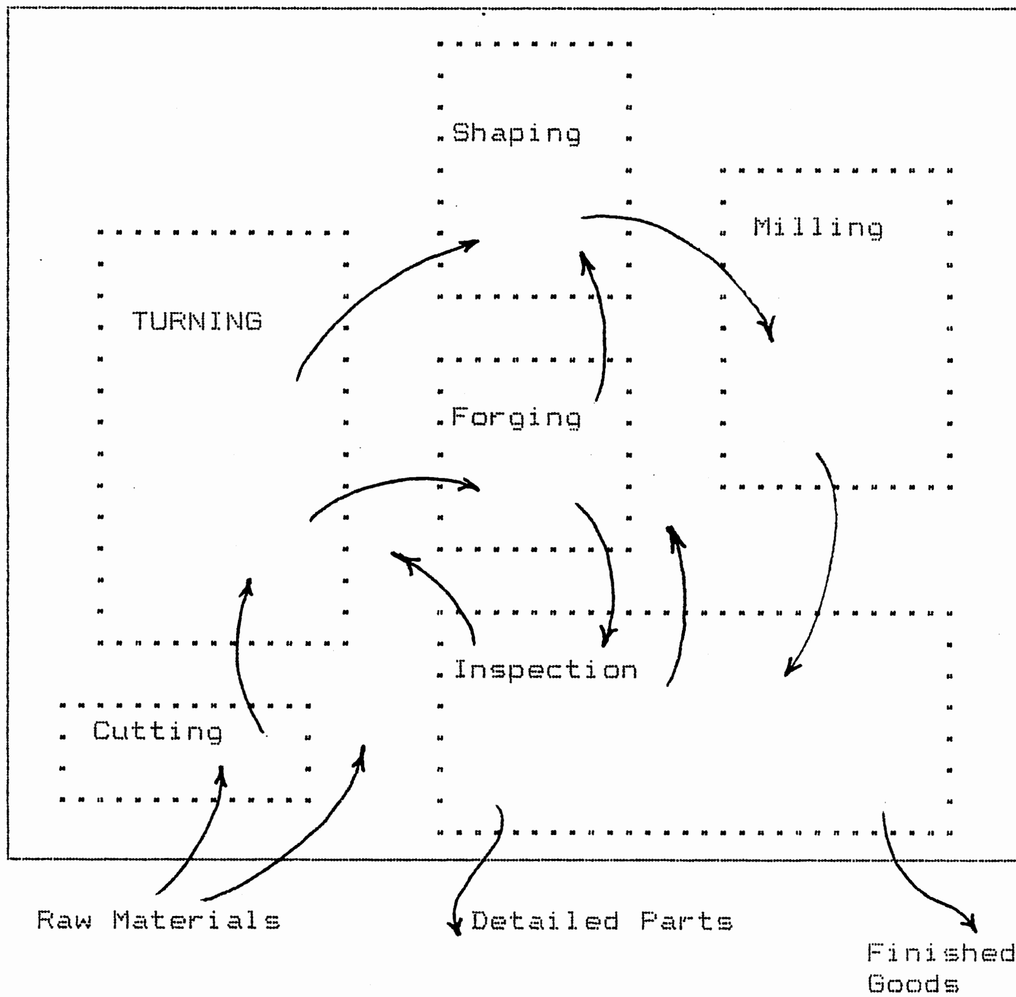


Figure 14. Example of a Fabrication Area

Looking closer at the fabrication area, more specifically if one department is considered, one can see that the department is in itself organized into cells which are in sets of similar machines. Figure 15 illustrates an example of one department in a fabrication area.

3- The assembly area

Assembly is usually the last operation performed on a product. The assembly area differs from one organization to another. Although, nowadays, automatic assembly is being introduced more and more to the manufacturing process, manual assembly is still very much in use. Figure 16 shows a possible layout of an assembly area. In the same assembly area, different products could be assembled at the same time. This could be done by dedicating different and independent assembly lines. A possible configuration of such lines is illustrated in Figure 17.

Trying to simulate a complete manufacturing operation could be very complex depending on the size of the organization. This simulation might even be economically infeasible in a very large operation, especially if detailed information is required. For this reason, the model designed in this project is to simulate the manufacturing environment at different levels. Figure 18 shows how the simulation could be performed at four different, but interrelated levels. At the first level, simulation of flow of materials and products between areas could be performed. Level two

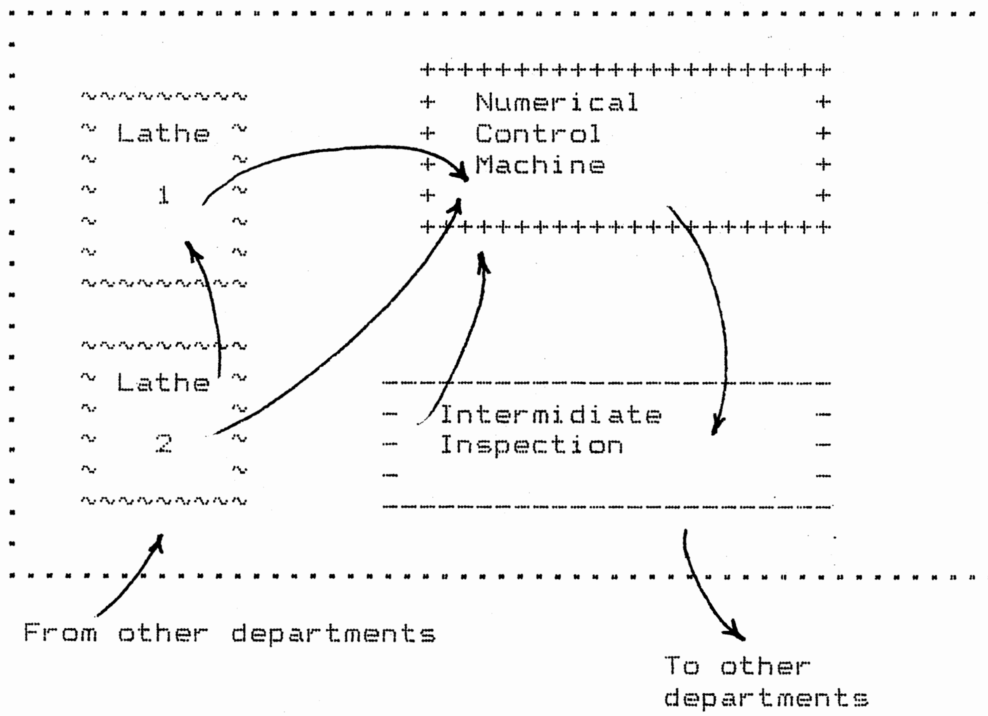


Figure 15. Turning Department

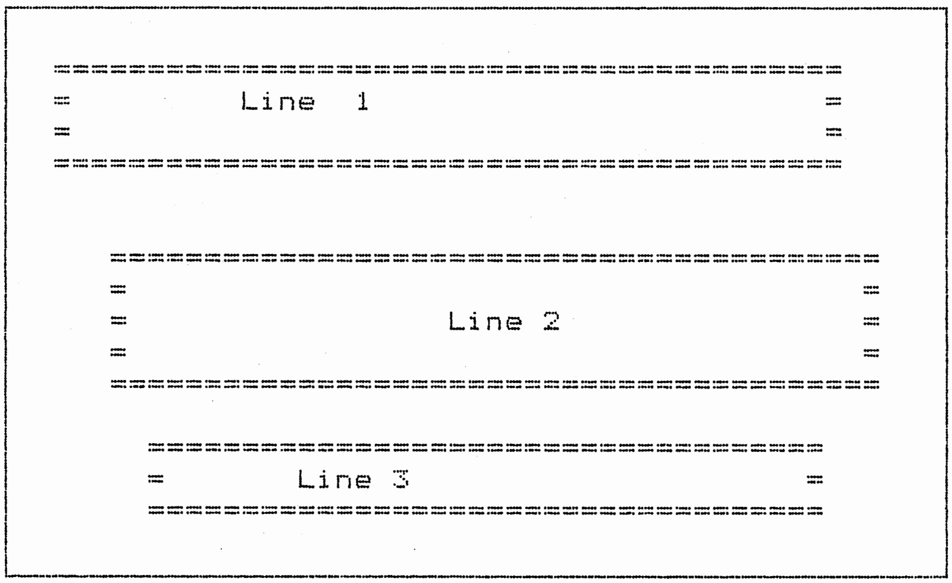


Figure 16. A Possible Layout of an Assembly Area

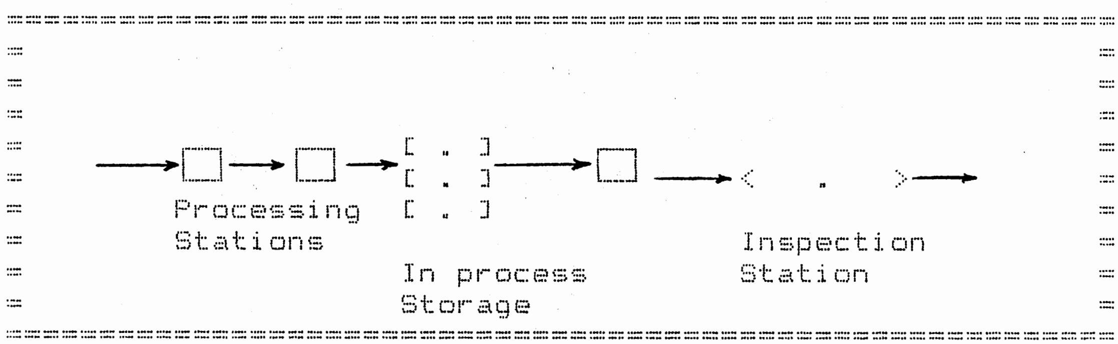


Figure 17. Assembly Line and its Components

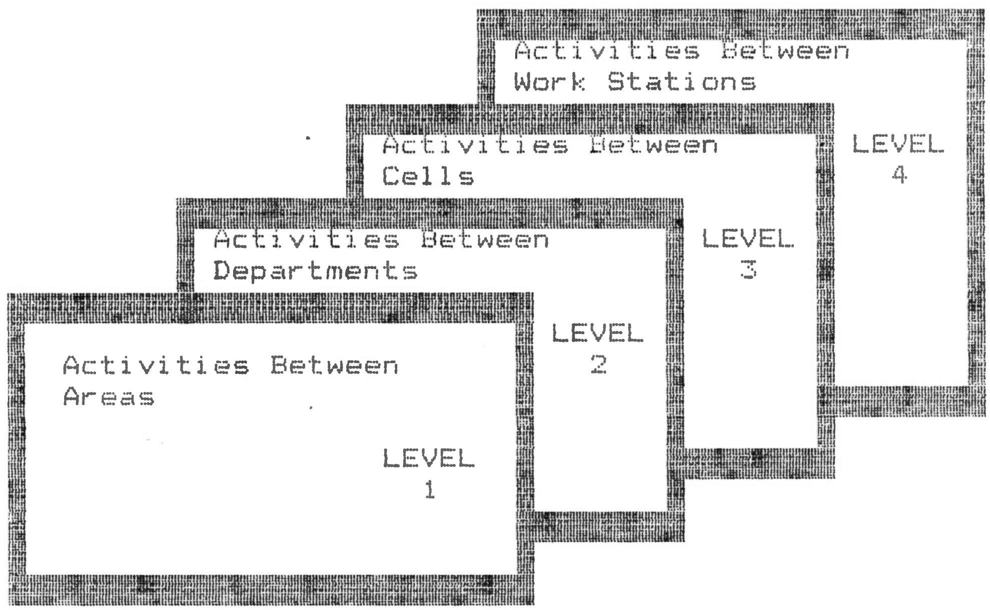


Figure 18. Levels of Activities

focuses more on the smaller picture: the simulation takes place now in one area and between its different departments. If more details are needed, the simulation could be taken further to a chosen department, in which flow of entities between cells is simulated. In the lowest level, the model gives the user the option to simulate the flow of products between workstations or machine operations. This break down of the simulation of the system into different levels does not take any powers out of the model. In fact, to meet the objectives stated in this project, a simulation of the whole operation is not mandatory. Figure 19 shows this breakdown.

Input to the Model

This model is designed so that manufacturing engineers could use it to simulate a manufacturing environment. Depending on the level used, the input to the model could vary. Generally, three sets of inputs are required from the operator. In the first set, the user describes the layout and the configuration of the manufacturing environment. In the second set, more detailed description of the flow of materials and the route they take is fed to the model. Also, all the time parameters, such as arrival rates and processing times, are included. In the third and last set, selection of some measures of performance takes place. A more detailed explanation of these three sets of inputs will follow.

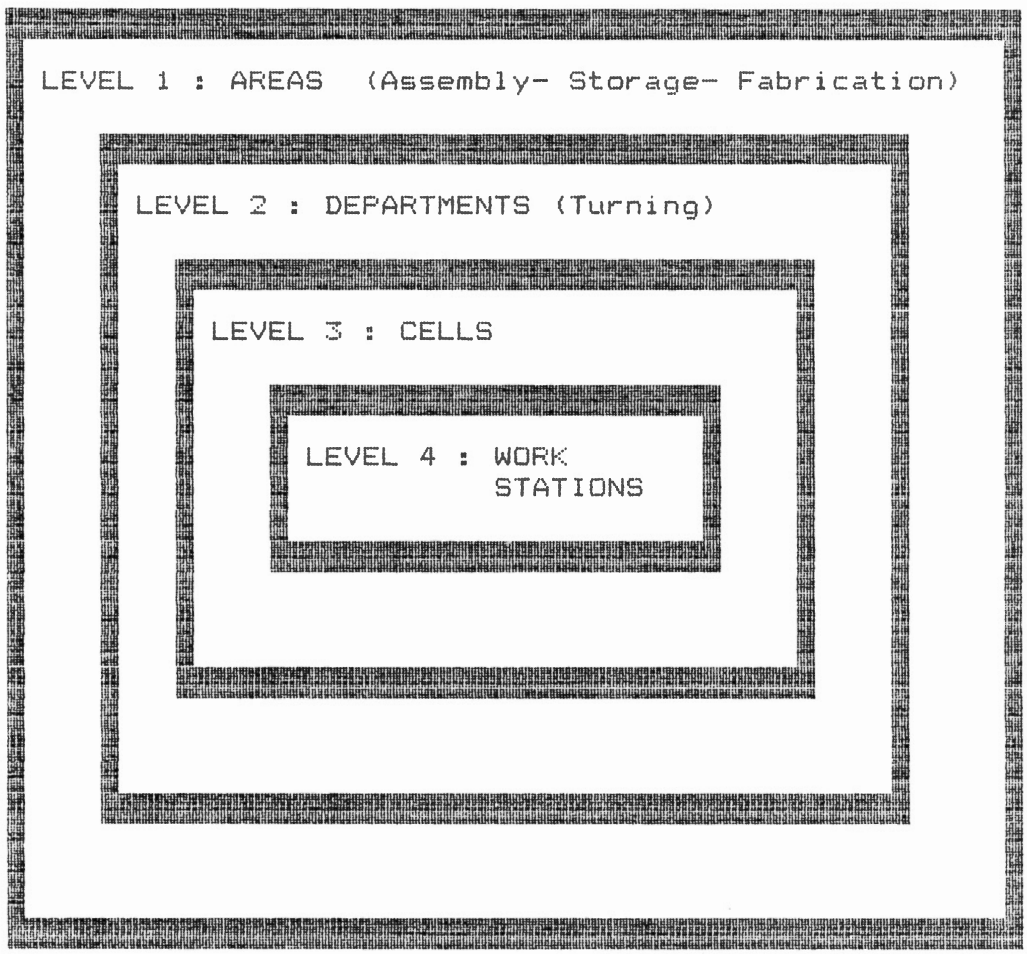


Figure 19. Levels of Simulation

Output From the Model

The purpose of the model is to evaluate a given situation. This evaluation is defined by some predetermined measures of performances. So, in other words, the main output from the simulation consists of the summary report including the statistics associated with these measures. For record keeping, an echo of the input of the user is also provided. A detailed discussion of the measures of performance chosen is presented in the next chapter. Because of the different levels included in this model, the analysis of the results provided from the simulation of one level could lead to a decision on whether or not a simulation at a lower level is needed.

Model Assumptions and Restrictions

In previous chapters, it has been mentioned that a simulation model can never be exactly like the real world situation: the model is just an attempt to represent, if possible, the situation or the configuration. Due to the nature of this project, some assumptions and restrictions had to be taken or were imposed. In fact, as its name states, the model is to be used for different situations. It should also be mentioned at this stage, that some of the restrictions were due to the limitations of the simulation language used (SLAM II). Also, because of the size of the project and the time constraint for submitting this study, some features were excluded from this first version of the model. These features, discussed in the Recommendation Section, could be installed at a later date.

The model is designed to accommodate a maximum of thirty five facilities and a maximum mix of five different products. It was felt that these numbers are big enough to demonstrate the power and utility of the software. Expending these numbers should not cause any problems; it is just a matter of expanding some arrays used in the programs.

A similar type of restriction concerns the number of random probability distributions used in the model. Four were judged to be the most used, they are: the constant distribution, the exponential distribution, the uniform distribution, and finally, the normal distribution. Once again, adding new distributions should not cause any problems.

The model is designed so that the product mix is fixed. Before performing the simulation, the user should already know the number of facilities, the number of products, and the route each product takes through the system. If two different products use the same machine, the first come first served (FIFO) approach is used. Later on, a module that assigns priorities could be added to the model. After each operation is performed, inspection takes place. The product either passes inspection and is scheduled for the next event, or is found to be defective . If defective, the product is considered to be either scrap, or to be reworkable. In this last case, it is assumed that the product is routed back to the same machine it exited before inspection; at that time it joins the queue with no assigned priority. It is also assumed that there is no travel time between facilities. In addition, delays due to machine breakdowns or machine maintenance are not included.

Other assumptions that had to be taken but should not affect the outcome of the simulation are:

* Single arrival distribution.
* The calling source is infinite.
* No jockeying of entities.
* No reneging of entities.
* No balking of entities.
* No time is lost between the release of an entity and the arrival of another.

Some other assumptions and restrictions were a consequence of the limitations of the simulation language (SLAM). In fact, before each facility, the queue has an infinite capacity; SLAM requires that the maximum number of entities in the queue has to be an integer and not a variable. For this reason, in this first version of the model the user cannot decide on this parameter. At a future date, this restriction could be omitted. Last but not least, once the user inputs all the data and the simulation has already started, changes cannot be performed and feedback cannot be provided until the simulation ends. The reason for this is the fact that, up to the date this project is being developed (March 1987), SLAM does not provide interactive mode.

To summarize, the restrictions and assumptions that had been made do not make the model unrealistic, and by no means, jeopardize the objectives sought from this project, especially when one knows that most of the mentioned restrictions and assumptions could be omitted in a later version of this software (this does not include the restrictions which are due to the limitation of the language used: SLAM).

Finally, in the remaining chapters, the following terms are to be used: areas, departments, cells, and workstations. A definition of each term is shown below:

1- Areas: different areas of the layout (i.e. production area, assembly area, etc).

2- Departments: different departments in the same area (i.e. turning department, milling department, etc).

3- Cells: group of service stations in one department.

4- Workstations: service stations.

CHAPTER VII

MEASURES OF PERFORMANCE

The purpose of running any simulation model is usually the search for some statistics that would help in making decisions or inferences about the system under study. These statistics are considered to be the measures of performance of the system. Depending on the nature of the system, different measures of performance are chosen.

Because this project is concerned primarily with manufacturing systems, most of the measures of performance are time related. To study the system and its performance, statistics had to be collected for both the facilities and the entities. One of the advantages in using SLAM is that it automatically updates such statistics.

The first measure of performance to discuss is the service activity statistics. SLAM provides the following:

1- The service average utilization.

2- The maximum idle time of the facility.

3- The current utilization of the facility when the simulation ends.

4- The number of entities serviced in the facility.

These statistics were found to be sufficient to analyze the facilities status.

The second type of statistics provided automatically by the language concerns the file statistics. The queues before the facilities fall in this category. SLAM provides the following in its report:

1- The average length of the queue.

2- The maximum length of the queue.

3- The length of the queue at the end of the simulation.

4- The average waiting time of entities in the queue.

The other kind of statistics concerns the entities themselves. Although these statistics are also collected by SLAM, they are not provided unless requested by for using specific nodes. For this study, the following statistics were collected:

1- The average time that all products spend in the system.

2- The average time that each kind of product spends in the system.

3- The average time between completion of products.

4- The time spent in the system by a defective product before it is scrapped.

5- The time between two scrapped products, both, similar and different.

6- The number of products that were serviced during the time of the simulation.

7- The number of products found to be defective and had to be thrown out.

8- The minimum time a product had spent in the system.

9- The maximum time a given product had spent in the system.

All the previous statistics are collected both for all the products, and for each kind of product separately.

Although other statistics could have been collected using user code, it was found that the statistics provided by the language were enough for the user to infer about the performance of the system as a whole, and the performance of each facility separately. Should this model be expanded, other measures of performance could always be added. The use and interpretation of each of these measures will be discussed later through some examples.

CHAPTER VIII

SOFTWARE DESCRIPTION

How Simulation Works

From the standpoint of queuing theory, a waiting line situation takes place when entities arrive to the system and join the queue. The service facility then choses the entities from the waiting line for service, according to predetermined rules and priorities. When service is completed, the process of choosing a new entity to be serviced is repeated. Thus, the principal factors in a queuing simulation are the entities and the servers. In this particular case, the system has a maximum of thirty five service facilities and can accommodate a mix of up to five different products at one time.

Programs Description

At the early stages of the software design, the idea was to have a single program, that is FORTRAN based, which interfaces with the simulation language SLAM. Knowing that one of the objectives of this project is to have an interactive software that prompts the user for the primary data needed for the simulation, the program had to have this property. One problem arose then: SLAM was loaded on the IBM main frame 3081; to use this language, simulation programs had to be submitted as batch jobs. For this reason, it was impossible to run interactive mode using just one program. To solve this problem, two and only two alternatives had to be considered: either have the user type all the needed information as data statements, or have two different programs. The first alternative was rejected because it conflicts with one of the objectives of this project; to exclude the user from having to change the code of the software. Consequently, the second alternative had to be chosen.

The software includes then two separate programs which interface through a permanent file used as a storage of the input data. The first program is the input program, the second one is the actual simulation module. Figure 20 is the overall picture of the software.

PROGRAM 1 : THE INPUT PROGRAM

This program includes a main program and thirteen subroutines. The overall organization of this program is shown in Figure 21. This program is menu driven. All the subroutines operate separately. Data is passed back and fourth through a common block (USER1). Throughout this program, data is stored in the already built file. Before discussing each subprogram independently, it might be helpful to go through the main variables used and their use:

1- LEVEL : level of the simulation. (See Figure 19.)

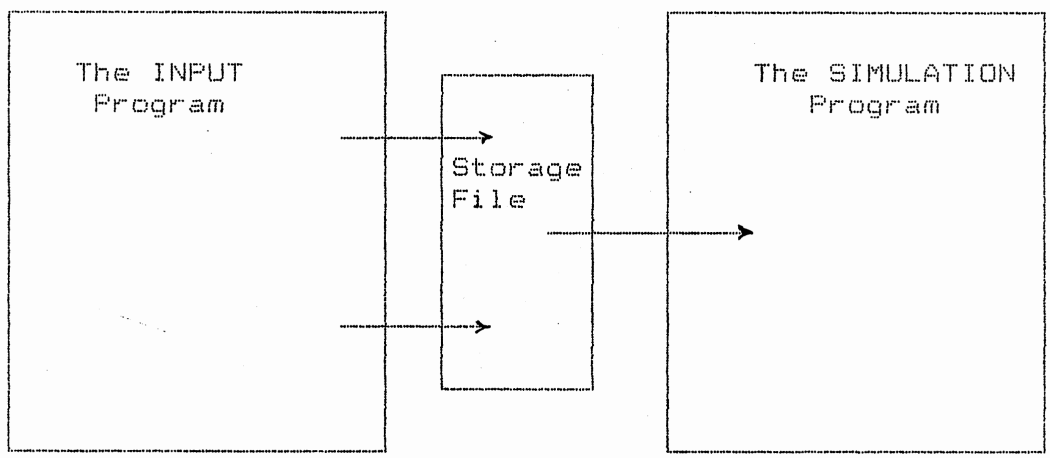


Figure 20. Overall Picture of IGPSSMP

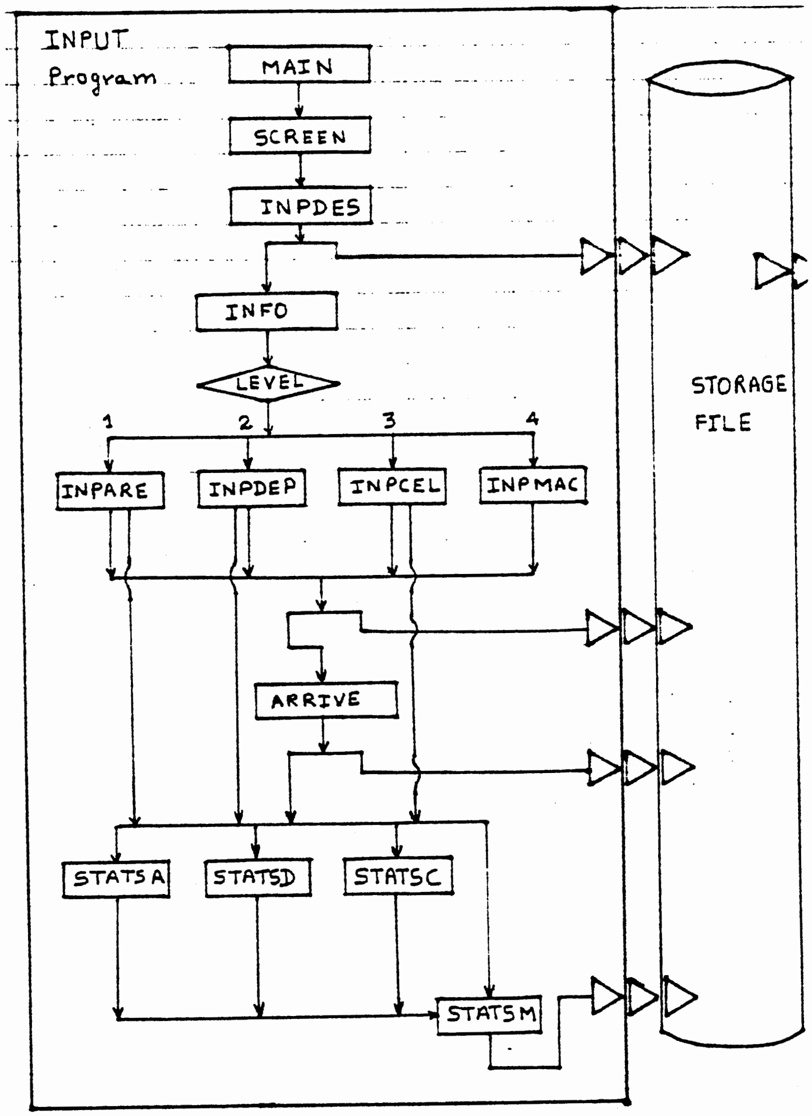


Figure 21. Overall Organization of the INPUT Program

2- NPROD : number of products arriving to the system.

3- MACH : number of service facilities in the system.

4- TIMED : duration of the simulation.

5- NMACHK(5) : stores the number of facilities each product needs to go through before it leaves the system.

6- ENTITY(5,35,2): the first argument marks the product, the second marks the service station, the third stores respectively the type of the processing function and the route the product takes.

7- PARA(5,35,3) : the first argument is the product, the second is the facility, the third represents respectively, the type of processing function, the first and second parameter of this function.

8- ARVAL(5,3) : the first parameter is for the product, the second is respectively for the type of arrival function and its two parameters.

9- DEF(5,35,2) : the first parameter is for the product, the second is for the facility, and the third is respectively for the proportion of products that passed inspection and the proportion of products found to be defective and cannot be reworked.

In the following sections, the different subprograms and their use will be described. The main program constitutes the driver of the program. Its listing (L.) and flowchart (F.) are shown in Appendix B. The subroutines will be discussed in the order they are called in the program.

SUBROUTINE CLSC : L. IN APPENDIX C

This subroutine clears the screen.

SUBROUTINE SCREEN : F. AND L. IN APPENDIX D

This subroutine is used only to provide the user with information about the used program.

SUBROUTINE INPDES : F. AND L. IN APPENDIX E

In this subroutine, the user decides on both the level and the duration of the simulation. Depending on this level, the appropriate input subroutines are accessed.

SUBROUTINE INFO : F. AND L. IN APPENDIX F

The purpose of this subroutine is to collect general information about the simulation to be performed. This information is vital for record keeping.

SUBROUTINE INPARE : F. AND L. IN APPENDIX G

Through this subroutine, the operator enters all the information when a simulation of areas (level one) is needed. The user is prompted for:

1- The number of products to simulate.
2- The number of service facilities in the system.
3- The number of facilities used by each product.
4- The route that each product takes through the system before it exits.
5- The processing function of each product for each facility.
6- The parameters of the processing function of each product for each facility.
7- The proportion that passes inspection of each product for each facility.
8- The proportion that is found to be defective and judged to be scrap.

Throughout this subroutine, the information received from the user is transferred to the storage file.

<u>SUBROUTINE</u> <u>ARRIVE</u> <u>:</u> F. AND L. IN APPENDIX H

This subprogram is also menu driven. It is used to accumulate information about the products to be simulated. This subroutine asks the user for:

1- The type of arrival rate function to the system for each kind of products.
2- The parameters of these functions.

Also, this subroutine updates the storage file.

SUBROUTINE STATSA : F. AND L. IN APPENDIX I

This is where the user has the option to ask for statistics to be collected on the measures of performance. In this first version of MDMSS, the user can decide on three measures of performance. This subroutine also accesses the storage file.

SUBROUTINE INPDEP : F. AND L. IN APPENDIX J

This subroutine accommodates the second simulation level: the department level. It operates exactly like INPARE.

SUBROUTINE STATSD : F. AND L. IN APPENDIX K

This subroutine, just like STATSA, gives the option of deciding on some measures of performance.

SUBROUTINE INPCEL : F. AND L. IN APPENDIX L

It is for the input for simulation at the third level: the cell level.

SUBROUTINE STATSC : F. AND L. IN APPENDIX M

This subroutine has the same role that STATSA has.

SUBROUTINE INPMAC : F. AND L. IN APPENDIX N

It is used to input information when the simulation is performed at the fourth and last level: the machine level.

SUBROUTINE STATSM : F. AND L. IN APPENDIX O

Updates the storage file with the measures of performance needed at the machine simulation level.

COMMENTS CONCERNING THE INPUT PROGRAM :

One can wonder why some of the input subroutines and statistics subroutines were repeated although they are very similar; the reason is simple: during the implementation of the software it was anticipated that other studies and projects might follow to update this software. Realizing how important the structure of a program is in easing future updates, it was decided that separate subprograms need to be written no matter how similar they are.

Also, it might be of use to future projects to explain how this program runs on the IBM main frame 3081. Because this system does not offer an automatic interactive mode, extensive research had to be made to bypass this setback. Finally, it was found that the only way to run interactive programs is to build a personal library. A detailed description and documentation of this procedure is shown in Appendix P.

PROGRAM 2 : THE SIMULATION PROGRAM (See Figure 22)

The most challenging portion of this project was to develop a simulation model that can fit many situations and

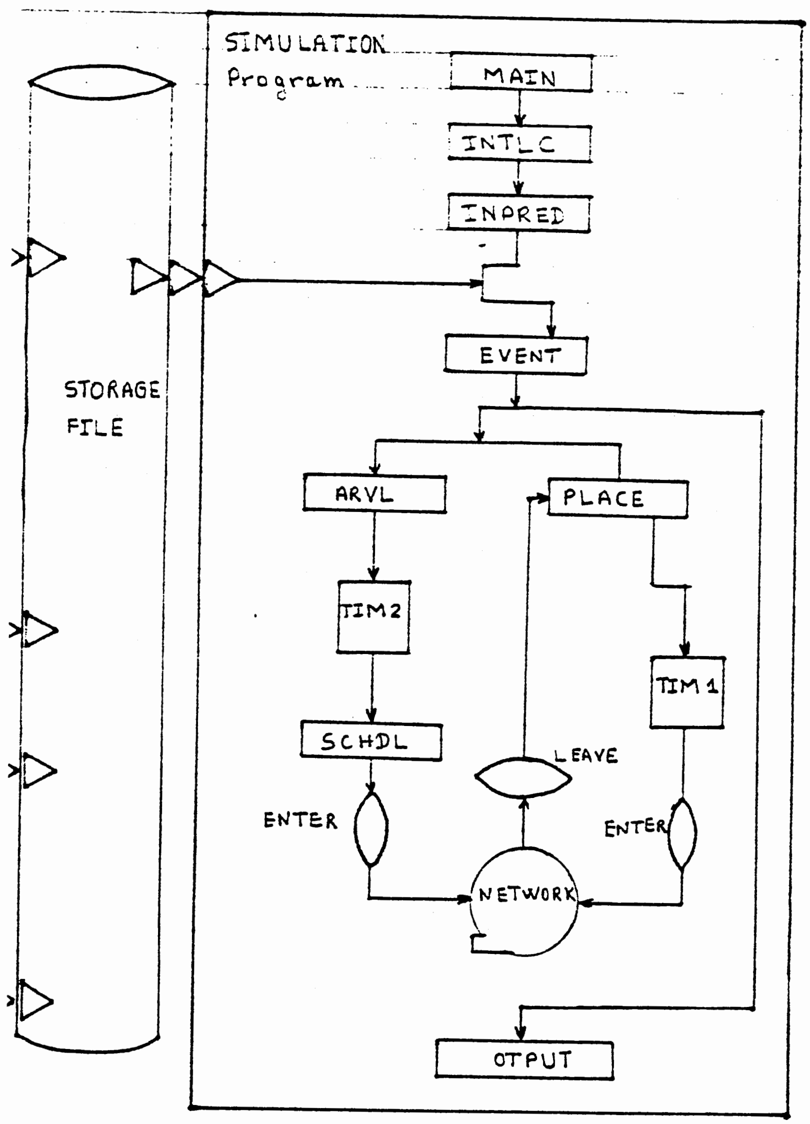


Figure 22. Overall Organization of the SIMULATION Program

systems (primarily in manufacturing) without accessing the code to make changes (even if they were minor). After an extensive study, it was found that to create such a model, one needs to take advantage of both the flexibility and power of the FORTRAN language and the capabilities of SLAM in filing entities, collecting and updating statistics, and processing the event calendar. Network Modeling With User-Written Inserts seemed to be the appropriate choice.

The developed program ended up including a main program, seven user-written subroutines, two user-written functions, and one network. Information was passed from the different subprograms through three common blocks: SCOM1, USER1, and USER2. SCOM1 is a common block set in the SLAM package, USER1 is the same common block used in the first program, and USER2 is a common block that transfers the following variables after they were processed from the storage file:

1- DATE : stores the date the simulation was performed.

2- PROJEC : has the simulation project number.

3- DEPART : is the department that required the simulation.

4- NAME : is the name of the operator.

5- REFE : is the reference of the simulation.

The model also used thirty five files which were dedicated to the queues for the maximum number of facilities in a given system. Last but not least, six attributes were found to be enough to manipulate entities (products) flowing in and from the system. These attributes are :

1- ATRIB(1) : this attribute is to differentiate between the entities (the kind of product).

2- ATRIB(2) : it is the process time for a given entity at a given machine.

3- ATRIB(3) : this attribute stores the address of the next facility the entity needs to visit.

4- ATRIB(4) : this attribute is set to (-1) when the product is ready to exit the system.

5- ATRIB(5) : is the probability that this product inspection.

6- ATRIB(6) : is the probability that if this product fails inspection it will be scrapped.

The main program (F. and L. in appendix Q) sets the needed parameters, such as the unit from where SLAM reads and the unit to where SLAM writes, then it calls SLAM which initiates the simulation. In the next paragraphs, the different subprograms are discussed to show the role they play in this general purpose simulation model.

<u>SUBROUTINE</u> <u>INTLC</u> <u>:</u> F. AND L. IN APPENDIX R

SLAM calls this subroutine as soon as it is executed. This subroutine calls INPRED, which reads all the relevant information from the storage file, and then schedules the arrival of the first products.

SUBROUTINE INPRED : F. AND L. IN APPENDIX S

As was mentioned earlier, the role of this subroutine is simply to read the information stored in the file.

SUBROUTINE EVENT : F. AND L. IN APPENDIX T

To perform the simulation, only two events were found to be necessary. This subroutine branches to the appropriate event whenever it is called. These two events are (1) the scheduling of a new arrival or (2) placing the entity (product) in the right portion of the network which, in the real system, symbolizes putting the product before the appropriate facility.

SUBROUTINE ARVL : F. AND L. IN APPENDIX U

In this subroutine, the kind of the next product to be scheduled is determined, according to this specification, the time of the next arrival is found and finally, the product is inserted into the network.

SUBROUTINE PLACE : F. AND L. IN APPENDIX V

This subroutine is the driver of the simulation. In fact, first, it checks to see if the duration of the simulation has been reached, and if the simulation continues, the subroutine checks the status of the entity just received. This subprogram updates the number of facilities the entity went through, then it compares this number to the number of facilities initially scheduled for the entity. If it is found that the entity has just left the last facility, attribute four is set to (-1) and the entity is put back to the network; otherwise, the attributes of the entity are updated (new facility address in ATRIB(3), new process time at that facility, and new defective proportions) and the entity is sent back to the network.

SUBROUTINE OTPUT : F. AND L. IN APPENDIX W

This subroutine, called by SLAM automatically when the simulation ends, gives an echo of the data used in the performed simulation.

FUNCTION TIM1 : F. AND L. IN APPENDIX X

This function simply generates the processing time for a certain product at a certain facility.

FUNCTION TIM2 : F. AND L. IN APPENDIX Y

Similar to TIM1, TIM2 generates the arrival rates of the different products to the system.

NETWORK : FIGURE AND LISTING IN APPENDIX Z

The network is formed of three major parts. The first is a decision making section, the second is a process section, and the third is the statistic collection section. When an entity gets to the network through the ENTER node, it is sent back to the discrete portion of the program with an event equal to two which means that the entity is going to subroutine PLACE. After all attributes are updated in that subroutine, the entity is sent back to the network. If it was determined that the entity has just been processed by the last service facility (ATRIB(4)=-1), the entity is terminated; on the other hand, if the entity still needs to be processed, using the third attribute (containing the address of the next service facility) the entity takes the appropriate branch and joins the queue before that facility. In the latter case, the entity either waits if the facility is busy, or advances if it is idle. At each facility, after the entity has been serviced, inspection takes place. Depending on the data given by the operator, a proportion of the entities passes inspection, those entities are sent back to the discrete code to be processed; if the entity (product) is found to be defective, it is either judged to be reworkable and sent back to the facility that it just left, or it is scrapped and terminated from the system. Before any entity leaves the system, it goes through the appropriate nodes so that statistics can be updated.

COMMENTS ABOUT THE PROGRAM

Without interfacing discrete event and network modeling, writing this program would have been very hard, if not impossible. Using network modeling alone would not have done the job, and using just discrete modeling event would have been a lot harder to develop. Because the information needed by the simulation was read from a file, the JCL used to run the simulation was slightly different from the usual one. Appendix A1 shows that JCL. Finally, the procedure that needs to be followed in running the software is to be discussed in the user manual which will follow in a later chapter.

CHAPTER IX

THE SOFTWARE QUICK USER MANUAL

Log on to the System

As it was stated before, to perform a simulation using the developed software requires running two separate programs; the first one inputs the needed data for the simulation, while the second program is actually the simulator. In this chapter, each program is to be discussed independently. But, before getting to the programs, it is necessary to describe the procedure of logging on to the system. The system used is the Oklahoma State University IBM 3081 main frame. First, the user needs to find a terminal connected to the University Computer Center (UCC) network. Then, the following steps are to be taken in this order:

1- Turn on the power through the power switch.

2- Depending on the kind of terminal and location, either Ctrl T or Alt T needs to be pressed at the same time (most of the terminals work with Ctrl T).

3- At this stage, the system should reply that the terminal is connected to the UCC network. The user is asked then to enter the system to work on: IBM or VAX. The user needs to type IBM (in capital letters).

4- One more time the system prompts the user to enter a new piece of information: the application. The user needs to type TSO.

5- The system then asks for the user number. For this project the number is U11296A (valid only for model development).

6- Then, the password needs to be entered (Available on request).

If these steps are carried out successfully, the user should be logged on with the screen showing the READY mode.

The Input Program

This program had to be loaded in a user library. The reason for this is because it had to be interactive. To run the program it was found to be mandatory to use a CLIST program. The listing of this program is shown in Figure 23:

```
0000010ALLOC F(FT03F001) DS(FT03F001.CNTL)
0000020CALL 'U11296A.LOAD.LIB(THESIS2)'
0000030EXIT
```

Figure 23: CLIST Program.

The first statement in this short program assigns unit 3 to the already built file (FT03F001.CNTL). FT03F001.CNTL was created the first time the INPUT program was submitted. The second statement of the CLIST program causes the INPUT program to run. In fact, when that program was submitted the system compiled it, linked it, and stored it in the user already built library (LOAD.LIB). The argument THESIS2 is the name of the compiled and linked version of the INPUT program in LOAD.LIB. To run this program, the screen of the terminal should show the READY mode. The user just needs to type the following statement:

EXEC THESIS2

At this point it is assumed that the user has all the data needed for the simulation: a detailed description of the input data will follow shortly. The user should not worry about making mistakes because the program is user friendly: it always checks the validity of the user input before it proceeds to the next question. Also, the program was written so that the user does not have to worry about any formats in typing the data: all inputs are free format.

Executing THESIS2 causes the INPUT program to run. The first screen, shown below in Figure 24, gives general information about the program.

```
                    MDMSS


                 VERSION 1.0


               DEVELOPED BY :
               IMED JAMOUSSI

       PLEASE REPORT ANY PROBLEMS TO :

              Dr. JOE H. MIZE
```

Figure 24: Screen 1

The user is then asked to enter some general information about the simulation. These are:

1- The date : the first eight characters are valid (i.e. 04-12-87 which is the 12 of April 1987).

2- The project number: it can not have more than 12 characters (i.e. SIMU1100).

3- The department the simulation is done for: the first twenty characters are meaningful (i.e. PRODUCTION CONTROL).

4- The operator name: the first twenty characters are recognized (i.e. JOE DOE).

5- The simulation reference: it has no more than five characters (i.e. 00001).

Next, the menu shown below is provided.

1- AREA SIMULATION	2- DEPARTMENT SIMULATION
3- CELL SIMULATION	4- MACHINE SIMULATION

The user has to enter one of four numbers (1-4). If he(she) fails to do that, the menu appears one more time.

The duration of the simulation is the next input. This time is a real number. The maximum is set to be 999999999.99 . The unit of time is left up to the user, but consistency is required.

Depending on the simulation level chosen, the program asks the user for the parameters of the system (properties of both the entities going through the system and the service facilities in it). A great similarity exists in the input of the four simulation levels. For this reason, only one level will be discussed (machine simulation, level 4).

Now the user is asked for:

1- The number of products: a maximum of 5 different products could be simulated at the same time. It is expected that the user enters a number between one and five inclusive. If he(she) fails to do so the same question reappears.

2- The number of machines in the layout or the system: 1 is a minimum, 35 is the maximum for each product.

3- The number of operations needed before the product leaves the system. The user can have as many as he(she) wants, no restriction exists.

4- The user has to chose then, according to the menu shown below, the arrival probability distribution for each product.

1- CONSTANT	2- NORMAL
3- UNIFORM	4- EXPONENTIAL

If CONSTANT, the time constant is to be entered. This number has to be greater than zero. If the function is NORMAL, the mean and the standard deviation are the next inputs. The mean has to be greater than zero, the standard deviation greater than or equal to zero. If the UNIFORM distribution is the arrival function, the lower and upper limits are the parameters asked for. The lower limit has to be greater than or equal to zero, the upper limit has to be greater than the lower limit. In the last case, the EXPONENTIAL distribution, the mean has to be greater than zero.

5- Now, for each product the route needs to be described to the program.

6- Next, the process probability distribution and their parameters are needed for each product at each workstation. The same distributions that were used to determine the arrival rates are included here.

7- The probability of a defective product is the next property to input. For each product and at each workstation, the user needs to enter the probability that the product passes inspection and the probability , if the product is defective, it cannot be reworked. These two numbers have to be between zero and one with their sum less than or equal to one.

8- Last but not least, the user has the option to chose extra statistics concerning some prechosen measures of performance. For each question, if the user wants to take the option, "1" needs to be typed; any other number means the opposite. These statistics concern:

- The inspection statistics.
- The times between completion of products.
- The interval statistics (time in system).

If this stage is reached, the system should be back to the READY mode because the run of the INPUT program THESIS2 has just been finished.

The Simulation Program

Compared to the INPUT program, the user does not have to do very much to have the simulation performed (which is one of the main objectives of this project). Assuming that the user is still on the system and the ready mode is displayed on the screen, to run the simulation the user needs to do the following:

1- First, free the storage file updated by the INPUT program. The user needs to type the following:

FREE FT03F001.CNTL

2- Second, submit the job by typing the statement below:

SUBMIT THESIS1.DATA

The system should prompt back and ask for the job character. Any alphanumeric single character could be typed (i.e H). At this stage the job should be submitted and execution should have begun. If the status of the job needs to be checked and the READY mode is still displayed, the user needs to type IOF . Through IOF and assuming the run was completed, the user can see the output of the simulation after choosing the job needed and typing 8 which will show the SLAM report on the screen. The output could also be sent to the local printer just by releasing the job, assuming that the user is still in the IOF mode (i.e. 1 R , which asks the system to release job number one to the local printer).

How to Read the Output Report

The SLAM output report has three parts: the SLAM echo report, the user echo results, and the SLAM summary report. The SLAM echo report is just a feedback that shows the general options, the limits on the files, the file summary, the random number streams, the initialization options, and the variables allocation. This first report should not constitute any interest to the analyst. The second report first gives some general information about the software and then echoes the input of the user. Reading this feedback should not cause any problems. It just provides the summary of each product separately (the arrival distribution and parameters, the route the product takes in the system, the processing time at each facility, and the information related to inspection).

The last and main report, the SLAM summary report, provides the statistics for variables based on observations, the file statistics, and the service activity statistics.

Statistics for Variables Based on Observations

This section of the output report gives the mean value of the time that an entity spends in the system and its standard deviation, both the maximum and minimum times that entities spent in the system, and finally the number of entities that reached that node. The nodes labels and their use are discussed below:

1- P? DEF TIME : collects statistics for defective products without distinguishing the different kinds.

2- P1 through P5 DEF TIME : same as before, except that each kind has it own statistics.

3- P1 through P5 BET DEF : collects statistics on the time between two defective products.

4- T SYSTEM TIME : time spent in the system by entities that passed inspection.

5- P1 through P5 SYSTEM TIME : time spent in the system by each product.

6- P? BET COMP : time elapsed before two entities left the system regardless of their kind.

7- P1 through P5 BET COMP : same as before but statistics are collected for each product type separately.

File Statistics

This section displays statistics for each of up to thirty five queues, depending upon the number of machines in the system. The label corresponds to the facility, for example Q11 is the queue for machine 11. These statistics give the average length of the queue and its standard deviation, its maximum and current length through the simulation, and finally the average waiting time in the queue.

Service Activity Statistics

For each facility, all the statistics are provided. They are the service capacity, the average utilization, the current utilization when the simulation ended, the entity served by the facility, and the maximum idle and busy times.

Using the statistics provided in this report will be discussed later through some examples. It should be noted though, that, depending on the system and the purpose of the simulation, the importance of each kind of statistic varies.

How to Prepare for a Simulation Session Using the Software

To help the user manage his(her) time better and to avoid possible confusion when using this software it is advised that he(she) complete Form A and Form B before even logging on to the system. Form A, shown in Figure 25, is designed to contain the general information needed for the

Date :

Project Number :

Department :

User Name :

Simulation Reference:

Simulation Level :

Duration of Simulation :

Number of Products to Simulate :

Number of Machine in the Layout :

Figure 25. Form A

simulation. Form B, shown in Figure 26, is used to keep records of the products and their different properties. It should be noted that for each kind of product a separate Form (Form B) is needed.

To illustrate how to fill the forms already mentioned, the following example is considered:

A simulation is to be performed on April 12, 1987. The project number for this simulation is IHJ123. The department that required this simulation is PRODUCTION. The name of the person that was asked to conduct the simulation is Joe Doe. The simulation reference is 1. The simulation is to be performed at the workstation level: level 4. The time of the simulation is 150 minutes. Two products are to be considered. The route of these products are shown respectively in Figure 27 and Figure 28. Five machines are used. Four operations are used for product one and three for product two. Product one arrives according to a constant distribution every 14 minutes. Product two arrives according to a constant distribution every 12 minutes. The processing times for product one are according to a constant distribution of 8 minutes in machine 1, a uniform distribution with a lower limit of 3 minutes and an upper limit of 10 minutes in machine 2, an exponential distribution with a mean of 6.5 minutes in machine 3, and a normal distribution with mean of 7 minutes and a standard deviation of 3 in machine 4. Product two is processed according to a constant distribution in all three machines. The constants

PRODUCT # :

Number of Operation Needed for the Product :

Arrival Probability Distribution
Type :
Parameter 1 :
Parameter 2 :

Facility	Process Probability Distribution			Inspection *	
	Type	Parameter		Good	Scrap
		1	2		
________	________	________	________	________	________
________	________	________	________	________	________
________	________	________	________	________	________
________	________	________	________	________	________
________	________	________	________	________	________

* This number should be between Zero and One.

Figure 26. Form B

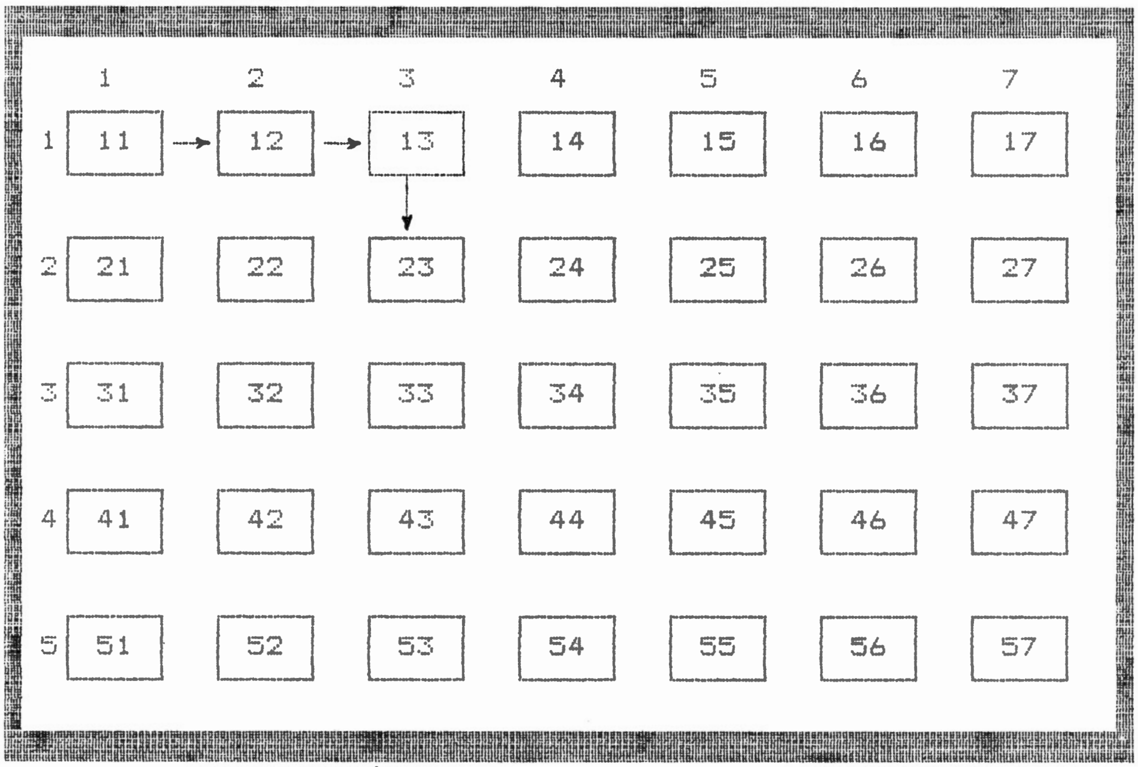


Figure 27. Route of Product 1

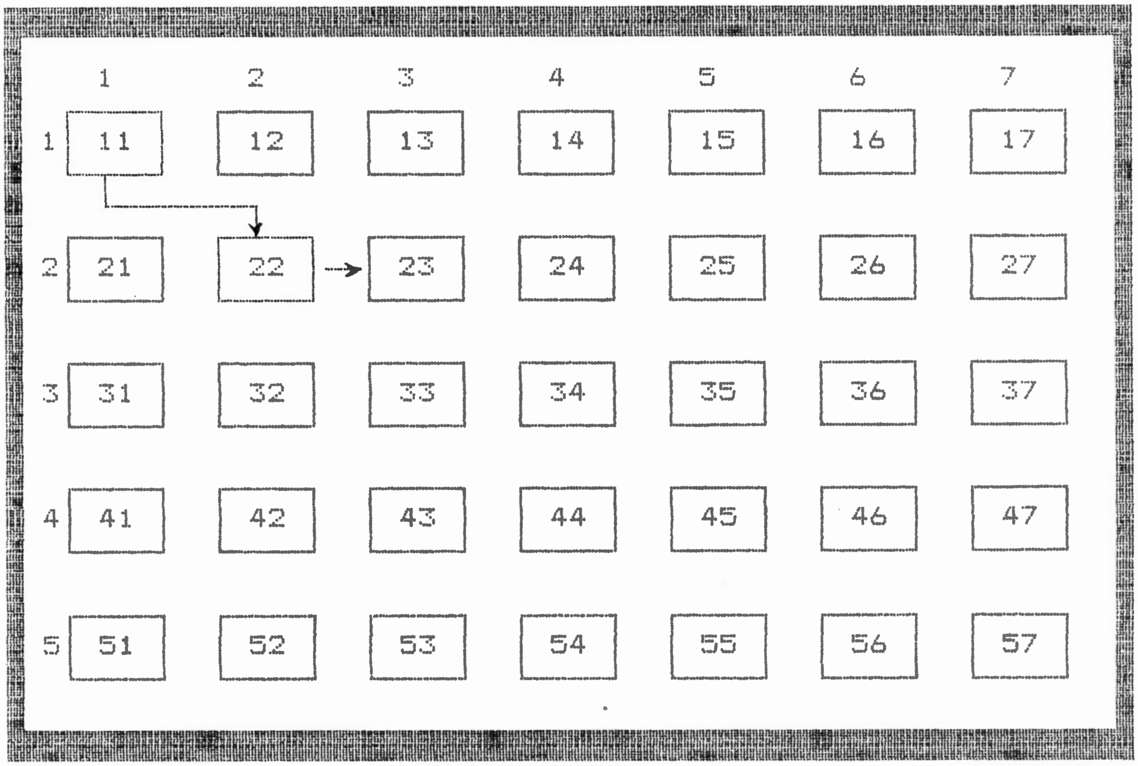


Figure 28. Route of Product 2

are 8, 9, and 7 minutes respectively. It is assumed that all the machines produce no defective products except machine three of product one which has a probability of .9 good products and a probability of .1 scrap, machine four of product one which has a probability of .9 good products and a probability of .01 scrap, and machine three of product two which has a probability of .85 good products and a probability of .11 scrap. Figures 29, 30 and 31 show the forms filled for this example. Figure 32 on pages 87 to 92 show the output obtained from running the previous example.

Date : 04-12-87

Project Number : IHJ123

Department : PRODUCTION

User Name : JOE DOE

Simulation Reference: 1

Simulation Level : 4

Duration of Simulation : 150.00

Number of Products to Simulate : 2

Number of Machine in the Layout : 5

Figure 29. Form A of Example

PRODUCT # : 1

Number of Operation Needed for the Product : 4

Arrival Probability Distribution
Type : CONSTANT
Parameter 1 : 14
Parameter 2 :

Facility	Process Probability Distribution			Inspection *	
	Type	Parameter		Good	Scrap
		1	2		
11	CONSTANT	8		1	0
12	UNIFORM	3	10	1	0
13	EXPONENTIAL	6.5		.9	.1
23	NORMAL	7	3	.9	.01

* This number should be between Zero and One.

Figure 30. Form B for Product 1

PRODUCT # : 2

Number of Operation Needed for the Product : 3

Arrival Probability Distribution
Type : CONSTANT
Parameter 1 : 12
Parameter 2 :

Facility	Process Probability Distribution			Inspection *	
	Type	Parameter		Good	Scrap
		1	2		
11	CONSTANT	8		1	0
22	CONSTANT	9		.9	.01
23	CONSTANT	7		.85	.11

* This number should be between Zero and One.

Figure 31. Form B for Product 2

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

Figure 32. Output of the Sample Example

GENERAL INFORMATION

DATE : 04-12-87

PROJECT # : IHJ123

SIMULATION REFERENCE : 1

DEPARTMENT : PRODUCTION

OPERATOR NAME : JOE DOE

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 150.00

NUMBER OF FACILITIES : 5

NUMBER OF PRODUCTS : 2

Figure 32. (continued)

PRODUCT # 1

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 14.0000 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 8.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 2 PLACE

PROCESSING DISTRIBUTION : UNIFORM

PARAMETER(S) : 3.0000 10.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 13. IN THE 3 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 6.5000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.100

GOES TO FACILITY 23. IN THE 4 PLACE

PROCESSING DISTRIBUTION : NORMAL

PARAMETER(S) : 7.0000 3.0000

PERCENTAGE OF REWORK : 0.090

PERCENTAGE OF SCRAP : 0.010

Figure 32. (continued)

```
                    PRODUCT # 2

ARRIVAL DISTRIBUTION : CONSTANT
   PARAMETER(S) :     12.0000        0.0000

GOES TO FACILITY 11. IN THE     1 PLACE
                PROCESSING DISTRIBUTION : CONSTANT
                PARAMETER(S) :      8.0000         0.0000
                PERCENTAGE OF REWORK : 0.000
                PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 22. IN THE     2 PLACE
                PROCESSING DISTRIBUTION : CONSTANT
                PARAMETER(S) :      9.0000         0.0000
                PERCENTAGE OF REWORK : 0.040
                PERCENTAGE OF SCRAP : 0.110

GOES TO FACILITY 23. IN THE     3 PLACE
                PROCESSING DISTRIBUTION : CONSTANT
                PARAMETER(S) :      7.0000         0.0000
                PERCENTAGE OF REWORK : 0.000
                PERCENTAGE OF SCRAP : 0.000
```

Figure 32. (continued)

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT IGPSSMP   BY IMED JAMOUSSI

DATE 4/12/1987   RUN NUMBER 1 OF 1

CURRENT TIME 0.1502E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P? DEF TIME	0.3800E+02	0.0000E+00	0.0000E+00	0.3800E+02	0.3800E+02	1
P1 DEF TIME		NO VALUES	RECORDED			
P1 BET DEF		NO VALUES	RECORDED			
P2 DEF TIME	0.3800E+02	0.0000E+00	0.0000E+00	0.3800E+02	0.3800E+02	1
P2 BET DEF		NO VALUES	RECORDED			
P3 DEF TIME		NO VALUES	RECORDED			
P3 BET DEF		NO VALUES	RECORDED			
P4 DEF TIME		NO VALUES	RECORDED			
P4 BET DEF		NO VALUES	RECORDED			
P5 DEF TIME		NO VALUES	RECORDED			
P5 BET DEF		NO VALUES	RECORDED			
T SYSTEM TIME	0.3731E+02	0.6470E+01	0.1734E+00	0.2400E+02	0.4800E+02	13
P1 SYSTEM TIME	0.3733E+02	0.3421E+01	0.9164E-01	0.3237E+02	0.4054E+02	6
P2 SYSTEM TIME	0.3729E+02	0.8600E+01	0.2306E+00	0.2400E+02	0.4800E+02	7
P3 SYSTEM TIME		NO VALUES	RECORDED			
P4 SYSTEM TIME		NO VALUES	RECORDED			
P5 SYSTEM TIME		NO VALUES	RECORDED			
P? BET COMP	0.8000E+01	0.2101E+01	0.2627E+00	0.5733E+01	0.1294E+02	12
P1 BET COMP	0.1538E+02	0.3035E+01	0.1973E+00	0.1273E+02	0.2057E+02	5
P2 BET COMP	0.1600E+02	0.2687E+01	0.1679E+00	0.1337E+02	0.1994E+02	6
P3 BET COMP		NO VALUES	RECORDED			
P4 BET COMP		NO VALUES	RECORDED			
P5 BET COMP		NO VALUES	RECORDED			

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE		AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11	QUEUE	2.2290	1.3821	5	4	15.2183
2	Q12	QUEUE	0.0000	0.0000	0	0	0.0000
3	Q13	QUEUE	0.0000	0.0000	0	0	0.0000
4	Q14	QUEUE	0.0000	0.0000	0	0	0.0000
5	Q15	QUEUE	0.0000	0.0000	0	0	0.0000
6	Q16	QUEUE	0.0000	0.0000	0	0	0.0000
7	Q17	QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21	QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22	QUEUE	0.0067	0.0813	1	0	0.1111
10	Q23	QUEUE	0.1410	0.3480	1	0	1.3234
11	Q24	QUEUE	0.0000	0.0000	0	0	0.0000
12	Q25	QUEUE	0.0000	0.0000	0	0	0.0000

Figure 32. (continued)

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	4.5809	0.9749	6	6	1.4640

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
1	FACILITY 11	1	0.9201	0.2711	1	0.0000	12.0000	138.2010	17
2	FACILITY 12	1	0.2738	0.4459	1	0.0000	28.0000	8.8738	7
3	FACILITY 13	1	0.1248	0.3304	0	0.0000	32.1993	6.1216	7
4	FACILITY 14	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
5	FACILITY 15	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
6	FACILITY 16	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
8	FACILITY 21	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
9	FACILITY 22	1	0.5393	0.4985	0	0.0000	20.0000	18.0000	9
10	FACILITY 23	1	0.7229	0.4476	1	0.0000	29.0000	55.3074	15
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	150.2010	0.0000	0

Figure 32. (continued)

CHAPTER X

SCOPE OF THE MODEL

In the second chapter of this project dealing with the objectives of the study, a few possible applications of the model were discussed to give the feel of the need to use the software. It was mentioned then that a more detailed description of the applications will follow. This section is just an attempt to go through the most important situations that could be studied through this software.

When implementing this study, the ability to simulate different manufacturing environments was considered to be the first and major goal. Simulation of a given manufacturing system could be carried out at different levels and also at different stages of the life of the system. MDMSS could be of use to determine potential problems in the existing system, to evaluate alternatives for future implementations of the system, or just to try to search for ways to ensure better use of the resources existing in the system.

In many manufacturing applications, sometimes, problems are detected in production. Although it is usually easy to discover the existence of these problems, solving them is more challenging: locating and identifying the problem is vital to find a quick solution. In this case, MDMSS could be recommended. In fact, it can, for example, trace back the problem up to the workstation that is delaying jobs. Once this is done, corrective action could be taken.

Scheduling is another application of MDMSS. With this software, the manufacturing engineer could simulate the routing of the products in a manufacturing environment. After the simulation is performed, analysis of the measures of performance, such as the status of the queues and the utilization of the service activities, should give the analyst a good idea on the validity of that routing. If the results were found unsatisfactory, a different schedule is to be developed (using the results from the previous simulation) and fed to the software to enable it to perform another simulation. This process could be repeated until the optimum alternative is found.

The continuous change nowadays in the supply and demand, as well as the increasing competition, force organizations to make changes in their processes in order to guarantee their survival. Making changes is always difficult, especially when these changes can slow down the existing input throughput in the system. For this reason, simulation is found to be probably the best tool to be used. MDMSS is a candidate model to be considered. As an example, introducing a new machine, a new product, or even a new line could be vital to the organization. Instead of carrying out the idea without being sure of the outcome (where huge amounts of money could be on line), the modification could be done on the computer first, then and only then, a decision is recommended.

Even when a process is in control, MDMSS could be used to see if the facilities are used in the best manner. This might sound contradictory, but it can happen: having a stable system might lead people to believe that they are using the system in the best way. Like Imam Ghazali (a philosopher) said once: doubt is a way to reach the truth; manufacturing engineers should ask themselves if the system is used in the most profitable way, and whether it could be improved. Answering these questions is sometimes not easy; MDMSS could simplify them.

To summarize, the developed software, although at its early stage (version 1.0), could be used in many situations. The purpose of this chapter was to familiarize the reader with the idea behind the software, and how and where it could be used. Specific situations were not discussed here because it was thought that they might draw an imaginary limit to the use of MDMSS. It is up to the engineer, more specifically to his(her) creativity, to get the most out of this software.

CHAPTER XI

VALIDATION AND VERIFICATION

Simulation is the science and art of representing a system with a model. In other words, a simulation model could be considered as a laboratory version of a system. Simulation models are used to understand or draw inferences about a system or a situation. Because of the importance of the inferences to the decision makers, the analyst needs to have high confidence in the model built. For this reason, a very critical (sometimes very challenging) step in the simulation is the evaluation of the model. This is done through validation and verification.

Verification of the Model

Kelton and Law defined verification as follows: " It is determining whether a simulation model performs as intended, i.e. debugging the computer program."[23] The following five techniques are generally used to debug the computer program:

1- Write and debug the program in modules.

2- Have other persons walk through the program to avoid any mental rut.

3- Use a trace to follow the logic carried out by the program.

4- Use the model for simple situations where anticipated results could be obtained which would be a basis to compare the results provided by the simulation.

5- If possible display simulation in a graphic mode.

This procedure is to be carried out in the next chapter when the model is going to be subject to verification and testing.

Validation of the Model

According to Kelton and Law, " validation is determining whether a simulation model (as opposed to the computer program) is as accurate representation of the real-world system under study "[23]. Before going into the validation approaches, it is only fair to note that a model is usually validated relative to some decision making criteria. Naylor and Finger developed an approach for validation[23]. This approach lies on the following steps:

1- Develop a model with high validity. Comparing the model to the system under study is the first step to validation, it is done through conversation with experts to understand the system to be validated, observation of the system, and using intuition.

2- Test the assumptions of the model empirically if possible. An important factor in this step is sensitivity analysis. This analysis determines the level of detail at which a subsystem is to be modeled.

3- Verify how representative the simulation output data is. Historical data is most commonly used. Validation of this software can not be generalized. When simulating a system, the analyst should make sure that he(she) has an accurate, or at least a close representation of the real-world system under study.

Steady State Analysis

When simulating a system, a steady state is said to be reached when a "GOOD" estimate of the measures of performance is found. Usually, these measures of performance are defined as limits as the time of the simulation goes to infinity. In practice, this can be very costly, depending on the complexity of the situation. Consequently, a trade off is to be considered between good results and high expenses. When statistics are collected using simulation, the initial conditions are usually a factor. The starting state could be initialized in different ways, probably the most widely used initial state is what is called the "empty and idle environment". Using this means initializing all service facilities to be idle and starting the simulation with no entities waiting in the system. Usually, steady state can be reached using this condition. Contrary to what most people think, steady state is not the lack of variability in the statistics results obtained from the simulation, but rather the consistency in the variability, if such variability should exist. Because of the nature of the initial condition, one needs to be sure that the simulation, or more precisely the simulation results, are independent from the starting conditions. In other words, theoretically, a justified initial condition should not predetermine the simulation outcome or even part of it.

To summarize, verification, validation, and steady state are issues that are more and more being discussed and considered to be possible setbacks for the simulation tools. In this project, verification of the model was performed (and will be discussed in the next chapter), validation though is going to be contingent to the system to be modeled. Finally, the steady state, also a parameter of the system, has to be determined by the analyst.

CHAPTER XII

PROGRAM TESTING

Testing MDMSS, or any other software as a matter of fact, is a challenging task. In fact, the different options offered by the software need to be checked, and various situations need to be anticipated.

To test MDMSS, it was decided that simple systems needed to be used; this way the simulation performed could be followed, and the validity of the software could be determined. The first used example consisted of a simulation of a single product that needs to be processed with two machines. All the relative data and results to this example are shown in Appendix B1. To decide if the simulation was performed the way it was supposed to be, a SLAM trace, shown in Appendix C1, was obtained. A detailed analysis of the trace showed that events were occurring in the proper sequence and that all the relationships in the model were properly observed during the simulation execution. It was therefore concluded that the software was working properly. The next step consisted of comparing the results of the simulation to the anticipated results. The arrival rate used for the product was set to a constant of ten units, the processing times at the first and second machines were also constant, equal respectively to twelve and fourteen units. One should anticipate that only five products could be processed if the duration of the simulation was one hundred units. That was exactly what was shown in the summary report. Only the two machines specified showed any utilization during the simulation. The same conclusion was drawn about the queues. The summary report also showed no defective parts in the system which agreed with the initial specification of the user. Concisely, through this first example, the software showed no sign of inconsistency or problems.

With the already described data, a second simulation was performed. This time the user specified that the statistics for variables based on observations need not to be collected. As Appendix D1 shows, the report clearly confirms that this option works.

Example three was used to check if the inspection option is available. One more time, a trace, shown in Appendix E1, was asked for to check the logic of the simulation. Following the trace as well as checking the results of the summary report shown in Appendix F1, made clear the validity of the inspection option in MDMSS.

In the last three examples, the different capabilities of the software were checked individually. It was anticipated that, mixing all of them in one example, would even convince the reader more about the validity, the flexibility, and the power of this software. For this reason, two products were simulated. The two products shared one machine, different defective percentages were allocated for different machines, and all statistics were required. To solve this problem analytically would have required a lot of time. A trace, shown in Appendix G1, was the best way to follow the simulation. After studying this trace and the results shown in Appendix H1, no doubts remained about the validity of MDMSS.

Schriber Production Shop Example Visited

At this stage of the research, MDMSS has been tested with only examples developed specifically for this purpose. It was anticipated that the credibility of this software would be a lot better if a model is tested with an example already run using a different model. After going through the literature, it was found that the most appropriate example is a case study provided by Schriber[34]: Simulation of a Production Shop. The same example was worked by Pritsker, with a little modification. The statement of the problem is shown in Appendix I1, the solution of Pritsker is in Appendix J1. It should be noted that a few changes had to be made so that the example and the software were compatible. In fact, the example states that the jobs arrive with an exponential interarrival time with a mean of 9.6 minutes; then twenty-four percent of the jobs in the stream are of Type 1, 44 percent are of Type 2, and the rest are of Type 3. In MDMSS the arrivals of jobs have to be independent; which means that before using the model, an approximation of these arrivals had to be made. Appendix K1 shows a simple simulation used to determine these arrivals. As Appendix L1 shows, the new arrival times for the different jobs were approximated to a constant distribution with Type 1 arriving on the average every 34.71 minutes, Type 2 every 23.24 minutes, and Type 3 every 22.66 minutes. One might ask if these are valid approximations? In this specific case the answer is yes because the example is only used to verify the validity of the software developed and the results are not to be used for decision making. Appendix M1 shows the results obtained from running MDMSS with the example of the Production Shop. Before any further analysis of these results, one should realize that the approximation of the arrivals of the jobs could play an important role in the error that could be accumulated leading to differences between these results and Pritsker results. In fact, in comparing the two sets of reports, some obvious similarities as well as some differences are apparent: the utilization of the machines are considerably close, the same conclusion holds with the number of jobs served during simulation, and some of the times spent in the system. On the other hand, some of the queue lengths are quite different. This difference is due primarily to the difference in the distribution of the arrivals of the jobs. Also it might be of importance to note that statistics were collected differently in this run and in Pritsker's model which could contribute to the differences in the results. To summarize, the results obtained with the MDMSS model, while different from those obtained by Pritsker's model, were sufficiently close that they added support to the validity of the software.

Claiming that the software works is based on good and valid tests; nevertheless, everything is possible and bugs may exist.

CHAPTER XIII

RECOMMENDATIONS FOR FUTURE IMPLEMENTATIONS

Developing new software of the magnitude of MDMSS requires a lot of research, insight, and especially time. A working core was the goal of this project. In fact, it was clear from the start that a complete model can never be achieved at this stage. The primary reason was related to time constraint to submit this project. But, all along the development of MDMSS, serious considerations were anticipated to make any future work and implementation of the software possible without any major changes; that is why the model is well documented both externally and internally. An easy structure of the programs has also been followed so that others interested in taking the software one step further do not find major problems to interface their work.

In the first chapter, it was mentioned that MDMSS is a software that, hopefully, is a candidate to be part of a bigger picture: one of the many models that form a very powerful simulator. Although this simulator is not now available, serious research is being done in the School of Industrial Engineering and Management at Oklahoma State University under the direction and supervision of Dr. Joe H. Mize to put this concept to application. This research is referred to as the computer integrated manufacturing system (CIMS) which is thought to be the next breakthrough in the sector of manufacturing. The order in which the recommendations are to be discussed later does not imply any priority form.

MDMSS does not give the user the chance to choose a seed for generation of random numbers needed to obtain some distributions. Although this may not appear to be a major setback, it could be of great importance when multiple independent simulations are needed. As it is now, if the program is run at any time, the program will select the seed. It happens that for independent runs, if the seed chosen is the same, the same exact results will be given.

When simulating a mix of products, it is assumed that the mix is fixed; which means that MDMSS has a static status in this first version. Although, this is true in some manufacturing environments, in others this restriction would misrepresent the actual system. To solve this problem, MDMSS should be able to simulate both types of systems. A new module needs to be added. This module has to give the user the opportunity to use either, a static or dynamic simulation. The restriction of having the same product mix throughout the simulation should disappear. This leads to talk about the initial conditions of the simulation. The software is designed such that the initial conditions are set to "empty idle" which means that when the simulation starts, there are no entities waiting for service and all facilities are idle. In most cases, this assumption is valid, but to have a real general purpose simulation model, such a limitation should not exist: the analyst should have the flexibility to decide on the initial conditions.

Although the INPUT program is menu driven, user friendly, self checking, and has free format input, it is still missing an important feature: the ability to access the storage file for a particular data item and change it. This feature was not overlooked, rather, it was judged not to be of primary importance at this stage. Random access to the file could be guaranteed either with the same INPUT program or with a new program written to read the file and give the user the chance to alter any value or values needed before the simulation is performed one more time. This option could be of great importance, especially if the amount of the data is large.

Multiple simulations are usually required to find the optimum solution to the problem. Iteration mode can be the best way to run the simulation in this case. For different runs, obviously, the data needs to be changed. These changes could be at two different levels: the first and traditional way to do it is simply through having the analyst input different sets of data before the simulation is performed. The second way is to develop an expert system module that, using some predetermined rules, accesses the simulation results and according to those results alters the data, or part of it and submits the simulation one more time. This procedure is repeated until good results are obtained. To interface either procedure with MDMSS, a CLIST program would need to be developed. This program has only commands that can be processed from the operating system of the computer used (it is not subject to FORTRAN or any other language).

Since the concept of expert systems and knowledge based system has been mentioned, it seems appropriate to describe how MDMSS could benefit from these concepts to be part of the bigger simulator to be implemented in the CIMS project. Any computer integrated manufacturing system has to have the main property of interfacing its different modules so that when one change is made, the entire system updates itself. Knowledge and expert systems seem to be the logical connector of the simulator: first a simulation is performed at one module when an optimum solution is found in that module then knowledge and expert systems should be able to update the data respective to their own module and a simulation is performed. Feedback to other modules is then passed through these connectors (expert and knowledge systems). What might be a challenging project is to try to anticipate and decide on the expert and knowledge base systems that are to connect MDMSS with the remaining CIMS simulator.

In this version of MDMSS, the queues in front of the service activities are assigned to have an infinite capacity. This assumption could be misleading in many instances. In fact the bigger the quantity of work in process is, the more the carrying costs are. The user should somehow have the option to decide on the capacity of those queues. A module could be added to MDMSS to allow this option. Through the same model or subprogram, the user should be able to set some rules that can perform logical decisions when a particular queue reaches its maximum capacity. This decision could vary from blocking the activity to making the next coming products to take different routes.

In real life manufacturing systems, when a product is serviced by a certain facility and before it reaches the next one, some travel time is spent. Currently, this time is not included in the model. The INPUT program should be modified to include this. Another way to do the same thing is to interface MDMSS with a travel time matrix which identifies the facility the entity is leaving and the one it is going to. The matrix determines then the travel time and feeds it to the simulation model.

Assigning priorities is another area that is subject to improvement. For simplification purposes, the first come first served rule was used. This was justifiable for two reasons; the first one is the fact that this rule stands sometimes even in the real systems, the second is that assumptions had to be made to develop the model. Now that the core is working, adding a new option such as this is encouraged.

Other options that should be added to the model for greater flexibility are including random facility breakdowns, rescheduling of rework to different machines or even areas, and provide batch processing and assembly. All these options were not overlooked. In fact, it was believed that it is wiser to develop a basic model that works properly and then enhance it later, rather than have a model that has many options and can not even be verified because of its complexity. Facilities breakdowns is a legitimate phenomenon to be considered. The modeler should include both scheduled and unscheduled maintenance and repair in the system. Now, after inspection, the entity simulated takes one of the three next events: either it passes inspection and is scheduled to a new operation, or it is determined to be scrap and is thrown out of the system, or finally it was found to be defective and judged to be reworkable. In this last case, currently, the product is routed back to the machine it just left which is a fair assumption but not the only possible one. That is why, future versions of MDMSS should have the user decide where the defects need to go. Last but not least, assembly and consequently batch processing should be developed. The current version of the software could be easily changed to accommodate this option.

As was mentioned in the early chapters, simulation of a system could be performed at different levels (four of them). Another improvement of the software is have the INPUT program read data for all the levels together, then, an expert system should be developed as the driver of the simulation: the expert system submits the simulation at one level and then according to the results obtained, the purpose of the simulation, and some predetermined logic, the expert system should chose the next level to be simulated if the need to do so exists.

Finally, at a later stage when the model is found to be as complete as it could be, it would be wise to transfer this package to diskettes so that it could be used at the personal computer level. One might argue that a project of this magnitude would be hard to run on a personal computer, but with the new technology these "little" computers are getting more powerful as well as faster everyday. If ever this is done, the package can be interfaced with the graphic portion of SLAM II: TESS.

The recommendations given in this section, although numerous, should not be considered as weak points of the existing version. Just getting the software to the point it is now, was very challenging. The reason is that the concept had to be created from zero, with no previous experiences or examples as guidelines. The model had also to be designed in a way that it can accommodate new implementation easily.

CHAPTER XIV

SUMMARY AND CONCLUSION

The objectives of this project were set in chapter II: a Menu-Driven Manufacturing System Simulator had to be developed. The tool created, MDMSS, satisfies all the objectives stated then. Although MDMSS, in this first version, has limited options, it shows that a comprehensive simulation model is feasible. The model designed and created is a tool developed primarily for the use of the manufacturing engineers. Using MDMSS could solve problems or provide educated inferences for decision makers. One of the advantages of the software is its ability to represent a system without building the actual system, disturbing the system, or destroying it. All this can be done without writing any programs. Through running two programs the user causes the software to first picture the system and then perform the simulation. The job of the engineer is simply to analyze the data and make any recommendations to the management or to his(her) supervisors.

It should be noted though, that the actual version of the program does not offer great flexibility in modeling, but this should not be considered as a weakness of the software. In fact, in the process of designing and creating this model, it was anticipated that by the end of this project only a core of the software is going to be available for use. This goal was achieved, and MDMSS is ready to be used. Later on, other projects are encouraged to complete what has been started and achieved in this first version of this simulation software: MDMSS.

BIBLIOGRAPHY

1- Arumugam, V. "Priority Seqencing in a Real World Job Shop." Simulation, Vol. 45, No. 4 (October, 1985), pp. 179- 185.

2- Asfahl, C. R. Robots and Manufacturing Automation. New York: John Wiley & Sons, 1985.

3- Banks, J. and J. S. Carson II. "Process-Interaction Simulation Languages." Simulation, Vol. 44, No. 5 (May, 1985), pp. 225-234.

4- Barrett R. T. and S. Barman. "A SLAM II Simulation Study of a Simplified Flow Shop." Simulation, Vol. 47, No. 5 (Nonember, 1986), pp. 181-188.

5- Bedworth, D. D. and J. E. Bailey. Integrated Production Control Systems. New York: John Wiley & Sons, 1982.

6- Benjamin, S. B. and W. J. Fabrycky. Systems Engineering and Analysis. New Jersey: Prentice-Hall, Inc., 1981.

7- Bryan, O. F. and M. C. Natrella. "Testing Large-Scale Simulations." Byte, Vol. 10 (October, 1985), pp. 183-190.

8- Burch, J. G. Jr., F. R. Strater, and G. Grudnitski. Information Systems: Theory and Practice. New York: John Wiley & Sons, Inc., 1983.

9- Buzacott, J. A. "Modeling Manufacturing Systems." Robotics and Computer- Integrated Manufacturing, Vol. 2, No. 1 (1985), pp. 25-32.

10- "Catalog of Simulation Software." Simulation, Vol. 47, No. 4 (October, 1986), pp. 152-165.

11- Cheng, T. C. E. "Simulation of Flexible Manufacturing Systems." Simulation, Vol. 45, No. 6 (December, 1985), pp. 299-302.

12- "Computer Techniques Isolate Production Snags." Production, February, 1977.

13- Degarmo, P. E., J. T. Black, and R. A. Kohser. Materials and Processes in Manufacturing. New York: Macmillan Publishing Company, 1984.

14- Demmel, J. G. " CHAP-M: Computer Hierarchy and Analysis Program for Manufacturing." (Unpub. M.S. Report, Oklahoma State University, 1985.)

15- Groover, M. P., M. Weiss, R. N. Nagel, and N. G. Odrey. Industrial Robotics: Technology, Programming, and Applications. New York: McGraw-Hill Book Company, 1986.

16- Groover, M. P. Automation, Production Systems, and Computer-Aided Manufacturing. New Jersey: Prentice-Hall, Inc., 1980.

17- Hanifin, L. E. "Increased Transfer Line Productivity Utilizing Systems Simulation." (D. Engeneering dissertation, University of Detroit, 1975.)

18- Hanifin, L. E., S. G. Liberty, and K. Taraman. "Improved Transfer Line Efficiency Utilizing Systems Simulation." Technical Paper MR 75-169, Society of Manufacturing Engineers, Deaborn, Michigan, 1975.

19- Houston, T. R. "Why Models Go Wrong." Byte, Vol. 10 (October, 1985), pp. 151-164.

20- Hughes, C. H., C. P. Pfleeger, and L. L. Rose. Advanced Programming Techniques: A Second Course in Programming Using Fortran. New York: John Wiley & Sons, 1978.

21- "Innovative Computer Software Simulates Process - Productivity Improvement Saves Millions." Simulation, Vol. 44, No. 5 (May, 1985), p. 258.

22- Krick, E. V. An Introduction to Engineering: Methods, Concepts, and Issues. New York: John Wiley & Sons, 1976.

23- Law, A. M., and W. D. Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill Book Company, 1982.

24- Luker, P. A. and J. Stephenson. "Fourth UKSC Conference on Computer Simulation." Simulation, Vol. 44, No.2 (February, 1985), pp. 96-100.

25- Merchant, M. J. FORTRAN 77: Language and Style. California: Wadsworth Publishing Company, 1981.

26- Mittra, S. S. "Discrete System Simulation Concepts." Simulation, Vol. 43, No. 3 (September, 1984), pp. 142-144.

27- Mize, J. H. Production System Simulator (PROSIM V): A User's Manual. New Jersey: Prentice-Hall, Inc., 1971.

28- Phillips, D. T. and R. F. Slovick. " A GERTS III Q Application to a Production Line." Proceedings, 1974 Spring Annual Conference of AIIE, May, 1974.

29- Phillips, D. T. and A. A. B. Pritsker. "GERT Network Analysis of Complex Production Systems." International Journal of Production Research, Vol. 13, No. 3 (1975), pp. 223-237.

30- Pritsker, A. A. B. and C. D. Pegden. Introduction to Simulation and SLAM. New York: John Wiley & Sons, 1979.

31- Pritsker, A. A. B. Introduction to Simulation and SLAM II. New York: John Wiley & Sons, 1984.

32- Pritsker, A. A. B. Introduction to Simulation and SLAM II. New York: John Wiley & Sons, 1986.

33- Schallert, W. F. and C. R. Clark. Programming in FORTRAN. Massachusetts: Addison-Wesley Publishing Company, 1979.

34- Schriber, T. J. Simulation Using GPSS. New York: John Wiley & Sons, 1974.

35- "Software Package Previews Automated Factory Designs." Design News, Vol. 41 (June 3, 1985), p. 25.

36- Strenski, E. and M. Manfred. The research Paper Workbook. New York: Longman, Inc., 1981.

37- Tom, A. A. "Simulation Nets, a Simulation Modeling and Validation Tool." Simulation, Vol. 45, No. 2 (August, 1985), pp. 71-75.

38- Turner, W. C., J. H. Mize, and K. E. Case. Introduction to Industrial and Systems Engineering. New Jersey: Prentice-Hall, Inc., 1978.

39- Vollmann, T. E., W. L. Berry, and D. C. Whybark. Manufacturing Planning and Control Systems. Illinois: Richard D. Irwin, Inc., 1984.

APPENDICES

APPENDIX A

SOFTWARE LIST

SOFTWARE LIST

The following list is of the simulation languages comercialized as of October 1986 and their brief descriptions. These languages are available for minicomputers and mainframes[35]:

AC-2, SST, PLATO : Programs for the stepwise checkrating of heat exchangers. Education.

ACSL : Models and analyzes continuous systems.

ADSIM : A continuous systems simulation language including compiler, utilities, interactive environment and large libraries.

ASPEN PLUS : Simulates chemical process flow-sheets for proposed or operating plants.

AutoCode : Generates machine independent, real-time source code from simulation block diagrams.

BATCHFRAC : Batch distillation simulation program that solves unsteady state heat and material balance equations.

BEST-NETWORK : Network simulation tool for throughput and connectivity analysis/design. No programming necessary.

BORIS : Building-blocks oriented interactive modeling and simulation system.

BWR MODELS : Full scope BWR plant system dynamic models, all written in FORTRAN 77, as is or custom fit to the need.

CAMP : It derives system differential equations from bond graphs, block diagrams and their combination and delivers them to ACSL for continuous simulation.

DARE-INTERACTIVE : Interactive simulation of continuous systems with enhanced run-time experimentation facilities and interactive color graphics.

DESCTOP : Direct-executing simulation, 120 state variables, screen editor, about FORTRAN speed, Tektronix color, FFT.

DESIRE V 3.2 : Direct-executing simulation, 40 state variables, screen editor.

DSNP : A simulation language for analyzing thermal hydraulic transients.

DSS/2 : A transportable FORTRAN 77 code for the numerical integration of systems of ordinary and partial differential equations.

EASY5 : A software package which simulates dynamic response and performs control system analysis.

ENPORT-6 : Simulation of nonlinear dynamic systems modeled with bond graphs and block diagrams.

ESL : An advanced CSSL which provides both translator and interpreter execution.

GEMS-II : Network-based discrete-event simulation language. Designed for manufacturing systems.

GPSS : Company offers several subsets of GPSS for various equipment and purposes.

GPSS/H : State-of-the-art language based on GPSS. Features high-speed execution and interactive debugging.

GRD1 : A transportable FORTRAN 77 code for the numerical integration of nonstiff/stiff ODEs and one-dimensional PDEs on an adaptive grid.

HEXTRAN : An essential tool for engineers involved in heat recovery and energy optimization for the chemical processing industry.

HYSIM : A comprehensive and complete process flowsheet simulator for gas processing or oil refining.

Inter-SIM : Discrete-event simulation package with animation facilities.

ISIS 80 : A continuous system simulation program and language.

JADE : Software development environment which supports the development, prototyping and simulation of distributed systems.

LA/ACTION : Advance decision support system.

MANIP : Interactive modeling and simulation system for thermal engineers.

MAP/1 : A modeling and analysis program for batch manufacturing.

MAST : Simulation language for the study of integrated manufacturing.

MDOF : Missile system engagement simulation with multiple degrees-of-freedom.

MIDGET : Program generator for special-purpose operating systems (simulation language environment).

MIRANIM : Director-oriented and extensible three-dimensional computer animation system.

OPTIK : A suite of visual, interactive modeling decision-support tools, including discrete-event simulation and data base facilities.

PAWS : Simulation language for performance modeling of computer, communicating, and manufacturing systens, and other similar systems.

PIPEPHASE : Simulates the steady state flow of single and multiphase fluids in pipelines and piping networks.

PROCESS : Performs rigorous mass and energy balances. Unit operation modules are incorporated for simulation of process units.

PRO-MATLAB : Performs matrix analysis, control design and analysis, digital signal processing, systems identification, and engineering graphics.

PROMISE : Modeling of communicating parallel processes.

SANDYS : For time simulation of dynamic systems.

SCoP : Interactive simulation system based on C language. Menu driven; graphic output; includes numeric library and optimizer.

SIGMUS : Simulation software generator for multiprocessor systems, based on their structure and their instruction set.

SIMAN : General-purpose simulation language with special features for modeling manufacturing systems.

SIMULA : General-purpose programming language embodying the concepts of object-oriented programming. **SimuSolv** : Simulates behaviour, optimizers performance and estimates phenomenological parameters.

SLAM II : Comprehensive simulation language permitting discrete event, continuous, and network modeling.

SYSMOD : A general-purpose discrete/combined/ continuous simulation language.

SYSTEM BUILD : Graphical, block diagram tool for interactive modeling and simulation of dynamic systems.

TC-PROLOG : PROLOG based combined discrete/continuous simulation and poblem-solving system.

TESS : An integrated interactive simulation support system with relational database management system and graphics capabilities.

TPS : An interactive graphics real-time acquisition, processing analysis system.

Workstation : Software system for simulation of point-to-point digital communications systems.

See Why : Animated system simulation model, showing movement of entities through work/service stations, queues, etc.

Simfactory : It simplifies factory design and production analysis. Simfactory simulates service activities, it provides the user with reports and animated picture of the facility (i.e. factory) at work.

Micro Saint : This is a new package that allows to build simulation models by just responding to interactive menus. No computer programs have to be written.

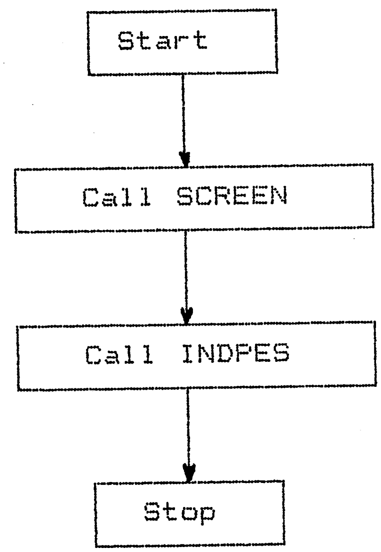
PROSIM : This simulator is constructed such that it can simulate a wide variety of production environments. It is mainly used as a teaching aid.

Xcell : It is a Factory Modeling System. This software package is used to evaluate the design of a factory. A model is build and the measures of performance of the factory are collected. Computer graphics are employed both in the construction of the model and in the display of the results.

SIMSCRIPT : This is another simulation language. It has been used for different kind of projects (i.e. govermental, industrial), it showed good performance.

APPENDIX B

FLOWCHART AND LISTING OF MAIN IN INPUT


Start
Call SCREEN
Call INDPES
Stop

```
AY 1985)            VS FORTRAN          DATE: MAR 08, 1987     TIME: 14:56:46

ONS (EXECUTE): NODECK,NOLIST,OPT(O)

ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NOT
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
       OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARLE

  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C==================================================================C
 C::::::                                                      ::::::C
 C......    PROJECT :  THESIS FOR MASTER OF SCIENCE DEGREE IN    ......C
 C......               IN INDUSTRIAL ENGINEERING                 ......C
 C......    CANDIDATE : IMED JAMOUSSI                           ......C
 C......    ADVISOR : DR. JOE MIZE                              ......C
 C......    LANGUAGE : FORTRAN & SLAM                           ......C
 C......    SYSTEM : IBM MAIN FRAME                             ......C
 C::::::                                                      ::::::C
 C==================================================================C
 C******************************************************************C
 C::::::                                                       ::::::C
 C......      LIST OF VARIABLES :                              ......C
 C......      ================                                 ......C
 C......       LEVEL : LEVEL OF SIMULATION (AREA,CELL...)      ......C
 C......       NPROD : NUMBER OF PRODUCTS IN THE SYSTEM        ......C
 C......       MACH : NUMBER OF STATIONS                       ......C
 C......       M(5)   : ARRAY TO KEEP TRACK OF PRODUCTS/MACHINES  ......C
 C......       NMACHK(5) : NUMBER OF MACHINES NEEDED PER PRODUCT  ......C
 C......       ENTITY(5,35,2) : 5 MACHINES, 35 STATIONS        ......C
 C......       ENTITY(K,KK,1): TYPE OF PROCESS FUNCTION        ......C
 C......       ENTITY(K,KK,2): FLAG TO KEEP TRACK OF THE ORDER   ......C
 C......       PARA(5,35,3) : 5 MACHINES, 35 STATIONS.         ......C
 C......       PARA(K,KK,1): PARAMETER 1 OF PROCESSING FUNCTION  ......C
 C......       PARA(K,KK,2): PARAMETER 2 OF PROCESSING FUNCTION  ......C
 C......       PARA(K,KK,3): FLAG FOR PROCESSING FUNCTION TYPE   ......C
 C......       ARVAL(5,3) : ARRIVAL RATES PER MACHINE          ......C
 C......       ARVAL(K,1) : TYPE OF FUNCTION                   ......C
 C......       ARVAL(K,2), ARVAL(K,3): FUNCTION PARAMETERS 1 AND 2......C
 C::::::                                                       ::::::C
 C*******************************************************************C
 C===================================================================C
 C::::::                                                       ::::::C
 C......    MAIN PROGRAM :                                     ......C
 C......    ------------                                       ......C
 C......                THIS PROGRAM IS TO   READ DATA AND     ......C
 C......    STORE IT IN A FILE                                 ......C
 C::::::                                                       ::::::C
 C===================================================================C
        CALL SCREEN
       CALL INPDES
       STOP
       END

 SOURCE STATEMENTS = 4, PROGRAM SIZE = 484 BYTES, PROGRAM NAME = MAIN      PAGE:

  NO DIAGNOSTICS GENERATED.

OF COMPILATION 1 ******
```

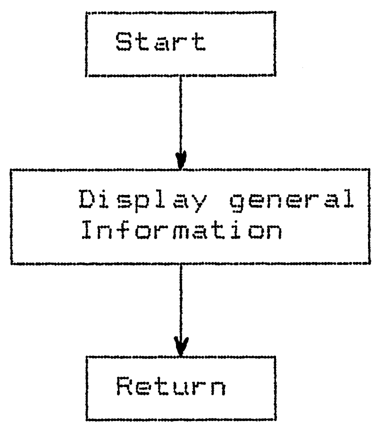
APPENDIX C

LISTING OF SUBROUTINE CLSC

```
MAY 1985)              VS FORTRAN          DATE: MAR 08, 1987     TIME: 14:56:48

'FECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM    OBJECT FIXED   NOTEST N'
        NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
          OPT(O) LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)       CHAR

  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

  C===================================================================C
  C::::::                                                       ::::::C
  C......     SUBROUTINE CLSC :                                 ......C
  C......     ---------------                                   ......C
  C......                       THIS SUBROUTINE IS TO CLEAR THE ......C
  C......     SCREEN.                                           ......C
  C::::::                                                       ::::::C
  C===================================================================C
        SUBROUTINE CLSC
  C
  C=====>> LOOP TO CLEAR SCREEN.
  C
        DO 10 IJK=1,25
              WRITE(6,5)
   5          FORMAT(' ')
   10   CONTINUE
        RETURN
        END

  SOURCE STATEMENTS = 7, PROGRAM SIZE = 588 BYTES, PROGRAM NAME = CLSC       PAGE:

   NO DIAGNOSTICS GENERATED.
```

APPENDIX D

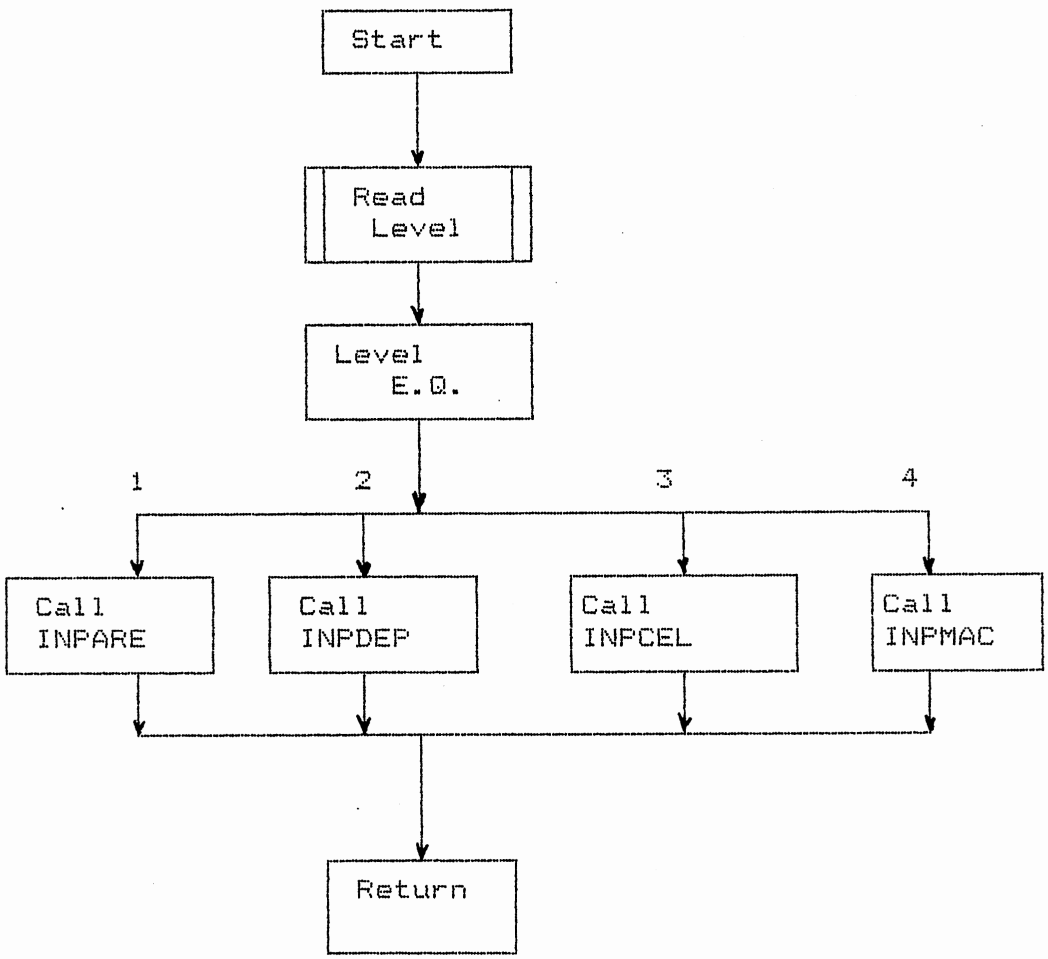
FLOWCHART AND LISTING OF SUBROUTINE SCREEN


Start
Display general
Information
Return

```
ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED   NOTEST N
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM    IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)       CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                           ::::::C
 C......     SUBROUTINE SCREEN :                                   ......C
 C......     -----------------                                     ......C
 C......                        SUBROUTINE TO SHOW FIRST SCREEN    ......C
 C::::::                                                           ::::::C
 C======================================================================C
       SUBROUTINE SCREEN
        CALL CLSC
       WRITE(6,10)
  10   FORMAT(///,38X,'MDMSS')
       WRITE(6,20)
  20   FORMAT(38X,'-----')
       WRITE(6,30)
  30   FORMAT(///,37X,'VERSION 1.0')
       WRITE(6,40)
  40   FORMAT(///,36X,'DEVELOPED BY :')
       WRITE(6,50)
  50   FORMAT(/,35X,'IMED JAMOUSSI')
       WRITE(6,60)
  60   FORMAT(//,27X,'PLEASE REPORT PROBLEMS TO :')
       WRITE(6,70)
  70   FORMAT(/,35X,'DR. JOE H. MIZE')
       DO 100 LJK=1,1200000
           DUM=DUM+1
 100   CONTINUE
       CALL CLSC
       RETURN
       END

 SOURCE STATEMENTS = 22, PROGRAM SIZE = 948 BYTES, PROGRAM NAME = SCREEN     PAGE:

  NO DIAGNOSTICS GENERATED.
```

APPENDIX E

FLOWCHART AND LISTING OF SUBROUTINE INPDES


Start
Read Level
Level E.Q.
1
2
3
4
Call INPARE
Call INPDEP
Call INPCEL
Call INPMAC
Return

```
'ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NC
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(0) LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)      CHARL

  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

  C======================================================================C
  C::::::                                                          ::::::C
  C......        SUBROUTINE INPDES :                               ......C
  C......        -----------------                                 ......C
  C......                           THIS SUBROUTINE GIVES THE USER ......C
  C......        THE CHANCE TO DECIDE ON THE LEVEL OF OPERATIPON.  ......C
  C::::::                                                          ::::::C
  C======================================================================C
        SUBROUTINE INPDES
  C
  C=====>> INITIALIZATION OF VARIABLES.
  C
        COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
       $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
        DO 1 K=1,5
          DO 1 KK=1,35
            DO 1 KKK=1,3
   1    PARA(K,KK,KKK)=0.
        DO 2 K=1,5
             NMACHK(K)=0
            DO 2 KK=1,3
   2    ARVAL(K,KK)=0.
        DO 3 K=1,5
            DO 3 KK=1,35
                DO 3 KKK=1,2
   3    ENTITY(K,KK,KKK)=0.
        DO 4 K=1,5
            DO 4 KK=1,35
                 DEF(K,KK,1)=1
    4            DEF(K,KK,2)=0.
  C=====>> INPUT FROM THE USER
  C
         CALL INFO
    5   CALL CLSC
        WRITE(6,10)
   10   FORMAT(' THIS PROGRAM IS USED TO INPUT DATA TO BE USED FOR')
        WRITE(6,11)
   11   FORMAT(' SIMULATION OF A MANUFACTURING SYSTEM.')
        WRITE(6,12)
   12   FORMAT(' THE SIMULATION CAN BE PERFORMED AT FOUR DIFFERENT')
        WRITE(6,13)
   13   FORMAT(' LEVELS.  PLEASE REFER TO THE MENU BELOW AND SELECT.')
        WRITE(6,14)
   14   FORMAT(//,24X,' MENU ')
        WRITE(6,15)
    15   FORMAT(24X,' ---- ')
        WRITE(6,16)
   16   FORMAT(7X,'1- AREA SIMULATION',6X,'2- DEPARTMENT SIMULATION')
        WRITE(6,17)
   17   FORMAT(7X,'3- CELL SIMULATION',6X,'4- MACHINE SIMULATION',//)
        READ(5,*,ERR=5) LEVEL
   21   FORMAT(1X,I1)
```

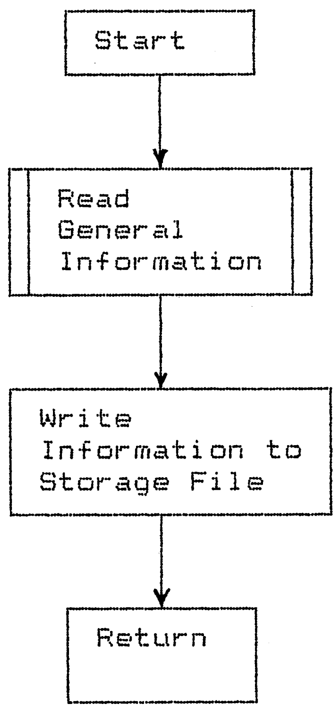
```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8

      IF(LEVEL.LT.1.OR.LEVEL.GT.4) GO TO 5
      WRITE(3,21) LEVEL
 24   WRITE(6,25)
 25   FORMAT(' ENTER THE DURATION OF THE SIMULATION ')
      READ(5,*,ERR=24) TIMED
      WRITE(3,27) TIMED
 27   FORMAT(1X,F12.2)
C
C=====>> DECISION
C
      GO TO(30,40,50,60),LEVEL
 30   CALL INPARE
      CALL STATSA
      RETURN
 40   CALL INPDEP
      CALL STATSD
      RETURN
 50   CALL INPCEL
      CALL STATSC
      RETURN
 60   CALL INPMAC
      CALL STATSM
      RETURN
      END
```

SOURCE STATEMENTS = 59, PROGRAM SIZE = 2668 BYTES, PROGRAM NAME = INPDES   PAGE

NO DIAGNOSTICS GENERATED.

APPENDIX F

FLOWCHART AND LISTING OF SUBROUTINE INFO



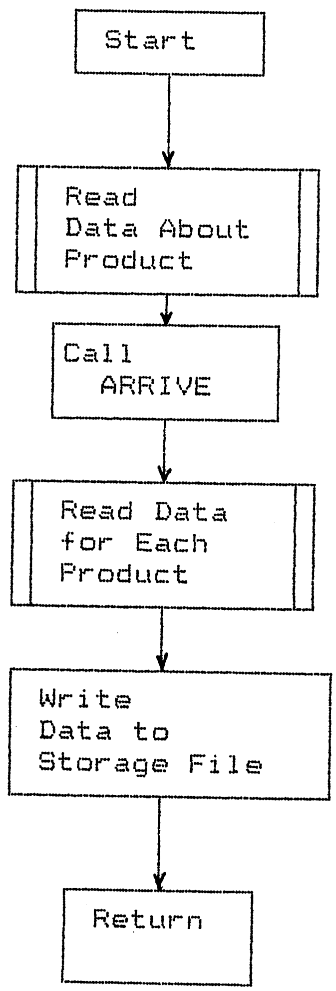
Start
Read General Information
Write Information to Storage File
Return

```
:CT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)       CHAR

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=================================================================C
C::::::                                                     ::::::C
C......    SUBROUTINE INFO     :                            ......C
C......    -----------------                                ......C
C......                     THIS SUBROUTINE IS USED TO INPUT ......C
C......    SOME GENERAL INFORMATION.                        ......C
C::::::                                                     ::::::C
C=================================================================C
      SUBROUTINE INFO
C
C=====>> INITIALIZATION OF VARIABLES.
C
         CHARACTER*8 DATE
      CHARACTER PROJEC*12,DEPART*20,NAME*20,REFE*5
      PROJEC='            '
      DEPART='                    '
      NAME  ='                    '
      REFE='     '
      CALL CLSC
      WRITE(6,10)
 10   FORMAT(' YOU ARE UP TO USE MDMSS. FIRST, PLEASE ANSWER THE')
      WRITE(6,11)
 11   FORMAT(' FOLLOWING QUESTIONS NEEDED FOR RECORD KEEPING PURPOSE')
      DO 13 LJP=1,900000
 13   CONTINUE
C     CALL CLSC
 15   WRITE(6,20)
 20   FORMAT(' PLEASE ENTER THE DATE FOLOWING THE FORMAT : MO-DY-YR')
          READ(5,21) DATE
 21       FORMAT(A8)
      CALL CLSC
 30   WRITE(6,40)
 40   FORMAT(' PLEASE ENTER THE PROJECT NUMBER.MAXIMUM 12 CHARACTERS')
      READ(5,45) PROJEC
 45   FORMAT(A12)
      CALL CLSC
 50   WRITE(6,60)
 60   FORMAT(' PLEASE ENTER THE DEPARTMENT THE SIMULATION IS DONE FOR')
      WRITE(6,61)
 61   FORMAT(' MAXIMUM OF 20 CHARACTERS.')
      READ(5,65) DEPART
 65   FORMAT(A20)
      CALL CLSC
 70   WRITE(6,80)
 80   FORMAT(' PLEASE ENTER THE OPERATOR NAME. MAXIMUM 20 CHARACTERS')
      READ(5,85) NAME
 85   FORMAT(A20)
      CALL CLSC
 90   WRITE(6,100)
 100  FORMAT(' PLEASE ENTER THE SIMULATION REFERENCE.(5 CHARACTERS)')
      READ(5,110) REFE
 110  FORMAT(A5)
```

```
AY 1985)            VS FORTRAN           DATE: APR 15, 1987     TIME: 07:07:27       N

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

      WRITE(3,120) DATE
  120 FORMAT(1X,A8)
      WRITE(3,130) PROJEC
  130 FORMAT(1X,A12)
      WRITE(3,140) DEPART
  140 FORMAT(1X,A20)
      WRITE(3,150) NAME
  150 FORMAT(1X,A20)
      WRITE(3,160) REFE
  160 FORMAT(1X,A5)
      RETURN
      END

SOURCE STATEMENTS = 52, PROGRAM SIZE = 2112 BYTES, PROGRAM NAME = INFO       PAGE

 NO DIAGNOSTICS GENERATED.
```

APPENDIX G

FLOWCHART AND LISTING OF SUBROUTINE INPARE


Start
Read
Data About
Product
Call
ARRIVE
Read Data
for Each
Product
Write
Data to
Storage File
Return

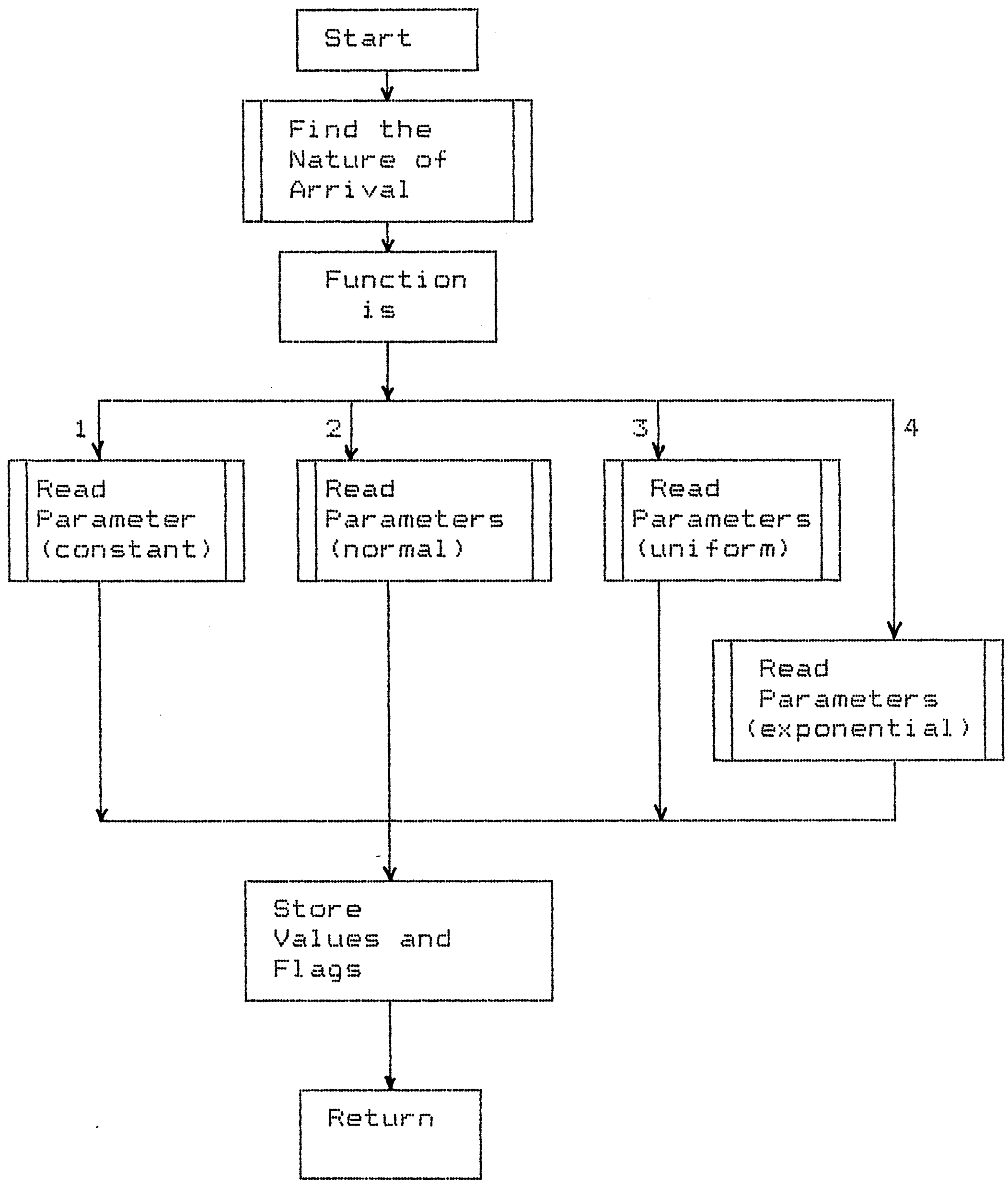
```
:ECT:   NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N(
        NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
          OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARI

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C==================================================================C
 C::::::                                                      ::::::C
 C......    SUBROUTINE INPARE :                              ......C
 C......    -----------------                                ......C
 C......                      THIS SUBROUTINE IS TO READ THE  ......C
 C......    THE INPUT OF THE USER WHEN SIMULATION OF AREAS IS  ......C
 C......    NEEDED.                                          ......C
 C::::::                                                      ::::::C
 C==================================================================C
       SUBROUTINE INPARE
 C
 C=====>> INITIALIZATION OF VARIABLES
 C
       COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
      $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
 C
 C=====>> INPUTS STARTS
 C
   5   CALL CLSC
       WRITE(6,10)
  10   FORMAT(' YOU ARE AT THE AREA LEVEL SIMULATION (LEVEL 1)')
       WRITE(6,11)
  11   FORMAT(//)
       WRITE(6,30)
  30   FORMAT(' HOW MANY PRODUCTS YOU WANT TO SIMULATE (MAXIMUM 5) ?')
       READ(5,*,ERR=5) NPROD
       IF(NPROD.LT.1.OR.NPROD.GT.5) GO TO 5
  35   WRITE(6,50)
  50   FORMAT(' HOW MANY AREAS IN THE LAYOUT (MAXIMUM 35) ?')
       READ(5,*,ERR=35) MACH
       IF(MACH.LT.0.OR.MACH.GT.35) GO TO 35
       DO 90 K=1,NPROD
  55      WRITE(6,70)K
  70      FORMAT(' HOW MANY AREAS ARE USED FOR PRODUCT ?  ',I2)
          READ(5,*,ERR=55) NMACHK(K)
  90   CONTINUE
       WRITE(3,91) NPROD
  91   FORMAT(1X,I1)
       WRITE(3,92) MACH
  92   FORMAT(1X,I2)
 C
 C=====>> FIND THE ARRIVAL RATES.
 C
       CALL ARRIVE
 C
 C=====>> MORE INFORMATION TO INPUT
 C
       CALL CLSC
       DO 130 K=1,NPROD
              DO 120 KK=1,NMACHK(K)
                     CALL CLSC
  95              WRITE(6,100)K,KK
```

```
 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

  100              FORMAT(' WHICH AREA DOES PRODUCT ',I2,' GO TO IN THE ',
     $                   I2,' PLACE')
                   READ(5,*,ERR=95) RAK
      IF(RAK.LT.11.OR.RAK.GT.17.AND.RAK.LT.21.OR.RAK.GT.27.AND.RAK.LT.
     $31.OR.RAK.GT.37.AND.RAK.LT.41.OR.RAK.GT.47.AND.RAK.LT.51.OR.
     $RAK.GT.57) GO TO 95
                   ENTITY(K,KK,2)=RAK
  120           CONTINUE
  130   CONTINUE
 C
 C=====>> INPUTS OF THE PROCESSING TIMES.
 C
        DO 170 K=1,NPROD
               DO 160 KK=1,NMACHK(K)
                       CALL CLSC
  135              WRITE(6,140)KK,K
  140              FORMAT(' WHAT IS THE PROCESSING TIME FOR AREA ',
     $                    I2,' WITH PRODUCT ',I2,/,' ENTER THE TYPE OF THE'
     $                 ,' FUNCTION',/,' 1= CONSTANT, 2= NORMAL, 3= UNIFORM'
     $                 ,' 4=EXPONENTIAL')
                   READ(5,*,ERR=135) FUN
                   IF(FUN.LT.1.OR.FUN.GT.4) GO TO 135
                   ENTITY(K,KK,1)=FUN
  160            CONTINUE
  170   CONTINUE
 C
 C=====>> INPUTS OF THE PARAMETERS FOR THE RANDOM FUNCTIONS
 C
        DO 320 K=1,NPROD
               DO 310 KK=1,NMACHK(K)
                   CALL CLSC
                      KKK=ENTITY(K,KK,1)
                      WRITE(6,180)K,KK
  180                 FORMAT(' FOR PRODUCT ',I2,' AREA ',I2,':')
                      GO TO(190,220,250,280),KKK
 C:::::CONSTANT
  190                 WRITE(6,200)
  200                 FORMAT(' ENTER THE TIME CONSTANT')
                      READ(5,*,ERR=190) CONS
                      IF(CONS.LE.0) GO TO 190
                      PARA(K,KK,1)=CONS
                      PARA(K,KK,3)=KKK
                      GO TO 310
 C:::::NORMAL
  220                 WRITE(6,230)
  230                 FORMAT(' ENTER THE MEAN, THE STANDARD DEVIATION')
                      READ(5,*,ERR=220) XMEA,STDE
                      IF(XMEA.LE.0)GO TO 220
                      IF(STDE.LT.0) GO TO 220
                      PARA(K,KK,1)= XMEA
                      PARA(K,KK,2)= STDE
                      PARA(K,KK,3)= KKK
                      GO TO 310
 C::::: UNIFORM
  250                 WRITE(6,260)
  260                 FORMAT(' ENTER THE LOWER AND UPPER LIMITS')
```

```
 *....*...1.........2.........3.........4.........5.........6.........7.*.......8
                      READ(5,*,ERR=250) XLL,XUL
                      IF(XLL.LT.O) GO TO 250
                      IF(XUL.LE.O)  GO TO 250
                      IF(XLL.GE.XUL) GO TO 250
                      PARA(K,KK,1)=XLL
                      PARA(K,KK,2)=XUL
                      PARA(K,KK,3)=KKK
                      GO TO 310
C::::: EXPONENTIAL
 280                  WRITE(6,290)
 290                  FORMAT(' ENTER THE MEAN')
                      READ(5,*,ERR=280) XMEA
                      IF(XMEA.LE.O)GO TO 280
                      PARA(K,KK,1)=XMEA
                      PARA(K,KK,3)=KKK
                      GO TO 310
 310        CONTINUE
 320     CONTINUE
C
C=====>> INPUT THE INSPECTION PARAMETERS.
C
      DO 328 K=1,NPROD
             DO 327  KK=1,NMACHK(K)
                      CALL CLSC
 321                  WRITE(6,322) K,KK
 322  FORMAT(' FOR PRODUCT ',I1,' AREA ',I2,' ENTER THE'
     $,' PERCENTAGE THAT PASSES INSPECTION :')
                      READ(6,*,ERR=321) DEF(K,KK,1)
      IF(DEF(K,KK,1).LT.O.OR.DEF(K,KK,1).GT.1) GO TO 321
 324                  WRITE(6,325)K,KK
 325  FORMAT(' FOR PRODUCT ',I1,' AREA ',I2,' ENTER THE'
     $,' PERCENTAGE OF SCRAP :')
                      READ(6,*,ERR=324) DEF(K,KK,2)
      IF(DEF(K,KK,2).LT.O.OR.DEF(K,KK,2).GT.1) GO TO 324
      IF((DEF(K,KK,1)+DEF(K,KK,2)).GT.1) GO TO 321
 327         CONTINUE
 328  CONTINUE
C
C=====>> WRITE DATA TO FILE
C
      DO  350      K=1,NPROD
      WRITE(3,345) NMACHK(K)
 345  FORMAT(1X,I2)
 350  CONTINUE
      DO  340      K=1,NPROD
              DO    335     KK=1,NMACHK(K)
              WRITE(3,330) ENTITY(K,KK,2),ENTITY(K,KK,1)
 330          FORMAT(1X,F3.0,2X,F2.0)
              WRITE(3,331) PARA(K,KK,1),PARA(K,KK,2),PARA(K,KK,3)
 331          FORMAT(1X,F10.4,1X,F10.4,1X,F2.0)
              WRITE(3,332) DEF(K,KK,1),DEF(K,KK,2)
 332          FORMAT(1X,F6.4,1X,F6.4)
 335          CONTINUE
 340  CONTINUE
      RETURN
      END
```

APPENDIX H

FLOWCHART AND LISTING OF SUBROUTINE ARRIVE

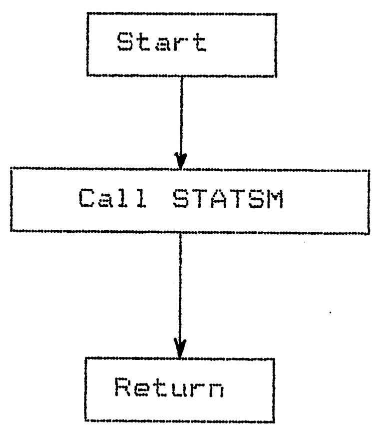

Start
Find the Nature of Arrival
Function is
1
2
3
4
Read Parameter (constant)
Read Parameters (normal)
Read Parameters (uniform)
Read Parameters (exponential)
Store Values and Flags
Return

```
ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N
      NOSYM NORENT   SDUMP AUTODBL(NONE)    NOSXM   IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHAI

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=================================================================C
C::::::                                                     ::::::C
C......      SUBROUTINE ARRIVE :                            ......C
C......      -----------------                              ......C
C......                          THROUGH THIS SUBROUTINE THE......C
C......      USER INPUTS THE ARRIVAL RATES FOR THE DIFFERENT......C
C......      PRODUCTS.                                      ......C
C::::::                                                     ::::::C
C=================================================================
      SUBROUTINE ARRIVE
C
C=====>> INITIALIZATION OF VARIABLES.
C
      COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
     $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
C
C=====>> INPUT STARTS
C
      CALL CLSC
      DO 190 K=1,NPROD
  5           CALL CLSC
             WRITE(6,10)K
 10          FORMAT(' CHOOSE THE ARRIVAL RATE FUNCTION FOR PRODUCT ',
     $              I2)
             WRITE(6,11)
 11          FORMAT(/,17X,' MENU')
             WRITE(6,12)
 12          FORMAT(17X,' ----')
             WRITE(6,13)
 13          FORMAT(/,7X,'1- CONSTANT',6X,'2- NORMAL')
             WRITE(6,14)
 14          FORMAT(7X,'3- UNIFORM ',6X,'4- EXPONENTIAL')
             READ(5,*,ERR=5) NFUN
             IF(NFUN.LT.1.OR.NFUN.GT.4) GO TO 5
             ARVAL(K,1)=NFUN
C
C:::::>> DECISION
C
             GO TO (30,60,90,120),NFUN
C::::::CONSTANT
 30          WRITE(6,35)
 35          FORMAT(' ENTER THE TIME CONSTANT')
             READ(5,*,ERR=30) CONS
             IF(CONS.LE.0) GO TO 30
             ARVAL(K,2)=CONS
             GO TO 180
C::::: NORMAL
 60          WRITE(6,65)
 65          FORMAT(' ENTER THE MEAN AND STANDARD DEVIATION')
             READ(5,*,ERR=60) XMEA,STDE
             IF(XMEA.LE.0) GO TO 60
             IF(STD.LT.0) GO TO 60
```

```
\Y 1985)            VS FORTRAN          DATE: APR 15, 1987    TIME: 07:07:26        N

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

               ARVAL(K,2)=XMEA
               ARVAL(K,3)=STDE
               GO TO 180
C::::: UNIFORM
 90            WRITE(6,95)
 95            FORMAT(' ENTER THE LOWER AND UPPER LIMITS')
               READ(5,*,ERR=90) XLL,XUL
               IF(XLL.LT.O) GO TO 90
               IF(XLL.GE.XUL) GO TO 90
               IF(XUL.LE.O) GO TO 90
               ARVAL(K,2)=XLL
               ARVAL(K,3)=XUL
               GO TO 180
C::::: EXPONENTIAL
 120           WRITE(6,130)
 130           FORMAT(' ENTER THE MEAN')
               READ(5,*,ERR=120) XMEA
               IF(XMEA.LE.O) GO TO 120
               ARVAL(K,2)=XMEA
               GO TO 180
C
C=====>> WRITE DATA TO FILE
C
 180     WRITE(3,185) ARVAL(K,1),ARVAL(K,2),ARVAL(K,3)
 185     FORMAT(1X,F2.0,1X,F10.4,1X,F10.4)
 190     CONTINUE
         RETURN
         END

 SOURCE STATEMENTS = 53, PROGRAM SIZE = 2472 BYTES, PROGRAM NAME = ARRIVE      PAGE

  NO DIAGNOSTICS GENERATED.
```

APPENDIX I

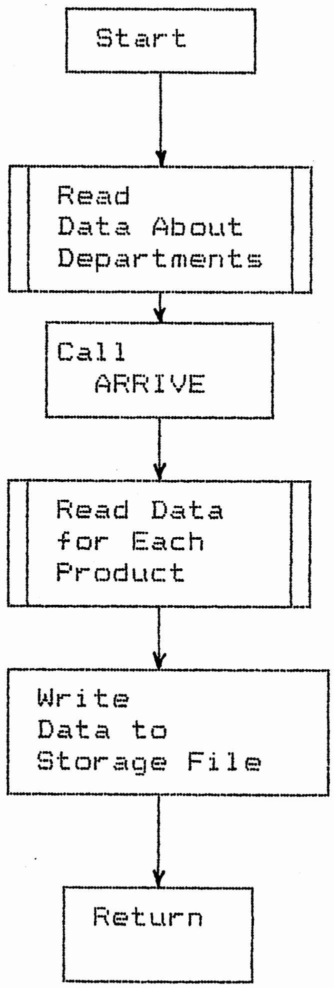
FLOWCHART AND LISTING OF SUBROUTINE STATSA


Start
Call STATSM
Return

```
:ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)      CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                           ::::::C
 C......    SUBROUTINE STATSA :                                   ......C
 C......    -----------------                                     ......C
 C......                      THROUGH THIS SUBROUTINE THE USER     ......C
 C......    CHOOSES THE STATISTICS FOR THE AREA SIMULATION         ......C
 C::::::                                                           ::::::C
 C======================================================================C
       SUBROUTINE STATSA
 C
 C=====>> INITIALIZATION OF VARIABLES.
 C
 C
 C=====>> INPUT STARTS.
 C
       CALL STATSM
       RETURN
       END

 SOURCE STATEMENTS = 4, PROGRAM SIZE = 484 BYTES, PROGRAM NAME = STATSA     PAGE:

  NO DIAGNOSTICS GENERATED.
```

APPENDIX J

FLOWCHART AND LISTING OF SUBROUTINE INPDEP


Start
Read
Data About
Departments
Call
ARRIVE
Read Data
for Each
Product
Write
Data to
Storage File
Return

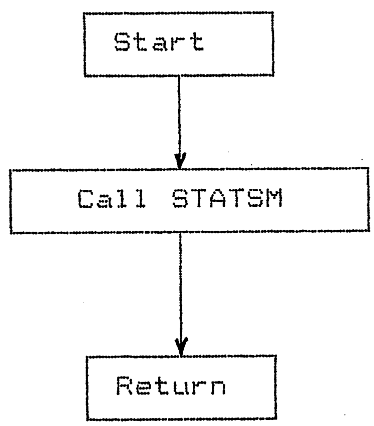
```
ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N(
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM    IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)       CHARI

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=====================================================================C
C::::::                                                         ::::::C
C......    SUBROUTINE INPDEP :                                  ......C
C......    -----------------                                    ......C
C......                       THROUGH THIS SUBROUTINE THE USER  ......C
C......    INPUTS ALL THE DATA NEEDED WHEN SIMULATION OF        ......C
C......    DEPARTMENTS IS NEEDED                                ......C
C::::::                                                         ::::::C
C=====================================================================C
      SUBROUTINE INPDEP
C
C=====>> INITIALIZATION OF VARIABLES
C
      COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
     $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
C
C=====>> INPUTS STARTS
C
 5     CALL CLSC
      WRITE(6,10)
 10   FORMAT(' YOU ARE AT THE DEPARTMENT LEVEL SIMULATION (LEVEL 2)')
      WRITE(6,11)
 11   FORMAT(//)
      WRITE(6,30)
 30   FORMAT(' HOW MANY PRODUCTS YOU WANT TO SIMULATE (MAXIMUM 5) ?')
      READ(5,*,ERR=5) NPROD
      IF(NPROD.LT.0.OR.NPROD.GT.5) GO TO 5
 35    WRITE(6,50)
 50   FORMAT(' HOW MANY DEPARTMENTS IN THE LAYOUT (MAXIMUM 35) ?')
      READ(5,*,ERR=35) MACH
      IF(MACH.LT.1.OR.MACH.GT.35) GO TO 35
      DO 90 K=1,NPROD
 60      WRITE(6,70)K
 70      FORMAT(' HOW MANY DEPARTMENTS ARE NEEDED FOR PRODUCT   ',I2)
         READ(5,*,ERR=60) NMACHK(K)
 90   CONTINUE
      WRITE(3,91) NPROD
 91   FORMAT(1X,I1)
      WRITE(3,92) MACH
 92   FORMAT(1X,I2)
C
C=====>> FIND THE ARRIVAL RATES.
C
      CALL ARRIVE
C
C=====>> MORE INFORMATION TO INPUT
C
      CALL CLSC
      DO 130 K=1,NPROD
             DO 120 KK=1,NMACHK(K)
                     CALL CLSC
 95             WRITE(6,100)K,KK
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8

 100      FORMAT(' WHICH DEPARTMENT DOES PRODUCT ',I2,' GO TO IN THE ',
     $                  I2,' PLACE')
                READ(5,*,ERR=90) RAK
      IF(RAK.LT.11.OR.RAK.GT.17.AND.RAK.LT.21.OR.RAK.GT.27.AND.RAK.LT.
     $31.OR.RAK.GT.37.AND.RAK.LT.41.OR.RAK.GT.47.AND.RAK.LT.51.OR.
     $RAK.GT.57) GO TO 95
                ENTITY(K,KK,2)=RAK
 120         CONTINUE
 130  CONTINUE
C
C=====>> INPUTS OF THE PROCESSING TIMES.
C
      DO 170 K=1,NPROD
             DO 160 KK=1,NMACHK(K)
                     CALL CLSC
 135            WRITE(6,140)KK,K
 140            FORMAT(' WHAT IS THE PROCESSING TIME FOR DEPARTMENT ',
     $                  I2,' WITH PRODUCT ',I2,/,' ENTER THE TYPE OF THE'
     $               ,' FUNCTION',/,' 1= CONSTANT, 2= NORMAL, 3= UNIFORM'
     $               ,' 4=EXPONENTIAL')
                READ(5,*,ERR=135) FUN
                IF(FUN.LT.1.OR.FUN.GT.4) GO TO 135
                ENTITY(K,KK,1)=FUN
 160          CONTINUE
 170  CONTINUE
C
C=====>> INPUTS OF THE PARAMETERS FOR THE RANDOM FUNCTIONS
C
      DO 320 K=1,NPROD
             DO 310 KK=1,NMACHK(K)
                     CALL CLSC
                    KKK=ENTITY(K,KK,1)
                    WRITE(6,180)K,KK
 180                FORMAT(' FOR PRODUCT ',I2,' DEPARTMENT ',I2,':')
                    GO TO(190,220,250,280),KKK
C:::::CONSTANT
 190                WRITE(6,200)
 200                FORMAT(' ENTER THE TIME CONSTANT')
                    READ(5,*,ERR=190) CONS
                    IF(CONS.LE.0) GO TO 190
                    PARA(K,KK,1)=CONS
                    PARA(K,KK,3)=KKK
                    GO TO 310
C:::::NORMAL
 220                WRITE(6,230)
 230                FORMAT(' ENTER THE MEAN, THE STANDARD DEVIATION')
                    READ(5,*,ERR=220) XMEA,STDE
                    IF(XMEA.LE.0) GO TO 220
                    IF(STDE.LT.0) GO TO 220
                    PARA(K,KK,1)= XMEA
                    PARA(K,KK,2)= STDE
                    PARA(K,KK,3)= KKK
                    GO TO 310
C::::: UNIFORM
 250                WRITE(6,260)
 260                FORMAT(' ENTER THE LOWER AND UPPER LIMITS')
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8

                     READ(5,*,ERR=250) XLL,XUL
                     IF(XLL.LT.O) GO TO 250
                     IF(XUL.LE.O) GO TO 250
                     IF(XLL.GE.XUL) GO TO 250
                     PARA(K,KK,1)=XLL
                     PARA(K,KK,2)=XUL
                     PARA(K,KK,3)=KKK
                     GO TO 310
C::::: EXPONENTIAL
 280                 WRITE(6,290)
 290                 FORMAT(' ENTER THE MEAN')
                     READ(5,*,ERR=280) XMEA
                     IF(XMEA.LE.O) GO TO 280
                     PARA(K,KK,1)=XMEA
                     PARA(K,KK,3)=KKK
                     GO TO 310
 310       CONTINUE
 320    CONTINUE
C
C=====>> INPUT THE INSPECTION PARAMETERS.
C
      DO 328 K=1,NPROD
             DO 327  KK=1,NMACHK(K)
                     CALL CLSC
 321                 WRITE(6,322) K,KK
 322  FORMAT(' FOR PRODUCT ',I1,' DEPARTMENT ',I2,' ENTER THE'
     $,' PERCENTAGE THAT PASSES INSPECTION :')
                     READ(6,*,ERR=321) DEF(K,KK,1)
      IF(DEF(K,KK,1).LT.O.OR.DEF(K,KK,1).GT.1) GO TO 321
 324                 WRITE(6,325)K,KK
 325  FORMAT(' FOR PRODUCT ',I1,' DEPARTMENT ',I2,' ENTER THE'
     $,' PERCENTAGE OF SCRAP :')
                     READ(6,*,ERR=324) DEF(K,KK,2)
      IF(DEF(K,KK,2).LT.O.OR.DEF(K,KK,2).GT.1) GO TO 324
      IF((DEF(K,KK,1)+DEF(K,KK,2)).GT.1) GO TO 321
 327         CONTINUE
 328  CONTINUE
C
C=====>> WRITE DATA TO FILE
C
      DO  350      K=1,NPROD
      WRITE(3,345) NMACHK(K)
 345  FORMAT(1X,I2)
 350  CONTINUE
      DO  340      K=1,NPROD
              DO   335      KK=1,NMACHK(K)
              WRITE(3,330) ENTITY(K,KK,2),ENTITY(K,KK,1)
 330          FORMAT(1X,F3.0,2X,F2.0)
              WRITE(3,331) PARA(K,KK,1),PARA(K,KK,2),PARA(K,KK,3)
 331          FORMAT(1X,F10.4,1X,F10.4,1X,F2.0)
              WRITE(3,332) DEF(K,KK,1),DEF(K,KK,2)
 332          FORMAT(1X,F6.4,1X,F6.4)
 335          CONTINUE
 340  CONTINUE
      RETURN
      END
```

APPENDIX K

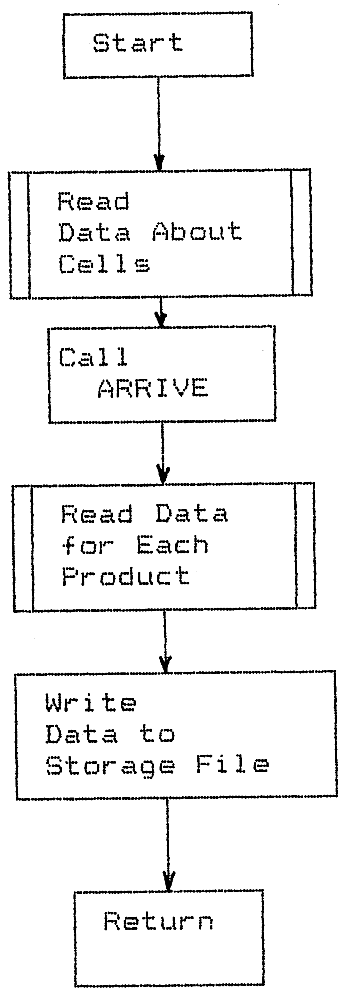
FLOWCHART AND LISTING OF SUBROUTINE STATSD


Start
Call STATSM
Return

```
ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N(
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)     CHARl

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C==================================================================C
 C::::::                                                       ::::::C
 C......     SUBROUTINE STATSD :                              ......C
 C......     -----------------                                ......C
 C......                         THIS SUBROUTINE IS USED TO DECIDE ......C
 C......     ON THE STATISTICS FOR DEPARTMENT SIMULATION      ......C
 C::::::                                                       ::::::C
 C==================================================================C
       SUBROUTINE STATSD
 C
 C=====>> INITIALIZATION OF VARIABLES
 C
 C
 C=====>> INPUT STARTS.
 C
       CALL STATSM
       RETURN
       END

 SOURCE STATEMENTS = 4, PROGRAM SIZE = 484 BYTES, PROGRAM NAME = STATSD     PAGE:

  NO DIAGNOSTICS GENERATED.
```

APPENDIX L

FLOWCHART AND LISTING OF SUBROUTINE INPCEL


Start
Read Data About Cells
Call ARRIVE
Read Data for Each Product
Write Data to Storage File
Return

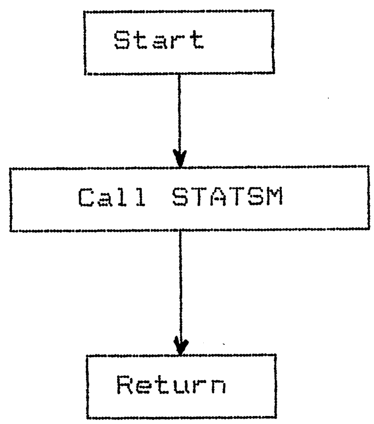
```
ECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NC
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARL

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

C==================================================================C
C::::::                                                      ::::::C
C......    SUBROUTINE INPCEL :                               ......C
C......    ----------------                                  ......C
C......                      THROUGH THIS SUBROUTINE THE USER ......C
C......    INPUTS ALL DATA RELEVANT TO THE SIMULATION OF CELLS......C
C::::::                                                      ::::::C
C==================================================================C
      SUBROUTINE INPCEL
C
C=====>> INITIALIZATION OF VARIABLES
C
      COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
     $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
C
C=====>> INPUTS STARTS
C
 5    CALL CLSC
      WRITE(6,10)
 10   FORMAT(' YOU ARE AT THE CELL LEVEL SIMULATION (LEVEL 3)')
      WRITE(6,11)
 11   FORMAT(//)
      WRITE(6,30)
 30   FORMAT(' HOW MANY PRODUCTS YOU WANT TO SIMULATE (MAXIMUM 5) ?')
      READ(5,*,ERR=5) NPROD
      IF(NPROD.LT.1.OR.NPROD.GT.5) GO TO 5
 35   WRITE(6,50)
 50   FORMAT(' HOW MANY CELLS IN THE LAYOUT (MAXIMUM 35) ?')
      READ(5,*,ERR=35) MACH
      IF(MACH.LT.1.OR.MACH.GT.35) GO TO 35
      DO 90 K=1,NPROD
 55      WRITE(6,70)K
 70      FORMAT(' HOW MANY CELLS ARE USED FOR PRODUCT  ',I2)
         READ(5,*,ERR=55) NMACHK(K)
 90   CONTINUE
      WRITE(3,91) NPROD
 91   FORMAT(1X,I1)
      WRITE(3,92) MACH
 92   FORMAT(1X,I2)
C
C=====>> FIND THE ARRIVAL RATES.
C
      CALL ARRIVE
C
C=====>> MORE INFORMATION TO INPUT
C
      CALL CLSC
      DO 130 K=1,NPROD
             DO 120 KK=1,NMACHK(K)
                    CALL CLSC
 95            WRITE(6,100)K,KK
 100           FORMAT(' WHICH CELL DOES PRODUCT ',I2,' GO TO IN THE ',
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8

     $                   I2,' PLACE')
                   READ(5,*,ERR=95) RAK
      IF(RAK.LT.11.OR.RAK.GT.17.AND.RAK.LT.21.OR.RAK.GT.27.AND.RAK.LT.
     $31.OR.RAK.GT.37.AND.RAK.LT.41.OR.RAK.GT.47.AND.RAK.LT.51.OR.
     $RAK.GT.57) GO TO 95
                   ENTITY(K,KK,2)=RAK
 120            CONTINUE
 130  CONTINUE
C
C=====>> INPUTS OF THE PROCESSING TIMES.
C
      DO 170 K=1,NPROD
             DO 160 KK=1,NMACHK(K)
                        CALL CLSC
 135               WRITE(6,140)KK,K
 140               FORMAT(' WHAT IS THE PROCESSING TIME FOR CELL ',
     $                    I2,' WITH PRODUCT ',I2,/,' ENTER THE TYPE OF THE'
     $                 ,' FUNCTION',/,' 1= CONSTANT, 2= NORMAL, 3= UNIFORM'
     $                 ,' 4=EXPONENTIAL')
                   READ(5,*,ERR=135) FUN
                   IF(FUN.LT.1.OR.FUN.GT.4) GO TO 135
                   ENTITY(K,KK,1)=FUN
 160             CONTINUE
 170  CONTINUE
C
C=====>> INPUTS OF THE PARAMETERS FOR THE RANDOM FUNCTIONS
C
      DO 320 K=1,NPROD
             DO 310 KK=1,NMACHK(K)
                        CALL CLSC
                       KKK=ENTITY(K,KK,1)
                       WRITE(6,180)K,KK
 180                   FORMAT(' FOR PRODUCT ',I2,' CELL ',I2,':')
                       GO TO(190,220,250,280),KKK
C::::::CONSTANT
 190                   WRITE(6,200)
 200                   FORMAT(' ENTER THE TIME CONSTANT')
                       READ(5,*,ERR=190) CONS
                       IF(CONS.LT.0) GO TO 190
                       PARA(K,KK,1)=CONS
                       PARA(K,KK,3)=KKK
                       GO TO 310
C::::::NORMAL
 220                   WRITE(6,230)
 230                   FORMAT(' ENTER THE MEAN, THE STANDARD DEVIATION')
                       READ(5,*,ERR=220) XMEA,STDE
                       IF(XMEA.LE.0) GO TO 220
                       IF(STDE.LT.0) GO TO 220
                       PARA(K,KK,1)= XMEA
                       PARA(K,KK,2)= STDE
                       PARA(K,KK,3)= KKK
                       GO TO 310
C:::::: UNIFORM
 250                   WRITE(6,260)
 260                   FORMAT(' ENTER THE LOWER AND UPPER LIMITS')
                       READ(5,*,ERR=250) XLL,XUL
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8
                    IF(XLL.LT.O) GO TO 250
                    IF(XUL.LE.O) GO TO 250
                    IF(XLL.GE.XUL) GO TO 250
                    PARA(K,KK,1)=XLL
                    PARA(K,KK,2)=XUL
                    PARA(K,KK,3)=KKK
                    GO TO 310
C::::: EXPONENTIAL
 280                WRITE(6,290)
 290                FORMAT(' ENTER THE MEAN')
                    READ(5,*,ERR=280) XMEA
                    IF(XMEA.LE.O) GO TO 280
                    PARA(K,KK,1)=XMEA
                    PARA(K,KK,3)=KKK
                    GO TO 310
 310         CONTINUE
 320      CONTINUE
C
C=====>> INPUT THE INSPECTION PARAMETERS.
C
      DO 328 K=1,NPROD
             DO 327  KK=1,NMACHK(K)
                    CALL CLSC
 321                WRITE(6,322) K,KK
 322  FORMAT(' FOR PRODUCT ',I1,' CELL ',I2,' ENTER THE'
     $,' PERCENTAGE THAT PASSES INSPECTION :')
                    READ(6,*,ERR=321) DEF(K,KK,1)
      IF(DEF(K,KK,1).LT.O.OR.DEF(K,KK,1).GT.1) GO TO 321
 324                WRITE(6,325)K,KK
 325  FORMAT(' FOR PRODUCT ',I1,' CELL ',I2,' ENTER THE'
     $,' PERCENTAGE OF SCRAP :')
                    READ(6,*,ERR=324) DEF(K,KK,2)
      IF(DEF(K,KK,2).LT.O.OR.DEF(K,KK,2).GT.1) GO TO 324
      IF((DEF(K,KK,1)+DEF(K,KK,2)).GT.1) GO TO 321
 327           CONTINUE
 328  CONTINUE
C
C=====>> WRITE DATA TO FILE
C
      DO  350       K=1,NPROD
      WRITE(3,345) NMACHK(K)
 345  FORMAT(1X,I2)
 350  CONTINUE
      DO  340       K=1,NPROD
              DO    335      KK=1,NMACHK(K)
              WRITE(3,330) ENTITY(K,KK,2),ENTITY(K,KK,1)
 330          FORMAT(1X,F3.O,2X,F2.O)
              WRITE(3,331) PARA(K,KK,1),PARA(K,KK,2),PARA(K,KK,3)
 331          FORMAT(1X,F10.4,1X,F10.4,1X,F2.O)
              WRITE(3,332) DEF(K,KK,1),DEF(K,KK,2)
 332          FORMAT(1X,F6.4,1X,F6.4)
 335          CONTINUE
 340  CONTINUE
      RETURN
      END
```

APPENDIX M

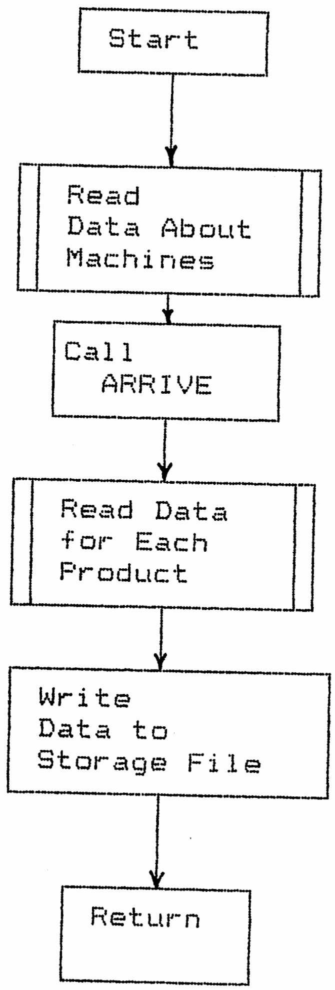
FLOWCHART AND LISTING OF SUBROUTINE STATSC


Start
Call STATSM
Return

```
CT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N
     NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
       OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHAR

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

C==================================================================C 00009000
C::::::                                                      ::::::C 00009010
C......     SUBROUTINE STATSC :                              ......C 00009020
C......     -----------------                                ......C 00009030
C......                       THIS SUBROUTINE IS USED TO DECIDE ......C 00009040
C......     ON STATISTICS WHEN SIMULATION OF CELLS IS NEEDED    ......C 00009050
C::::::                                                      ::::::C 00009060
C==================================================================C 00009070
      SUBROUTINE STATSC                                              00009080
C                                                                    00009090
C=====>> INITIALIZATION OF VARIABLES                                 00009100
C                                                                    00009110
C                                                                    00009120
C=====>> INPUT STARTS                                                00009130
C                                                                    00009140
      CALL STATSM                                                    00009150
      RETURN                                                         00009160
      END                                                            00009170

SOURCE STATEMENTS = 4, PROGRAM SIZE = 484 BYTES, PROGRAM NAME = STATSC     PAGE:

 NO DIAGNOSTICS GENERATED.
```

APPENDIX N

FLOWCHART AND LISTING OF SUBROUTINE INPMAC


Start
Read
Data About
Machines
Call
ARRIVE
Read Data
for Each
Product
Write
Data to
Storage File
Return

```
FECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NO
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)       CHARL

  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

  C================================================================C
  C::::::                                                     ::::::C
  C......      SUBROUTINE INPMAC :                             ......C
  C......      -----------------                               ......C
  C......                     FROM THIS SUBROUTINE THE USER    ......C
  C......      INPUTS ALL THE DATA FOR THE SIMULATION AT LEVEL 4  ......C
  C......      WHICH IS THE SIMULATION OF THE MACHINES OPERATIONS ......C
  C::::::                                                     ::::::C
  C================================================================C
         SUBROUTINE INPMAC
  C
  C=====>> INITIALIZATION OF VARIABLES
  C
        COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
       $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
  C
  C=====>> INPUTS STARTS
  C
    5   CALL CLSC
        WRITE(6,10)
   10   FORMAT(' YOU ARE AT THE MACHINE LEVEL SIMULATION (LEVEL 4)')
        WRITE(6,11)
   11   FORMAT(//)
   15   WRITE(6,30)
   30   FORMAT(' HOW MANY PRODUCTS YOU WANT TO SIMULATE (MAXIMUM 5) ?')
        READ(5,*,ERR=15) NPROD
        IF(NPROD.LT.1.OR.NPROD.GT.5) GO TO 5
   35   WRITE(6,50)
   50   FORMAT(' HOW MANY MACHINES IN THE LAYOUT (MAXIMUM 35) ?')
        READ(5,*,ERR=35) MACH
        IF(MACH.LT.1.OR.MACH.GT.35) GO TO 35
        DO 90 K=1,NPROD
   60      WRITE(6,70)K
   70      FORMAT(' HOW MANY OPERATIONS ARE USED FOR PRODUCT  ',I2)
           READ(5,*,ERR=60) NMACHK(K)
   90   CONTINUE
        WRITE(3,91) NPROD
   91   FORMAT(1X,I1)
        WRITE(3,92) MACH
   92   FORMAT(1X,I2)
  C
  C=====>> FIND THE ARRIVAL RATES.
  C
        CALL ARRIVE
  C
  C=====>> MORE INFORMATION TO INPUT
  C
        CALL CLSC
        DO 130 K=1,NPROD
              DO 120 KK=1,NMACHK(K)
                     CALL CLSC
   95           WRITE(6,100)K,KK
```

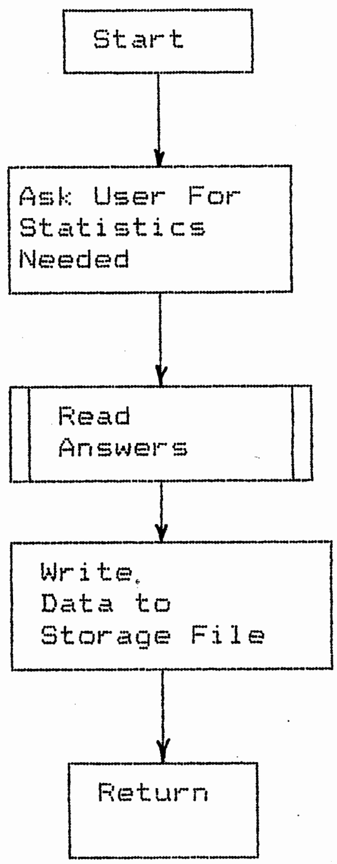
```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8
 100          FORMAT(' WHICH MACHINE DOES PRODUCT ',I2,' GO TO IN THE ',
     $                   I2,' PLACE')
                   READ(5,*,ERR=95) RAK
      IF(RAK.LT.11.OR.RAK.GT.17.AND.RAK.LT.21.OR.RAK.GT.27.AND.RAK.LT.
     $31.OR.RAK.GT.37.AND.RAK.LT.41.OR.RAK.GT.47.AND.RAK.LT.51.OR.
     $RAK.GT.57)
     $ GO TO 95
                   ENTITY(K,KK,2)=RAK
 120          CONTINUE
 130  CONTINUE
C
C=====>> INPUTS OF THE PROCESSING TIMES.
C
      DO 170 K=1,NPROD
              DO 160 KK=1,NMACHK(K)
                        CALL CLSC
 135               WRITE(6,140)KK,K
 140               FORMAT(' WHAT IS THE PROCESSING TIME FOR MACHINE ',
     $                    I2,' WITH PRODUCT ',I2,/,' ENTER THE TYPE OF THE'
     $                ,' FUNCTION',/,' 1= CONSTANT, 2= NORMAL, 3= UNIFORM'
     $                ,' 4=EXPONENTIAL')
                   READ(5,*,ERR=135) FUN
                   IF(FUN.LT.1.OR.FUN.GT.4.)GO TO 135
                   ENTITY(K,KK,1)=FUN
 160           CONTINUE
 170  CONTINUE
C
C=====>> INPUTS OF THE PARAMETERS FOR THE RANDOM FUNCTIONS
C
      DO 320 K=1,NPROD
              DO 310 KK=1,NMACHK(K)
                        CALL CLSC
                       KKK=ENTITY(K,KK,1)
                       WRITE(6,180)K,KK
 180                   FORMAT(' FOR PRODUCT ',I2,' MACHINE ',I2,':')
                       GO TO(190,220,250,280),KKK
C::::::CONSTANT
 190                   WRITE(6,200)
 200                   FORMAT(' ENTER THE TIME CONSTANT')
                       READ(5,*,ERR=190) CONS
                       IF(CONS.LE.0) GO TO 190
                       PARA(K,KK,1)=CONS
                       PARA(K,KK,3)=KKK
                       GO TO 310
C::::::NORMAL
 220                   WRITE(6,230)
 230                   FORMAT(' ENTER THE MEAN, THE STANDARD DEVIATION')
                       READ(5,*,ERR=220) XMEA,STDE
                       IF(XMEA.LE.0)GO TO 220
                       IF(STDE.LT.0) GO TO 220
                       PARA(K,KK,1)= XMEA
                       PARA(K,KK,2)= STDE
                       PARA(K,KK,3)= KKK
                       GO TO 310
C:::::: UNIFORM
 250                   WRITE(6,260)
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8
 260                    FORMAT(' ENTER THE LOWER AND UPPER LIMITS')
                        READ(5,*,ERR=250) XLL,XUL
                        IF(XLL.LT.0) GO TO 250
                        IF(XLL.GE.XUL) GO TO 250
                        IF(XUL.LE.0) GO TO 250
                        PARA(K,KK,1)=XLL
                        PARA(K,KK,2)=XUL
                        PARA(K,KK,3)=KKK
                        GO TO 310
C::::: EXPONENTIAL
 280                    WRITE(6,290)
 290                    FORMAT(' ENTER THE MEAN')
                        READ(5,*,ERR=280) XMEA
                        IF(XMEA.LE.0) GO TO 280
                        PARA(K,KK,1)=XMEA
                        PARA(K,KK,3)=KKK
                        GO TO 310
 310        CONTINUE
 320     CONTINUE
C
C=====>> INPUT THE INSPECTION PARAMETERS.
C
      DO 328 K=1,NPROD
             DO 327  KK=1,NMACHK(K)
                        CALL CLSC
 321                    WRITE(6,322) K,KK
 322  FORMAT(' FOR PRODUCT ',I1,' MACHINE ',I2,' ENTER THE'
     $,' PERCENTAGE THAT PASSES INSPECTION :')
                        READ(6,*,ERR=321) DEF(K,KK,1)
      IF(DEF(K,KK,1).LT.0.OR.DEF(K,KK,1).GT.1) GO TO 321
 324                    WRITE(6,325) K,KK
 325  FORMAT(' FOR PRODUCT ',I1,' MACHINE ',I2,' ENTER THE'
     $,' PERCENTAGE OF SCRAP :')
                        READ(6,*,ERR=324) DEF(K,KK,2)
      IF(DEF(K,KK,2).LT.0.OR.DEF(K,KK,2).GT.1) GO TO 324
      IF((DEF(K,KK,1)+DEF(K,KK,2)).GT.1) GO TO 321
 327         CONTINUE
 328  CONTINUE
C
C=====>> WRITE DATA TO FILE
C
      DO  350       K=1,NPROD
      WRITE(3,345) NMACHK(K)
 345  FORMAT(1X,I2)
 350  CONTINUE
      DO  340       K=1,NPROD
              DO    335      KK=1,NMACHK(K)
              WRITE(3,330) ENTITY(K,KK,2),ENTITY(K,KK,1)
 330          FORMAT(1X,F3.0,2X,F2.0)
              WRITE(3,331) PARA(K,KK,1),PARA(K,KK,2),PARA(K,KK,3)
 331          FORMAT(1X,F10.4,1X,F10.4,1X,F2.0)
              WRITE(3,332) DEF(K,KK,1),DEF(K,KK,2)
 332          FORMAT(1X,F6.4,1X,F6.4)
 335          CONTINUE
 340  CONTINUE
      RETURN
```

```
(MAY 1985)            VS FORTRAN          DATE: APR 15, 1987    TIME: 07:07:26      N
   *....*...1.........2.........3.........4.........5.........6.........7.*.......8
         END                                                                00002920
   SOURCE STATEMENTS = 118, PROGRAM SIZE = 6176 BYTES, PROGRAM NAME = INPMAC     PAG
    NO DIAGNOSTICS GENERATED.
```

APPENDIX O

FLOWCHART AND LISTING OF SUBROUTINE STATSM


Start
Ask User For Statistics Needed
Read Answers
Write Data to Storage File
Return

```
'ECT:   NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED   NOTEST N
        NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
          OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)      CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=====================================================================C
C:::::                                                          ::::::C
C.....    SUBROUTINE STATSM :                                   ......C
C.....     -----------------                                    ......C
C.....                        THIS SUBROUTINE IS USED WHEN      ......C
C.....     SIMULATION OF MACHINES IS DONE, TO DECIDE ON STATISTICS....C
C:::::                                                          ::::::C
C=====================================================================C
      SUBROUTINE STATSM
C
C=====>> INITIALIZATION OF VARIABLES.
C
      REAL XX(100)
      DO 5 IK=1,100
 5    XX(IK)=0
C
C=====>> INPUT STARTS
C
      CALL CLSC
      WRITE(6,10)
 10   FORMAT(' THIS SUBROUTINE GIVES YOU THE OPTION TO CHOOSE SOME',
     $'STATISTICS.')
      WRITE(6,11)
 11   FORMAT(' TO ANSWER THE QUESTIONS, TYPE 1 FOR YES AND 0 FOR NO.')
      WRITE(6,15)
 15   FORMAT(' DO YOU WANT TO SEE ALL STATISTICS AVAILABLE')
      READ(5,30) NANS
      IF(NANS.NE.1) GO TO 16
        XX(1)=1
        XX(2)=1
        XX(3)=1
        XX(4)=1
        XX(5)=1
      GO TO 75
 16   WRITE(6,20)
 20   FORMAT(' WOULD YOU LIKE THE INSPECTION STATISTICS ?')
      READ(5,30,ERR=16) NANS
 30   FORMAT(I1)
      IF(NANS.EQ.1) XX(1)=1
 35   WRITE(6,40)
 40   FORMAT(' DO YOU WANT THE STATISTICS OF THE TIME BETWEEN',
     $' COMPLETION OF PRODUCTS ?')
      READ(5,30,ERR=35) NANS
      IF(NANS.EQ.1) XX(2)=1
 50   WRITE(6,60)
 60   FORMAT(' DO YOU WANT THE INTERVAL STATISTIC (TIME IN SYSTEM)?')
      READ(5,30,ERR=50) NANS
      IF(NANS.EQ.1) XX(3)=1
 75   DO 80 IK=1,5
 80   WRITE(3,90) XX(IK)
 90   FORMAT(1X,F2.0)
      RETURN
```

```
 (MAY 1985)            VS FORTRAN          DATE: APR 15, 1987    TIME: 07:07:27      N
    *....*...1.........2.........3.........4.........5.........6.........7.*.......8
)          END                                                           00009700
*    SOURCE STATEMENTS = 37, PROGRAM SIZE = 2068 BYTES, PROGRAM NAME = STATSM      PAGE
*     NO DIAGNOSTICS GENERATED.
```

APPENDIX P

JCL OF THE INPUT PROGRAM

JCL FOR INPUT PROGRAM

To use interactive mode on the IBM main frame a library had to be created. The JCL needed for this purpose is shown in the next page. Once the library is created, a different JCL has to be used. This JCL is Shown in page 182.

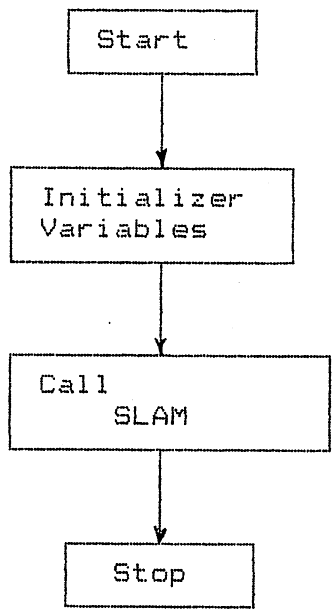
```
//FIRSTIM JOB (nnnnn,sss-ss-ssss),'PROJEC',TIME=0,40,CLASS=A,
// MSGCLASS=X
/*PASSWORD xxxx
/*JOBPARM ROOM=P
//  EXEC FORTVCL
//FORT.SYSIN DD *
(Source Code)
      .
      .
//LKED.SYSLMOD DD DSNAME=UnnnnnA.LOAD.LIB,DISP(NEW,CATLG),
//  UNIT=STORAGE,SPACE=(TRK,(10,2,2)),
//  DCB=(LRECL=0,BLKSIZE=23476,RECFN=U,DSORG=PO)
//LKED.SYSIN DD *
  NAME progname(R)
//

    Where,
           nnnnn= Your 5 Digit Project Number
     sss-ss-ssss= Your Social Security Number
UnnnnnA.LOAD.LIB= A Suggested Name for the Library
     progname(R)= The Name of This Program or Subroutine
                  The (R) is required and Means Replace
                  If New File, The (R) Will Create
```

```
//OTHERTM JOB (nnnnn,sss-ss-ssss),'PROJEC',TIME=0,40,CLASS=A,
// MSGCLASS=X
/*PASSWORD xxxx
/*JOBPARM ROOM=P
//  EXEC FORTVCL
//FORT.SYSIN DD *
(Source Code)
      .
      .
//LKED.SYSLMOD DD DSNAME=UnnnnnA.LOAD.LIB,DISP=OLD,UNIT=3380
//LKED.SYSIN DD *
  NAME progname(R)
//
```

```
    Where,
           nnnnn= Your 5 Digit Project Number
     sss-ss-ssss= Your Social Security Number
UnnnnnA.LOAD.LIB= A Suggested Name for the Library
     progname(R)= The Name of This Program or Subroutine
                  The (R) is required and Means Replace
                  If New File, The (R) Will Create
```

APPENDIX Q

FLOWCHART AND LISTING OF MAIN IN THE SIMULATION PROGRAM

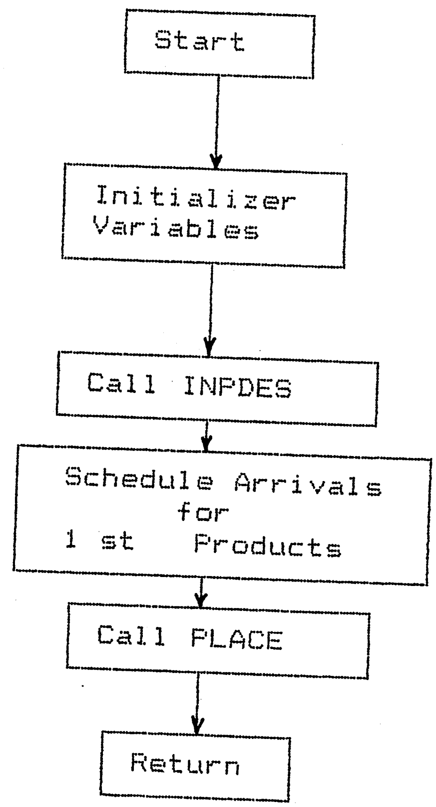

Start
Initializer
Variables
Call
SLAM
Stop

```
IONS (EXECUTE): GOSTMT

=ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED   NOTEST N
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHAR

   *....*...1.........2.........3.........4.........5.........6.........7.*.......8

   C=================================================================C
   C::::::                                                     ::::::C
   C......   PROJECT :  THESIS FOR MASTER OF SCIENCE DEGREE IN   ......C
   C......              INDUSTRIAL ENGINEERING                 ......C
   C......   CANDIDATE : IMED JAMOUSSI                         ......C
   C......   ADVISOR : DR. JOE MIZE                            ......C
   C......   LANGUAGE : FORTRAN & SLAM                         ......C
   C......   SYSTEM : IBM MAIN FRAME                           ......C
   C::::::                                                     ::::::C
   C=================================================================C
   C*****************************************************************C
   C::::::                                                      ::::::C
   C......     LIST OF VARIABLES :                              ......C
   C......     ================                                 ......C
   C......      LEVEL : LEVEL OF SIMULATION (AREA,CELL...)      ......C
   C......      NPROD : NUMBER OF PRODUCTS IN THE SYSTEM        ......C
   C......      MACH : NUMBER OF STATIONS                       ......C
   C......      M(5)  : ARRAY TO KEEP TRACK OF PRODUCTS/MACHINES ......C
   C......      NMACHK(5) : NUMBER OF MACHINES NEEDED PER PRODUCT ......C
   C......      ENTITY(5,35,2) : 5 MACHINES, 35 STATIONS        ......C
   C......      ENTITY(K,KK,1): TYPE OF PROCESS FUNCTION        ......C
   C......      ENTITY(K,KK,2): FLAG TO KEEP TRACK OF THE ORDER ......C
   C......      PARA(5,35,3) : 5 MACHINES, 35 STATIONS.         ......C
   C......      PARA(K,KK,1): PARAMETER 1 OF PROCESSING FUNCTION ......C
   C......      PARA(K,KK,2): PARAMETER 2 OF PROCESSING FUNCTION ......C
   C......      PARA(K,KK,3): FLAG FOR PROCESSING FUNCTION TYPE ......C
   C......      ARVAL(5,3) : ARRIVAL RATES PER MACHINE          ......C
   C......      ARVAL(K,1) : TYPE OF FUNCTION                   ......C
   C......      ARVAL(K,2), ARVAL(K,3): FUNCTION PARAMETERS 1 AND 2......C
   C::::::                                                      ::::::C
   C*****************************************************************C
   C=================================================================C
   C::::::                                                      ::::::C
   C......   MAIN PROGRAM :                                     ......C
   C......   ------------                                       ......C
   C......               THIS PROGRAM IS TO INITIALIZE SOME     ......C
   C......   VARIABLES AND THEN ACCESS TO SLAM.                 ......C
   C::::::                                                      ::::::C
   C=================================================================C
         DIMENSION NSET(10000)
         COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
        1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
         COMMON QSET(10000)
         EQUIVALENCE (NSET(1),QSET(1))
         NNSET=10000
         NCRDR=5
         NPRNT=6
         NTAPE=1
         CALL SLAM
         STOP
```

```
 (MAY 1985)           VS FORTRAN          DATE: APR 15, 1987    TIME: 14:03:17      N
     *....*...1.........2.........3.........4.........5.........6.........7.*.......8
11         END                                                               00000570
S*   SOURCE STATEMENTS = 11, PROGRAM SIZE = 952 BYTES, PROGRAM NAME = MAIN        PAGE:
S*    NO DIAGNOSTICS GENERATED.
```

APPENDIX R

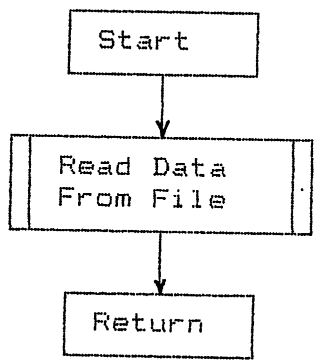
FLOWCHART AND LISTING OF SUBROUTINE INTLC


Start
Initializer Variables
Call INPDES
Schedule Arrivals for 1 st Products
Call PLACE
Return

```
ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(0) LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)      CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                           ::::::C
 C......     SUBROUTINE INITIALIZE :                              ......C
 C......     ---------------------                                ......C
 C......                           THIS SUBROUTINE INITIALIZES    ......C
 C......     THE INITIAL CONDITIONS.                              ......C
 C::::::                                                           ::::::C
 C======================================================================C
       SUBROUTINE INTLC
 C
 C=====>> INITIALIZATION OF VARIABLES
 C
       COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
       COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
      $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
       DO 10 K=1,5
          ATRIB(K)=0
          XX(K)=0
  10   CONTINUE
 C=====>> THE FOLLOWING IS TO GET THE USER INPUT.
 C
       CALL INPRED
 C
 C=====>> ASSIGN THE FIRST ARRIVALS AND PUT THEM IN THE APPROPRIATE
 C=====>> MACHINE.
 C
       DO 20 J=1,NPROD
          M(J)=0
          XTIM=TIM2(J)
          ATRIB(1)=J
          ATRIB(4)=0.
           ATRIB(7)=TNOW
          CALL SCHDL(1,XTIM,ATRIB)
  20      CONTINUE
       RETURN
       END

 SOURCE STATEMENTS = 18, PROGRAM SIZE = 1540 BYTES, PROGRAM NAME = INTLC      PAGE

  NO DIAGNOSTICS GENERATED.

OF COMPILATION 2 ******
```

APPENDIX S

FLOWCHART AND LISTING OF SUBROUTINE INPRED

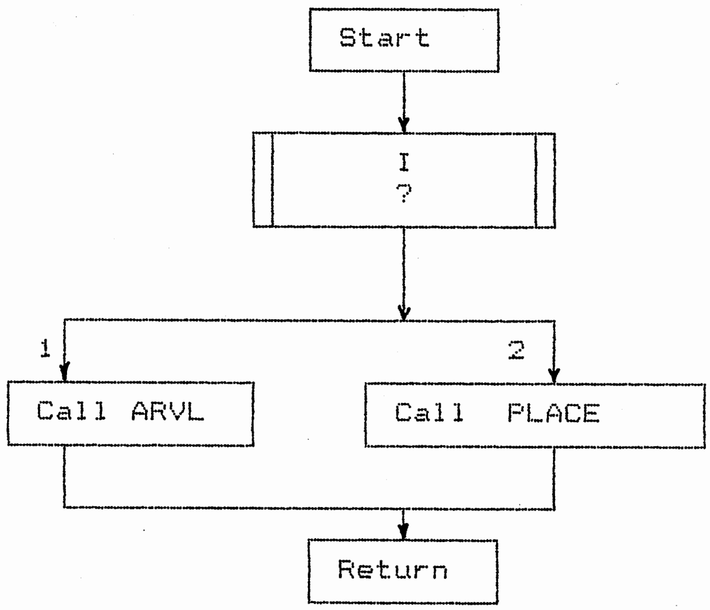

Start
Read Data
From File
Return

```
:ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED   NOTEST NC
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)       CHARL

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                           ::::::C
 C......    SUBROUTINE INPRED :                                   ......C
 C......    -----------------                                     ......C
 C......                      THIS SUBROUTINE ACCESSES ALL THE    ......C
 C......    RELEVANT INFORMATION ALREADY STORED INTO A FILE.      ......C
 C::::::                                                           ::::::C
 C======================================================================C
       SUBROUTINE INPRED
       COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
       COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
      $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
       CHARACTER DATE*8,PROJEC*12,DEPART*20,NAME*20,REFE*5
       COMMON/USER2/DATE,PROJEC,DEPART,NAME,REFE
 C
       READ(3,1) DATE
   1   FORMAT(1X,A8)
       READ(3,2) PROJEC
   2   FORMAT(1X,A12)
       READ(3,3) DEPART
   3   FORMAT(1X,A20)
       READ(3,4) NAME
   4   FORMAT(1X,A20)
       READ(3,5) REFE
   5   FORMAT(1X,A5)
       READ(3,10) LEVEL
  10   FORMAT(1X,I1)
       READ(3,15)TIMED
  15   FORMAT(1X,F12.2)
       GO TO (100,300,500,700),LEVEL
 C
 C=====>> READ AREA SIMULATION DATA.
 C
  100  CONTINUE
       GO TO 700
 C
 C=====>> READ DEPARTMENT SIMULATION DATA.
 C
  300  CONTINUE
       GO TO 700
 C
 C=====>> READ CELL SIMULATION DATA.
 C
  500  CONTINUE
       GO TO 700
 C
 C=====>> READ MACHINE SIMULATION DATA.
 C
  700  CONTINUE
       READ(3,701)NPROD
  701  FORMAT(1X,I1)
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8

      READ(3,702)MACH
 702  FORMAT(1X,I2)
C
      DO 710 K=1,NPROD
         READ(3,703) ARVAL(K,1),ARVAL(K,2),ARVAL(K,3)
 703     FORMAT(1X,F2.0,1X,F10.4,1X,F10.4)
 710  CONTINUE
C
      DO 715 K=1,NPROD
         READ(3,711)NMACHK(K)
 711     FORMAT(1X,I2)
 715  CONTINUE
C
      DO 730 K=1,NPROD
            DO 720 KK=1,NMACHK(K)
               READ(3,716)ENTITY(K,KK,2),ENTITY(K,KK,1)
 716           FORMAT(1X,F3.0,2X,F2.0)
               READ(3,717)PARA(K,KK,1),PARA(K,KK,2),PARA(K,KK,3)
 717           FORMAT(1X,F10.4,1X,F10.4,1X,F2.0)
               READ(3,718)DEF(K,KK,1),DEF(K,KK,2)
 718           FORMAT(1X,F6.4,1X,F6.4)
 720        CONTINUE
 730  CONTINUE
C
      DO 740 K=1,5
         READ(3,735) XX(K)
 735     FORMAT(1X,F2.0)
 740  CONTINUE
 1000 RETURN
      END

SOURCE STATEMENTS = 55, PROGRAM SIZE = 3156 BYTES, PROGRAM NAME = INPRED     PAGE

 NO DIAGNOSTICS GENERATED.

' OF COMPILATION 4 ******
```

APPENDIX T

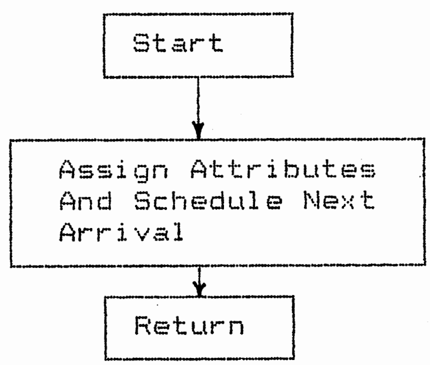
FLOWCHART AND LISTING OF SUBROUTINE EVENT


Start
I
?
1
2
Call ARVL
Call PLACE
Return

```
AY 1985)            VS FORTRAN          DATE: APR 15, 1987    TIME: 14:03:17

ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST N'
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C=====================================================================C00000950
 C::::::                                                         ::::::C00000960
 C......       SUBROUTINE EVENT :                                ......C00000970
 C......       ----------------                                  ......C00000980
 C......                        THIS SUBROUTINE IS TO CALL THE   ......C00000990
 C......       APPROPRIATE EVENT USING THE EVENT CALENDAR.       ......C00001000
 C::::::                                                         ::::::C00001010
 C=====================================================================C00001020
       SUBROUTINE EVENT(I)                                              00001030
       GO TO (1,2),I                                                    00001040
  1    CALL ARVL                                                        00001050
       RETURN                                                           00001060
  2    CALL PLACE                                                       00001070
       RETURN                                                           00001080
       END                                                              00001090

 SOURCE STATEMENTS = 7, PROGRAM SIZE = 596 BYTES, PROGRAM NAME = EVENT      PAGE:

  NO DIAGNOSTICS GENERATED.

OF COMPILATION 3 ******
```

APPENDIX U

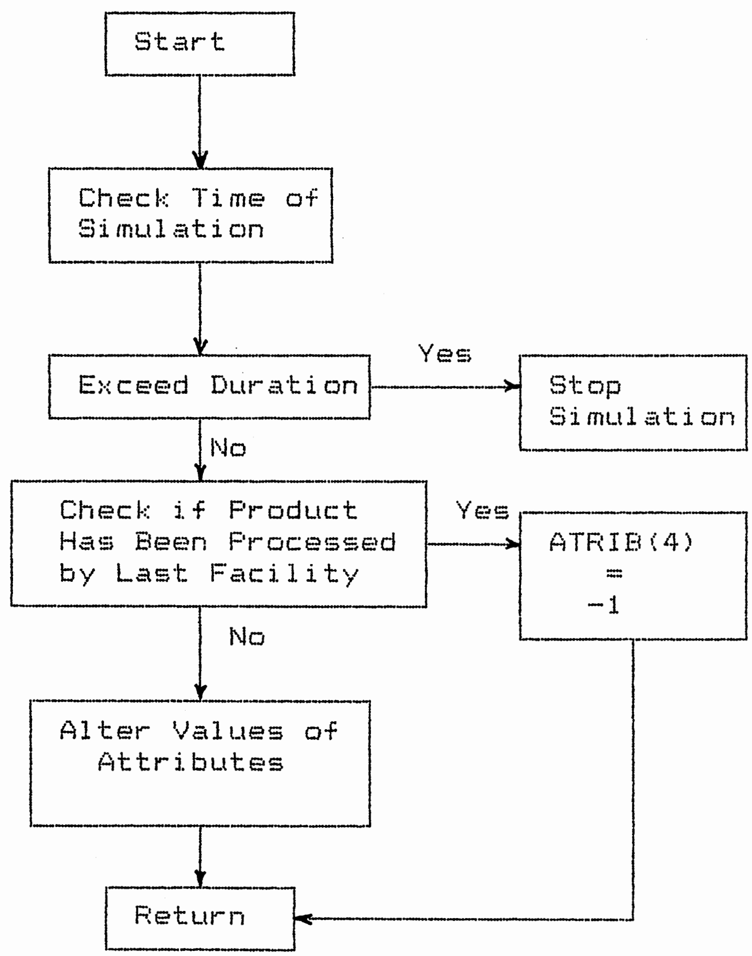
FLOWCHART AND LISTING OF SUBROUTINE ARVL


Start
Assign Attributes And Schedule Next Arrival
Return

```
=ECT:   NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED   NOTEST N
        NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
          OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHAR

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                          ::::::C
 C......                                                          ......C
 C......      SUBROUTINE ARVL :                                   ......C
 C......      ---------------                                     ......C
 C......                      THROUGH THIS SUBROUTINE THE PROGRAM.....C
 C......      SCHEDULES NEW ARRIVALS.                             ......C
 C::::::                                                          ::::::C
 C======================================================================C
       SUBROUTINE ARVL
 C
 C=====>> INITIALIZATION OF VARIABLES.
 C
       COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
       COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
      $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
 C
 C=====>> FIND THE CORRECT PRODUCT AND FIND THE INCOMING ONE
 C
       KKK=ATRIB(1)
       XTIM=TIM2(KKK)
       GO TO (100,200,300,400,500),KKK
 C
 C=====>> MARK THE PRODUCT WITH ITS KIND SPECIFICATION.
 C
  100  ATRIB(1)=1
       GO TO 600
  200  ATRIB(1)=2
       GO TO 600
  300  ATRIB(1)=3
       GO TO 600
  400  ATRIB(1)=4
       GO TO 600
  500  ATRIB(1)=5
       GO TO 600
 C
 C=====>> PUT THE PRODUCT IN THE CORRECT MACHINE.
 C
  600  ATRIB(4)=0.
       ATRIB(7)=TNOW
       CALL SCHDL(1,XTIM,ATRIB)
       CALL ENTER(1,ATRIB)
       RETURN
       END

 SOURCE STATEMENTS = 22, PROGRAM SIZE = 1560 BYTES, PROGRAM NAME = ARVL       PAGE

  NO DIAGNOSTICS GENERATED.

)F COMPILATION 5 ******
```

APPENDIX V

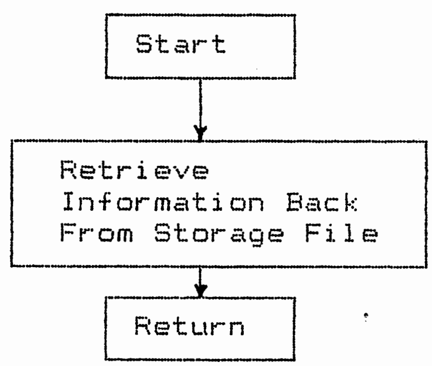
FLOWCHART AND LISTING OF SUBROUTINE PLACE


Start
Check Time of Simulation
Exceed Duration
Yes
Stop Simulation
No
Check if Product Has Been Processed by Last Facility
Yes
ATRIB(4) = -1
No
Alter Values of Attributes
Return

```
ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NC
      NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
        OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)       CHARL

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=====================================================================C
C:::::                                                           :::::C
C......      SUBROUTINE PLACE :                                  ......C
C......      ----------------                                    ......C
C......                        THIS SUBROUTINE IS THE LIAISON    ......C
C......      BETWEEN THE DESCRETE PART AND THE NETWORK CODE.     ......C
C......      IT PLACES THE ENTITY IN THE RIGHT SECTION OF THE    ......C
C......      NETWORK.                                            ......C
C:::::                                                           :::::C
C=====================================================================C
      SUBROUTINE PLACE
C
C=====>> INITIALIZATION OF VARIABLES.
C
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
     $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
C
C=====>> GET THE PRODUCT.
C
      IF(TNOW.GE.TIMED) MSTOP=-1
      J=ATRIB(1)
C
C=====>> GET THE NEXT MACHINE (OR AREA, OR DEPARTMENT, OR CELL)
C
      ATRIB(4)=ATRIB(4)+1
      ML=ATRIB(4)
C
C=====>> IF LAST MACHINE TERMINATE
C
      IF(ML.LE.NMACHK(J)) GO TO 50
      ATRIB(4)=-1.
      RETURN
C
C=====>> GET THE MACHINE SPECIFICATION (IN THE LAYOUT)
C
 50   KK= ENTITY(J,ML,2)
      ATRIB(3)=KK
      ATRIB(5)=DEF(J,ML,1)
      ATRIB(6)=DEF(J,ML,2)
C
C=====>> FIND THE PROCESS TIME
C
      ATRIB(2)=TIM1(J,ML)
C
C=====>> ENTERS IN THE RIGHT SECTION OF THE NETWORK
C
      RETURN
      END

SOURCE STATEMENTS = 17, PROGRAM SIZE = 1676 BYTES, PROGRAM NAME = PLACE       PAGE:
```

APPENDIX W

FLOWCHART AND LISTING OF SUBROUTINE OTPUT


Start
Retrieve Information Back From Storage File
Return

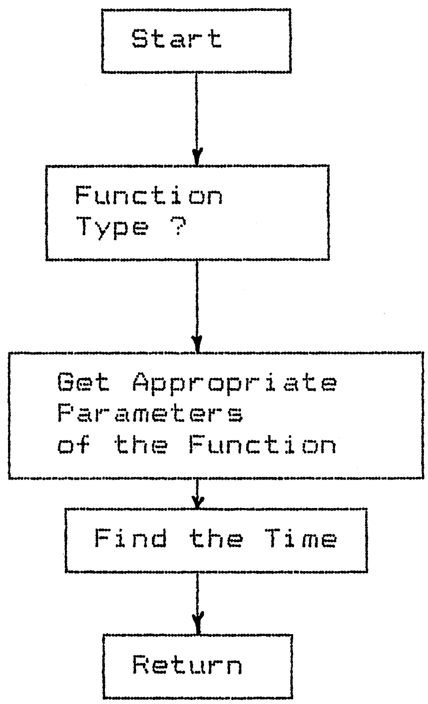
```
:ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NO
      .NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)     CHARL

 *....*...1.........2.........3.........4.........5.........6.........7.*.......8

 C======================================================================C
 C::::::                                                          ::::::C
 C......     SUBROUTINE OTPUT :                                  ......C
 C......     ----------------                                    ......C
 C......                      THIS SUBROUTINE IS TO GIVE OUTPUT ......C
 C......     OTHER THAN THE SLAM OUTPUT.                         ......C
 C::::::                                                          ::::::C
 C======================================================================C
       SUBROUTINE OTPUT
 C
 C=====>> INITIALIZATION OF VARIABLES.
 C
       COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
       COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
      $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
       CHARACTER DATE*8,PROJEC*12,DEPART*20,NAME*20,REFE*5
       COMMON/USER2/DATE,PROJEC,DEPART,NAME,REFE
       CHARACTER*11 STOR1,STOR2
       STOR1='           '
       STOR2='           '
 C
 C=====>> OUTPUT
 C
       WRITE(6,10)
  10   FORMAT('1',/////,36X,'MDMSS')
       WRITE(6,20)
  20   FORMAT(36X,'-----')
       WRITE(6,30)
  30   FORMAT(//,26X,'MENU-DRIVEN MANUFACTURING SYSTEM')
       WRITE(6,40)
  40   FORMAT(36X,'SIMULATOR')
       WRITE(6,60)
  60   FORMAT(///,34X,'VERSION  1.0')
       WRITE(6,70)
  70   FORMAT(//,31X,'RELEASED APRIL 1987')
       WRITE(6,80)
  80   FORMAT(//,33X,'DEVELOPED BY :')
       WRITE(6,90)
  90   FORMAT(33X,'IMED JAMOUSSI')
       WRITE(6,100)
  100  FORMAT(/,28X,'UNDER THE SUPERVISION OF :')
       WRITE(6,110)
  110  FORMAT(33X,'DR. JOE H. MIZE')
       WRITE(6,120)
  120  FORMAT(//,38X,'AT')
       WRITE(6,130)
  130  FORMAT(/,28X,'OKLAHOMA STATE UNIVERSITY')
       WRITE(6,140)
  140  FORMAT(23X,'DEPARTMENT OF INDUSTRIAL ENGINEERING')
       WRITE(6,150)
  150  FORMAT(///,21X,'FOR MORE INFORMATION CALL (405)-624-6055')
```

```
      WRITE(6,160)
  160 FORMAT('1',//////,31X,'GENERAL INFORMATION')
      WRITE(6,170)
  170 FORMAT(31X,'-------------------')
      WRITE(6,180) DATE
  180 FORMAT(///,11X,'DATE : ',A8)
      WRITE(6,190) PROJEC
  190 FORMAT(/,11X,'PROJECT # : ',A12)
      WRITE(6,200) REFE
  200 FORMAT(/,11X,'SIMULATION REFERENCE : ',A5)
      WRITE(6,210) DEPART
  210 FORMAT(/,11X,'DEPARTMENT : ',A20)
      WRITE(6,220) NAME
  220 FORMAT(/,11X,'OPERATOR NAME : ',A20)
      WRITE(6,230) LEVEL
  230 FORMAT(/,11X,'LEVEL OF SIMULATION : ',I1)
      WRITE(6,240) TIMED
  240 FORMAT(/,11X,'DURATION OF SIMULATION : ',F12.2)
      WRITE(6,250) MACH
  250 FORMAT(/,11X,'NUMBER OF FACILITIES : ',I2)
      WRITE(6,260) NPROD
  260 FORMAT(/,11X,'NUMBER OF PRODUCTS : ',I1)
C
C=====>> INFORMATION ABOUT THE PRODUCT.
C
      DO 1000 K=1,NPROD
         WRITE(6,270) K
  270    FORMAT('1',/,36X,'PRODUCT # ',I1)
            IF(ARVAL(K,1).EQ.1.) STOR1='CONSTANT'
            IF(ARVAL(K,1).EQ.2.) STOR1='NORMAL'
            IF(ARVAL(K,1).EQ.3.) STOR1='UNIFORM'
            IF(ARVAL(K,1).EQ.4.) STOR1='EXPONENTIAL'
            WRITE(6,280) STOR1
            WRITE(6,290) ARVAL(K,2),ARVAL(K,3)
             DO 950 KK=1,NMACHK(K)
              IF(ENTITY(K,KK,1).EQ.1.) STOR2='CONSTANT'
              IF(ENTITY(K,KK,1).EQ.2.) STOR2='NORMAL'
              IF(ENTITY(K,KK,1).EQ.3.) STOR2='UNIFORM'
              IF(ENTITY(K,KK,1).EQ.4.) STOR2='EXPONENTIAL'
  280          FORMAT(//,11X,'ARRIVAL DISTRIBUTION : ',A11)
  290          FORMAT(14X,'PARAMETER(S) : ',F10.4,4X,F10.4)
               WRITE(6,300) ENTITY(K,KK,2),KK
  300          FORMAT(//,11X,'GOES TO FACILITY ',F3.0,' IN THE ',I4,
     $              ' PLACE')
               WRITE(6,310) STOR2
  310          FORMAT(/,26X,'PROCESSING DISTRIBUTION : ',A11)
               WRITE(6,320) PARA(K,KK,1),PARA(K,KK,2)
  320          FORMAT(/,26X,'PARAMETER(S) : ',F10.4,4X,F10.4)
               GL=1-DEF(K,KK,1)-DEF(K,KK,2)
               WRITE(6,330) GL
  330          FORMAT(/,26X,'PERCENTAGE OF REWORK : ',F5.3)
               WRITE(6,340) DEF(K,KK,2)
  340          FORMAT(/,26X,'PERCENTAGE OF SCRAP : ',F5.3)
  950          CONTINUE
 1000     CONTINUE
       RETURN
```

```
AY 1985)            VS FORTRAN          DATE: APR 15, 1987    TIME: 14:03:18      N
 *....*...1.........2.........3.........4.........5.........6.........7.*.......8
       END                                                               00004670
SOURCE STATEMENTS = 89, PROGRAM SIZE = 5112 BYTES, PROGRAM NAME = OTPUT      PAGE
 NO DIAGNOSTICS GENERATED.
```

APPENDIX X

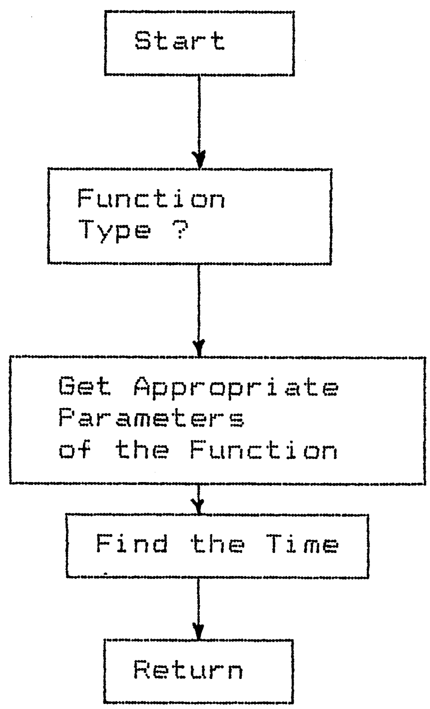
FLOWCHART AND LISTING OF FUNCTION TIM1


Start
Function
Type ?
Get Appropriate
Parameters
of the Function
Find the Time
Return

```
:ECT:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST !
       NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
         OPT(O) LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)      CHA

  *....*...1.........2.........3.........4.........5.........6.........7.*.......8

  C==================================================================C
  C::::::                                                      ::::::C
  C......                                                      ......C
  C......       FUNCTION TIM1 :                                ......C
  C......       -------------                                  ......C
  C......                     THIS FUNCTION FINDS THE RANDOM   ......C
  C......       PROCESSING TIMES PER MACHINE, PER PRODUCT.     ......C
  C::::::                                                      ::::::C
  C==================================================================C
        REAL FUNCTION TIM1(J,JJ)
  C
  C=====>> INITIALIZATION OF VARIABLES.
  C
        COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
       $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
        COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
       $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
  C
  C=====>> FIND THE NATURE OF THE FUNCTION AND THEN FIND THE RANDOM TIME
  C
        JJJ=ENTITY(J,JJ,1)
        GO TO (10,20,30,40),JJJ
  C:::::CONSTANT
   10   TIM1=PARA(J,JJ,1)
        RETURN
  C:::::NORMAL
   20   TIM1=RNORM(PARA(J,JJ,1),PARA(J,JJ,2),1)
        RETURN
  C::::::UNIFORM
   30   TIM1=UNFRM(PARA(J,JJ,1),PARA(J,JJ,2),1)
        RETURN
  C::::::EXPONENTIAL
   40   TIM1=EXPON(PARA(J,JJ,1),1)
        RETURN
        END

  SOURCE STATEMENTS =  14, PROGRAM SIZE =  1828 BYTES, PROGRAM NAME = TIM1      PAG

   NO DIAGNOSTICS GENERATED.

)F COMPILATION 7 ******
```

APPENDIX Y

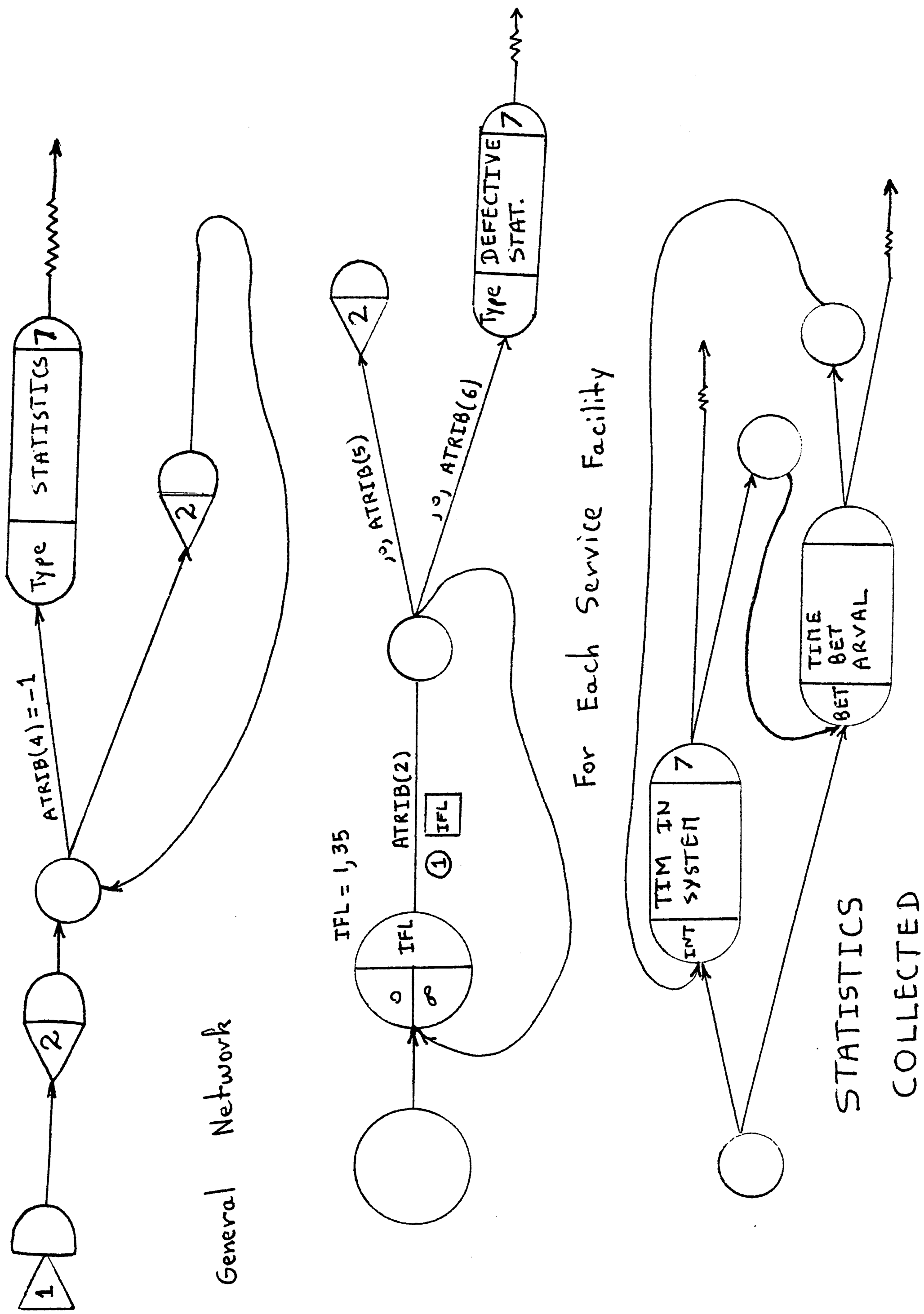
FLOWCHART AND LISTING OF FUNCTION TIM2


Start
Function Type ?
Get Appropriate Parameters of the Function
Find the Time
Return

```
:T:  NOLIST NOMAP NOXREF   GOSTMT NODECK   SOURCE NOTERM   OBJECT FIXED  NOTEST NO
     NOSYM NORENT   SDUMP AUTODBL(NONE)   NOSXM   IL
       OPT(O) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)      CHARL

*....*...1.........2.........3.........4.........5.........6.........7.*.......8

C=====================================================================C
C::::::                                                          ::::::C
C......      FUNCTION TIM2 :                                     ......C
C......      -------------                                       ......C
C......                    THIS FUNCTION IS FOR ARRIVAL RATES.   ......C
C::::::                                                          ::::::C
C=====================================================================C
      REAL FUNCTION TIM2(J)
C
C=====>> INITIALIZATION OF VARIABLES.
C
       COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     $,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USER1/LEVEL,NPROD,M(5),MACH,NMACHK(5),ENTITY(5,35,2),
     $PARA(5,35,3),ARVAL(5,3),DEF(5,35,2),TIMED
C
C=====>> DETERMINE THE NATURE OF THE RANDOM FUNCTION AND DETERMINE
C=====>> THE TIME.
C
      JJ=ARVAL(J,1)
      GO TO(10,20,30,40),JJ
C:::::::CONSTANT
 10   TIM2=ARVAL(J,2)
      RETURN
C:::::::NORMAL
 20   TIM2=RNORM(ARVAL(J,2),ARVAL(J,3),1)
      RETURN
C:::::::UNIFORM
 30   TIM2=UNFRM(ARVAL(J,2),ARVAL(J,3),1)
      RETURN
C:::::::EXPONENTIAL
 40   TIM2=EXPON(ARVAL(J,2),1)
C
      RETURN
      END

SOURCE STATEMENTS = 14, PROGRAM SIZE = 1652 BYTES, PROGRAM NAME = TIM2     PAGE:

 NO DIAGNOSTICS GENERATED.

 COMPILATION 8 ******
```

APPENDIX Z

FIGURE AND LISTING OF THE NETWORK


1
2
ATRIB(4) = -1
Type
STATISTICS
7
2
General Network
IFL = 1,35
IFL
0
∞
ATRIB(2)
① IFL
ATRIB(5)
ATRIB(6)
2
Type
DEFECTIVE STAT.
7
For Each Service Facility
INT
TIM IN SYSTEM
7
BET
TIME BET ARVAL
STATISTICS COLLECTED

```
GEN,IMED JAMOUSSI,IGPSSMP,4/12/1987,1;
LIMITS,35,10,200;
NETWORK;
;NETWORK START
;BRANCH TO THE APPROPRIATE WORKSTATION.
E1      ENTER,1,1;
E5      EVENT,2,1;
E6      GOON;
EY      GOON,1;
        ACT,,ATRIB(4).EQ.-1.,LNEW;
        ACT,,ATRIB(3).EQ.11.,M1;
        ACT,,ATRIB(3).EQ.12.,M2;
        ACT,,ATRIB(3).EQ.13.,M3;
        ACT,,ATRIB(3).EQ.14.,M4;
        ACT,,ATRIB(3).EQ.15.,M5;
        ACT,,ATRIB(3).EQ.16.,M6;
        ACT,,ATRIB(3).EQ.17.,M7;
        ACT,,ATRIB(3).EQ.21.,M8;
        ACT,,ATRIB(3).EQ.22.,M9;
        ACT,,ATRIB(3).EQ.23.,M10;
        ACT,,ATRIB(3).EQ.24.,M11;
        ACT,,ATRIB(3).EQ.25.,M12;
        ACT,,ATRIB(3).EQ.26.,M13;
        ACT,,ATRIB(3).EQ.27.,M14;
        ACT,,ATRIB(3).EQ.31.,M15;
        ACT,,ATRIB(3).EQ.32.,M16;
        ACT,,ATRIB(3).EQ.33.,M17;
        ACT,,ATRIB(3).EQ.34.,M18;
        ACT,,ATRIB(3).EQ.35.,M19;
        ACT,,ATRIB(3).EQ.36.,M20;
        ACT,,ATRIB(3).EQ.37.,M21;
        ACT,,ATRIB(3).EQ.41.,M22;
        ACT,,ATRIB(3).EQ.42.,M23;
        ACT,,ATRIB(3).EQ.43.,M24;
        ACT,,ATRIB(3).EQ.44.,M25;
        ACT,,ATRIB(3).EQ.45.,M26;
        ACT,,ATRIB(3).EQ.46.,M27;
        ACT,,ATRIB(3).EQ.47.,M28;
        ACT,,ATRIB(3).EQ.51.,M29;
        ACT,,ATRIB(3).EQ.52.,M30;
        ACT,,ATRIB(3).EQ.53.,M31;
        ACT,,ATRIB(3).EQ.54.,M32;
        ACT,,ATRIB(3).EQ.55.,M33;
        ACT,,ATRIB(3).EQ.56.,M34;
        ACT,,ATRIB(3).EQ.57.,M35;
;MACHINE 11
M1      GOON,1;
Q11     QUEUE(1);
        ACT/1,ATRIB(2);FACILITY 11;
        GOON;
        ACT,,ATRIB(5),E2;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q11;
E2      EVENT,2,1;
        ACT,,,EY;
;MACHINE 12
M2      GOON,1;
Q12     QUEUE(2);
        ACT/2,ATRIB(2);FACILITY 12;
        GOON;
        ACT,,ATRIB(5),E3;
        ACT,,ATRIB(6),DEF;
```

```
        ACT,,1-ATRIB(5)-ATRIB(6),Q12;
E3      EVENT,2,1;
        ACT,,,EY;
;MACHINE 13
M3      GOON,1;
Q13     QUEUE(3);
        ACT/3,ATRIB(2);FACILITY 13;
        GOON;
        ACT,,ATRIB(5),E36;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q13;
E36     EVENT,2,1;
        ACT,,,EY;
;MACHINE 14
M4      GOON,1;
Q14     QUEUE(4);
        ACT/4,ATRIB(2);FACILITY 14;
        GOON;
        ACT,,ATRIB(5),E4;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q14;
E4      EVENT,2;
        ACT,,,EY;
;MACHINE 15
M5      GOON,1;
Q15     QUEUE(5);
        ACT/5,ATRIB(2);FACILITY 15;
        GOON;
        ACT,,ATRIB(5),E55;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q15;
E55      EVENT,2;
        ACT,,,EY;
;MACHINE 16
M6      GOON,1;
Q16     QUEUE(6);
        ACT/6,ATRIB(2);FACILITY 16;
        GOON;
        ACT,,ATRIB(5),E66;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q16;
E66      EVENT,2;
        ACT,,,EY;
;MACHINE 17
M7      GOON,1;
Q17     QUEUE(7);
        ACT/7,ATRIB(2);FACILITY 17;
        GOON;
        ACT,,ATRIB(5),E7;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q17;
E7      EVENT,2;
        ACT,,,EY;
;MACHINE 21
M8      GOON,1;
Q21     QUEUE(8);
        ACT/8,ATRIB(2);FACILITY 21;
        GOON;
        ACT,,ATRIB(5),E8;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q21;
E8      EVENT,2;
        ACT,,,EY;
;MACHINE 22
M9      GOON,1;
Q22     QUEUE(9);
```

```
        ACT/9,ATRIB(2);FACILITY 22;
        GOON;
        ACT,,ATRIB(5),E9;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q22;
E9      EVENT,2;
        ACT,,,EY;
;MACHINE 23
M10     GOON,1;
Q23     QUEUE(10);
        ACT/10,ATRIB(2);FACILITY 23;
        GOON;
        ACT,,ATRIB(5),E10;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q23;
E10     EVENT,2;
        ACT,,,EY;
;MACHINE 24
M11     GOON,1;
Q24     QUEUE(11);
        ACT/11,ATRIB(2);FACILITY 24;
        GOON;
        ACT,,ATRIB(5),E11;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q24;
E11     EVENT,2;
        ACT,,,EY;
;MACHINE 25
M12     GOON,1;
Q25     QUEUE(12);
        ACT/12,ATRIB(2);FACILITY 25;
        GOON;
        ACT,,ATRIB(5),E12;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q25;
E12     EVENT,2;
        ACT,,,EY;
;MACHINE 26
M13     GOON,1;
Q26     QUEUE(13);
        ACT/13,ATRIB(2);FACILITY 26;
        GOON;
        ACT,,ATRIB(5),E13;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q26;
E13     EVENT,2;
        ACT,,,EY;
;MACHINE 27
M14     GOON,1;
Q27     QUEUE(14);
        ACT/14,ATRIB(2);FACILITY 27;
        GOON;
        ACT,,ATRIB(5),E14;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q27;
E14     EVENT,2;
        ACT,,,EY;
;MACHINE 31
M15     GOON,1;
Q31     QUEUE(15);
        ACT/15,ATRIB(2);FACILITY 31;
        GOON;
        ACT,,ATRIB(5),E15;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q31;
E15     EVENT,2;
```

```
       ACT,,,EY;
;MACHINE 32
M16    GOON,1;
Q32    QUEUE(16);
       ACT/16,ATRIB(2);FACILITY 32;
       GOON;
       ACT,,ATRIB(5),E16;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q32;
E16    EVENT,2;
       ACT,,,EY;
;MACHINE 33
M17    GOON,1;
Q33    QUEUE(17);
       ACT/17,ATRIB(2);FACILITY 33;
       GOON;
       ACT,,ATRIB(5),E17;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q33;
E17    EVENT,2;
       ACT,,,EY;
;MACHINE 34
M18    GOON,1;
Q34    QUEUE(18);
       ACT/18,ATRIB(2);FACILITY 34;
       GOON;
       ACT,,ATRIB(5),E18;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q34;
E18    EVENT,2;
       ACT,,,EY;
;MACHINE 35
M19    GOON,1;
Q35    QUEUE(19);
       ACT/19,ATRIB(2);FACILITY 35;
       GOON;
       ACT,,ATRIB(5),E19;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q35;
E19    EVENT,2;
       ACT,,,EY;
;MACHINE 36
M20    GOON,1;
Q36    QUEUE(20);
       ACT/20,ATRIB(2);FACILITY 36;
       GOON;
       ACT,,ATRIB(5),E20;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q36;
E20    EVENT,2;
       ACT,,,EY;
;MACHINE 37
M21    GOON,1;
Q37    QUEUE(21);
       ACT/21,ATRIB(2);FACILITY 37;
       GOON;
       ACT,,ATRIB(5),E21;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q37;
E21    EVENT,2;
       ACT,,,EY;
;MACHINE 41
M22    GOON,1;
Q41    QUEUE(22);
       ACT/22,ATRIB(2);FACILITY 41;
       GOON;
```

```
       ACT,,ATRIB(5),E22;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q41;
E22    EVENT,2;
       ACT,,,EY;
;MACHINE 42
M23    GOON,1;
Q42    QUEUE(23);
       ACT/23,ATRIB(2);FACILITY 42;
       GOON;
       ACT,,ATRIB(5),E23;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q42;
E23    EVENT,2;
       ACT,,,EY;
;MACHINE 43
M24    GOON,1;
Q43    QUEUE(24);
       ACT/24,ATRIB(2);FACILITY 43;
       GOON;
       ACT,,ATRIB(5),E24;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q43;
E24    EVENT,2;
       ACT,,,EY;
;MACHINE 44
M25    GOON,1;
Q44    QUEUE(25);
       ACT/25,ATRIB(2);FACILITY 44;
       GOON;
       ACT,,ATRIB(5),E25;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q44;
E25    EVENT,2;
       ACT,,,EY;
;MACHINE 45
M26    GOON,1;
Q45    QUEUE(26);
       ACT/26,ATRIB(2);FACILITY 45;
       GOON;
       ACT,,ATRIB(5),E26;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q45;
E26    EVENT,2;
       ACT,,,EY;
;MACHINE 46
M27    GOON,1;
Q46    QUEUE(27);
       ACT/27,ATRIB(2);FACILITY 46;
       GOON;
       ACT,,ATRIB(5),E27;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q46;
E27    EVENT,2;
       ACT,,,EY;
;MACHINE 47
M28    GOON,1;
Q47    QUEUE(28);
       ACT/28,ATRIB(2);FACILITY 47;
       GOON;
       ACT,,ATRIB(5),E28;
       ACT,,ATRIB(6),DEF;
       ACT,,1-ATRIB(5)-ATRIB(6),Q47;
E28    EVENT,2;
       ACT,,,EY;
;MACHINE 51
```

```
M29     GOON,1;
Q51     QUEUE(29);
        ACT/29,ATRIB(2);FACILITY 51;
        GOON;
        ACT,,ATRIB(5),E29;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q51;
E29     EVENT,2;
        ACT,,,EY;
;MACHINE 52
M30     GOON,1;
Q52     QUEUE(30);
        ACT/30,ATRIB(2);FACILITY 52;
        GOON;
        ACT,,ATRIB(5),E30;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q52;
E30     EVENT,2;
        ACT,,,EY;
;MACHINE 53
M31     GOON,1;
Q53     QUEUE(31);
        ACT/31,ATRIB(2);FACILITY 53;
        GOON;
        ACT,,ATRIB(5),E31;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q53;
E31     EVENT,2;
        ACT,,,EY;
;MACHINE 54;
M32     GOON,1;
Q54     QUEUE(32);
        ACT/32,ATRIB(2);FACILITY 54;
        GOON;
        ACT,,ATRIB(5),E32;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q54;
E32     EVENT,2;
        ACT,,,EY;
;MACHINE 55;
M33     GOON,1;
Q55     QUEUE(33);
        ACT/33,ATRIB(2);FACILITY 55;
        GOON;
        ACT,,ATRIB(5),E33;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q55;
E33     EVENT,2;
        ACT,,,EY;
;MACHINE 56;
M34     GOON,1;
Q56     QUEUE(34);
        ACT/34,ATRIB(2);FACILITY 56;
        GOON;
        ACT,,ATRIB(5),E34;
        ACT,,ATRIB(6),DEF;
        ACT,,1-ATRIB(5)-ATRIB(6),Q56;
E34     EVENT,2;
        ACT,,,EY;
;MACHINE 57
M35     GOON,1;
Q57     QUEUE(35);
        ACT/35,ATRIB(2);FACILITY 57;
        GOON;
        ACT,,ATRIB(5),E35;
        ACT,,ATRIB(6),DEF;
```

```
       ACT,,1-ATRIB(5)-ATRIB(6),Q57;
E35    EVENT,2;
       ACT,,,EY;
;PARTS DEFECTIVES. CAN NOT BE REWORKED.
DEF    GOON;
       ACT,,XX(1).NE.1.,DEF1;
       ACT,,XX(1).EQ.1.,D2;
D2     COLCT,INT(7),P? DEF TIME;
       GOON;
       ACT,,ATRIB(1).EQ.1.,DF1;
       ACT,,ATRIB(1).EQ.2.,DF2;
       ACT,,ATRIB(1).EQ.3.,DF3;
       ACT,,ATRIB(1).EQ.4.,DF4;
       ACT,,ATRIB(1).EQ.5.,DF5;
DF1    COLCT,INT(7),P1 DEF TIME;
       COLCT,BET,P1 BET DEF;
       ACT,,,DEND;
DF2    COLCT,INT(7),P2 DEF TIME;
       COLCT,BET,P2 BET DEF;
       ACT,,,DEND;
DF3    COLCT,INT(7),P3 DEF TIME;
       COLCT,BET,P3 BET DEF;
       ACT,,,DEND;
DF4    COLCT,INT(7),P4 DEF TIME;
       COLCT,BET,P4 BET DEF;
       ACT,,,DEND;
DF5    COLCT,INT(7),P5 DEF TIME;
       COLCT,BET,P5 BET DEF;
       ACT,,,DEND;
DEND   GOON;
DEF1    TERM;
;GET THE PRODUCT OUT OF THE SIMULATION
LNEW   GOON,1; NO MORE PROCESSING
       ACT,,XX(3).NE.1.,LA2;
       ACT,,XX(3).EQ.1.,D1;
;THIS SECTION OF THE NETWORK IS FOR STATISTICS COLLECTION.
;STATISTICS ABOUT THE TIME IN THE SYSTEM
D1     COLCT,INT(7),T SYSTEM TIME;
       ACT,,ATRIB(1).EQ.1.,SP1;
       ACT,,ATRIB(1).EQ.2.,SP2;
       ACT,,ATRIB(1).EQ.3.,SP3;
       ACT,,ATRIB(1).EQ.4.,SP4;
       ACT,,ATRIB(1).EQ.5.,SP5;
;PRODUCT 1 SYSTEM TIME
SP1    COLCT,INT(7),P1 SYSTEM TIME;
       ACT,,,TEND;
;PRODUCT 2 SYSTEM TIME
SP2    COLCT,INT(7),P2 SYSTEM TIME;
       ACT,,,TEND;
;PRODUCT 3 SYSTEM TIME
SP3    COLCT,INT(7),P3 SYSTEM TIME;
       ACT,,,TEND;
;PRODUCT 4 SYSTEM TIME
SP4    COLCT,INT(7),P4 SYSTEM TIME;
       ACT,,,TEND;
;PRODUCT 5 SYSTEM TIME
SP5    COLCT,INT(7),P5 SYSTEM TIME;
       ACT,,,TEND;
TEND   GOON,1;
LA2    GOON;
       ACT,,XX(2).NE.1.,LAS1;
       ACT,,XX(2).EQ.1.,D3;
D3     COLCT,BET,P? BET COMP;
       ACT,,ATRIB(1).EQ.1.,BE1;
       ACT,,ATRIB(1).EQ.2.,BE2;
       ACT,,ATRIB(1).EQ.3.,BE3;
```

```
       ACT,,ATRIB(1).EQ.4.,BE4;
       ACT,,ATRIB(1).EQ.5.,BE5;
BE1    COLCT,BET,P1 BET COMP;
       ACT,,,LAS1;
BE2    COLCT,BET,P2 BET COMP;
       ACT,,,LAS1;
BE3    COLCT,BET,P3 BET COMP;
       ACT,,,LAS1;
BE4    COLCT,BET,P4 BET COMP;
       ACT,,,LAS1;
BE5    COLCT,BET,P5 BET COMP;
       ACT,,,LAS1;
LAS1   GOON;
LAST   TERM;
       END;
C
;TIME OF SIMULATION
FIN;
```

APPENDIX A1

JCL FOR THE SIMULATION PROGRAM

JCL FOR THE SIMULATION PROGRAM

The JCL shown in next page is for the SIMULATION program. The cards are self explanotory. In the last one FT03F001 designate the file to read from. This file was created and data was put in through the INPUT program.

```
//U11296AH JOB (XXXXX,                                                      JOB 3691
// 447-78-6302),'THESIS',TIME=(0,40),CLASS=A,
// MSGCLASS=E
***PASSWORD                                                                  0000003
***JOBPARM ROOM=I,FORMS=9001                                                 0000004
// EXEC SLAMCLG,REGION=5500K                                                00000050
//FORT.SYSIN DD *                                                           00000060
//SLAM.SYSIN DD *                                                           00004690
//FT03F001    DD    DSN=U11296A.FT03F001.CNTL,DISP=SHR                      00009450
```

APPENDIX B1

EXAMPLE 1

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

GENERAL INFORMATION

DATE : 03-08-87

PROJECT # : TEST 1

SIMULATION REFERENCE : 1

DEPARTMENT : PRODUCTION

OPERATOR NAME : IMED JAMOUSSI

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 100.00

NUMBER OF FACILITIES : 2

NUMBER OF PRODUCTS : 1

PRODUCT # 1

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 10.0000 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 12.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 2 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 14.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT IGPSSMP BY IMED JAMOUSSI

DATE 4/12/1987 RUN NUMBER 1 OF 1

CURRENT TIME 0.1000E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P? DEF TIME		NO VALUES RECORDED				
P1 DEF TIME		NO VALUES RECORDED				
P1 BET DEF		NO VALUES RECORDED				
P2 DEF TIME		NO VALUES RECORDED				
P2 BET DEF		NO VALUES RECORDED				
P3 DEF TIME		NO VALUES RECORDED				
P3 BET DEF		NO VALUES RECORDED				
P4 DEF TIME		NO VALUES RECORDED				
P4 BET DEF		NO VALUES RECORDED				
P5 DEF TIME		NO VALUES RECORDED				
P5 BET DEF		NO VALUES RECORDED				
T SYSTEM TIME	0.3400E+02	0.6325E+01	0.1860E+00	0.2600E+02	0.4200E+02	5
P1 SYSTEM TIME	0.3400E+02	0.6325E+01	0.1860E+00	0.2600E+02	0.4200E+02	5
P2 SYSTEM TIME		NO VALUES RECORDED				
P3 SYSTEM TIME		NO VALUES RECORDED				
P4 SYSTEM TIME		NO VALUES RECORDED				
P5 SYSTEM TIME		NO VALUES RECORDED				
P? BET COMP	0.1400E+02	0.0000E+00	0.0000E+00	0.1400E+02	0.1400E+02	4
P1 BET COMP	0.1400E+02	0.0000E+00	0.0000E+00	0.1400E+02	0.1400E+02	4
P2 BET COMP		NO VALUES RECORDED				
P3 BET COMP		NO VALUES RECORDED				
P4 BET COMP		NO VALUES RECORDED				
P5 BET COMP		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE		AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11	QUEUE	0.6600	0.5869	2	1	7.3333
2	Q12	QUEUE	0.3600	0.4800	1	1	5.1429
3	Q13	QUEUE	0.0000	0.0000	0	0	0.0000
4	Q14	QUEUE	0.0000	0.0000	0	0	0.0000
5	Q15	QUEUE	0.0000	0.0000	0	0	0.0000
6	Q16	QUEUE	0.0000	0.0000	0	0	0.0000
7	Q17	QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21	QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22	QUEUE	0.0000	0.0000	0	0	0.0000
10	Q23	QUEUE	0.0000	0.0000	0	0	0.0000
11	Q24	QUEUE	0.0000	0.0000	0	0	0.0000
12	Q25	QUEUE	0.0000	0.0000	0	0	0.0000

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	3.1800	0.5896	5	4	2.1781

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
1	FACILITY 11	1	0.9000	0.3000	1	0.0000	10.0000	90.0000	7
2	FACILITY 12	1	0.7800	0.4142	1	0.0000	22.0000	78.0000	5
3	FACILITY 13	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
4	FACILITY 14	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
5	FACILITY 15	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
6	FACILITY 16	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
8	FACILITY 21	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
9	FACILITY 22	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
10	FACILITY 23	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0

APPENDIX C1

TRACE 1

SLAM II TRACE BEGINNING AT TNOW= 0.0000E+00

TNOW	JEVNT	NODE ARRIVAL LABEL	TYPE	CURRENT VARIABLE BUFFER					ACTIVITY SUMMARY INDEX	DURATION	END NODE
0.1000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		011	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									1	12.0000	
0.2000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		011	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
0.2200E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									1	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	011
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			

		Q12	QUEUE	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									2	14.0000	
0.3000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
0.3400E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									1	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
		Q12	QUEUE	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
0.3600E+02			GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									2	14.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q12
									0	0.0000	E3
		E3	EVENT	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	0.0000	LNEW
		LNEW	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LA2
									0	0.0000	D1
		D1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	NOT RELEASED	SP2
									0	NOT RELEASED	SP3
									0	NOT RELEASED	SP4
									0	NOT RELEASED	SP5
									0	0.0000	SP1
		SP1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	0.0000	TEND
		TEND	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			

		LA2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LAS1
									0	0.0000	D3
		D3	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	NOT RELEASED	BE2
									0	NOT RELEASED	BE3
									0	NOT RELEASED	BE4
									0	NOT RELEASED	BE5
									0	0.0000	BE1
		BE1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
									0	0.0000	LAS1
		LAS1	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
		LAST	TERM	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.1000E+01			
0.4000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
0.4600E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									1	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
		Q12	QUEUE	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.1000E+01			
0.5000E+02		END OF TRACE									

APPENDIX D1

EXAMPLE 2

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

GENERAL INFORMATION

DATE : 03-08-87

PROJECT # : TEST 2

SIMULATION REFERENCE : 2

DEPARTMENT : PRODUCTION

OPERATOR NAME : IMED JAMOUSSI

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 100.00

NUMBER OF FACILITIES : 2

NUMBER OF PRODUCTS : 1

PRODUCT # 1

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 10.0000 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 12.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 2 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 14.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT IGPSSMP   BY IMED JAMOUSSI

DATE 4/12/1987   RUN NUMBER   1 OF   1

CURRENT TIME   0.1000E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P? DEF TIME		NO VALUES RECORDED				
P1 DEF TIME		NO VALUES RECORDED				
P1 BET DEF		NO VALUES RECORDED				
P2 DEF TIME		NO VALUES RECORDED				
P2 BET DEF		NO VALUES RECORDED				
P3 DEF TIME		NO VALUES RECORDED				
P3 BET DEF		NO VALUES RECORDED				
P4 DEF TIME		NO VALUES RECORDED				
P4 BET DEF		NO VALUES RECORDED				
P5 DEF TIME		NO VALUES RECORDED				
P5 BET DEF		NO VALUES RECORDED				
T SYSTEM TIME		NO VALUES RECORDED				
P1 SYSTEM TIME		NO VALUES RECORDED				
P2 SYSTEM TIME		NO VALUES RECORDED				
P3 SYSTEM TIME		NO VALUES RECORDED				
P4 SYSTEM TIME		NO VALUES RECORDED				
P5 SYSTEM TIME		NO VALUES RECORDED				
P? BET COMP		NO VALUES RECORDED				
P1 BET COMP		NO VALUES RECORDED				
P2 BET COMP		NO VALUES RECORDED				
P3 BET COMP		NO VALUES RECORDED				
P4 BET COMP		NO VALUES RECORDED				
P5 BET COMP		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11 QUEUE	0.6600	0.5869	2	1	7.3333
2	Q12 QUEUE	0.3600	0.4800	1	1	5.1429
3	Q13 QUEUE	0.0000	0.0000	0	0	0.0000
4	Q14 QUEUE	0.0000	0.0000	0	0	0.0000
5	Q15 QUEUE	0.0000	0.0000	0	0	0.0000
6	Q16 QUEUE	0.0000	0.0000	0	0	0.0000
7	Q17 QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21 QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22 QUEUE	0.0000	0.0000	0	0	0.0000
10	Q23 QUEUE	0.0000	0.0000	0	0	0.0000
11	Q24 QUEUE	0.0000	0.0000	0	0	0.0000
[illegible]	[illegible]	0.0000	0.0000	0	0	0.0000

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	3.1800	0.5896	5	4	2.5238

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
1	FACILITY 11	1	0.9000	0.3000	1	0.0000	10.0000	90.0000	7
2	FACILITY 12	1	0.7800	0.4142	1	0.0000	22.0000	78.0000	5
3	FACILITY 13	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
4	FACILITY 14	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
5	FACILITY 15	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
6	FACILITY 16	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
8	FACILITY 21	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
9	FACILITY 22	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
10	FACILITY 23	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0

APPENDIX E1

TRACE 2

SLAM II TRACE BEGINNING AT TNOW= 0.0000E+00

TNOW	JEVNT	NODE ARRIVAL LABEL	NODE ARRIVAL TYPE	CURRENT VARIABLE BUFFER					ACTIVITY SUMMARY INDEX	ACTIVITY SUMMARY DURATION	ACTIVITY SUMMARY END NODE
0.1000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									1	12.0000	
0.2000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
0.2200E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									1	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			

		Q12	QUEUE	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									2	14.0000	
0.3000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
0.3400E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									1	12.0000	
									0	NOT RELEASED	E2
									0	NOT RELEASED	Q11
									0	0.0000	DEF
		DEF	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	DEF1
									0	0.0000	D2
		D2	COLCT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	DF2
									0	NOT RELEASED	DF3
									0	NOT RELEASED	DF4
									0	NOT RELEASED	DF5
									0	0.0000	DF1
		DF1	COLCT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
			COLCT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	0.0000	DEND
		DEND	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		DEF1	TERM	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
0.3600E+02			GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q12
									0	0.0000	E3
		E3	EVENT	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	0.0000	LNEW
		LNEW	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	NOT RELEASED	LA2
									0	0.0000	D1
		D1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	NOT RELEASED	SP2

									0	NOT RELEASED	SP2
									0	NOT RELEASED	SP4
									0	NOT RELEASED	SP5
									0	0.0000	SP1
		SP1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	0.0000	TEND
		TEND	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
		LA2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	NOT RELEASED	LAS1
									0	0.0000	D3
		D3	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	NOT RELEASED	BE2
									0	NOT RELEASED	BE3
									0	NOT RELEASED	BE4
									0	NOT RELEASED	BE5
									0	0.0000	BE1
		BE1	COLCT	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
									0	0.0000	LAS1
		LAS1	GOON	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
		LAST	TERM	0.1000E+01	0.1400E+02	0.1200E+02	-0.1000E+01	0.8000E+00			
0.4000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
0.4600E+02			GOON	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									1	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1200E+02	0.1100E+02	0.1000E+01	0.9000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
		Q12	QUEUE	0.1000E+01	0.1400E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									2	14.0000	
0.5000E+02		END OF TRACE									

APPENDIX F1

EXAMPLE 3

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

GENERAL INFORMATION

DATE : 03-08-87

PROJECT # : RUN 3

SIMULATION REFERENCE : 3

DEPARTMENT : PRODUCTION

OPERATOR NAME : IMED JAMOUSSI

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 100.00

NUMBER OF FACILITIES : 2

NUMBER OF PRODUCTS : 1

PRODUCT # 1

```
ARRIVAL DISTRIBUTION : CONSTANT
   PARAMETER(S) :     10.0000          0.0000

GOES TO FACILITY 11. IN THE     1 PLACE
                 PROCESSING DISTRIBUTION : CONSTANT
                 PARAMETER(S) :     12.0000          0.0000
                 PERCENTAGE OF REWORK : 0.050
                 PERCENTAGE OF SCRAP : 0.050

GOES TO FACILITY 12. IN THE     2 PLACE
                 PROCESSING DISTRIBUTION : CONSTANT
                 PARAMETER(S) :     14.0000          0.0000
                 PERCENTAGE OF REWORK : 0.100
                 PERCENTAGE OF SCRAP : 0.100
```

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT IGPSSMP   BY IMED JAMOUSSI

DATE 4/12/1987   RUN NUMBER 1 OF

CURRENT TIME 0.1000E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P? DEF TIME	0.1400E+02	0.0000E+00	0.0000E+00	0.1400E+02	0.1400E+02	1
P1 DEF TIME	0.1400E+02	0.0000E+00	0.0000E+00	0.1400E+02	0.1400E+02	1
P1 BET DEF		NO VALUES	RECORDED			
P2 DEF TIME		NO VALUES	RECORDED			
P2 BET DEF		NO VALUES	RECORDED			
P3 DEF TIME		NO VALUES	RECORDED			
P3 BET DEF		NO VALUES	RECORDED			
P4 DEF TIME		NO VALUES	RECORDED			
P4 BET DEF		NO VALUES	RECORDED			
P5 DEF TIME		NO VALUES	RECORDED			
P5 BET DEF		NO VALUES	RECORDED			
T SYSTEM TIME	0.3133E+02	0.6110E+01	0.1950E+00	0.2600E+02	0.3800E+02	3
P1 SYSTEM TIME	0.3133E+02	0.6110E+01	0.1950E+00	0.2600E+02	0.3800E+02	3
P2 SYSTEM TIME		NO VALUES	RECORDED			
P3 SYSTEM TIME		NO VALUES	RECORDED			
P4 SYSTEM TIME		NO VALUES	RECORDED			
P5 SYSTEM TIME		NO VALUES	RECORDED			
P? BET COMP	0.2600E+02	0.2828E+01	0.1088E+00	0.2400E+02	0.2800E+02	2
P1 BET COMP	0.2600E+02	0.2828E+01	0.1088E+00	0.2400E+02	0.2800E+02	2
P2 BET COMP		NO VALUES	RECORDED			
P3 BET COMP		NO VALUES	RECORDED			
P4 BET COMP		NO VALUES	RECORDED			
P5 BET COMP		NO VALUES	RECORDED			

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE		AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11	QUEUE	0.6600	0.5869	2	1	7.3333
2	Q12	QUEUE	0.4400	0.6974	2	2	6.2857
3	Q13	QUEUE	0.0000	0.0000	0	0	0.0000
4	Q14	QUEUE	0.0000	0.0000	0	0	0.0000
5	Q15	QUEUE	0.0000	0.0000	0	0	0.0000
6	Q16	QUEUE	0.0000	0.0000	0	0	0.0000
7	Q17	QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21	QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22	QUEUE	0.0000	0.0000	0	0	0.0000
10	Q23	QUEUE	0.0000	0.0000	0	0	0.0000
11	Q24	QUEUE	0.0000	0.0000	0	0	0.0000
12	Q25	QUEUE	0.0000	0.0000	0	0	0.0000

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	3.0800	0.5231	5	4	2.4839

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTIT COUNT
1	FACILITY 11	1	0.9000	0.3000	1	0.0000	10.0000	90.0000	7
2	FACILITY 12	1	0.6800	0.4665	1	0.0000	22.0000	54.0000	4
3	FACILITY 13	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
4	FACILITY 14	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
5	FACILITY 15	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
6	FACILITY 16	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
8	FACILITY 21	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
9	FACILITY 22	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
10	FACILITY 23	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0

APPENDIX G1

TRACE 3

SLAM II TRACE BEGINNING AT TNOW= 0.0000E+00

TNOW	JEVNT	NODE ARRIVAL LABEL	NODE ARRIVAL TYPE	CURRENT VARIABLE BUFFER					ACTIVITY SUMMARY INDEX	DURATION	END NODE
0.1000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		Q11	QUEUE	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									1	1.0000	
0.1100E+02			GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
		Q12	QUEUE	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									2	12.0000	
0.1200E+02	1			0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			

		Q11	QUEUE	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									1	8.0000	
0.2000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		Q11	QUEUE	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
			GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									1	1.0000	
									0	NOT RELEASED	E2
									0	NOT RELEASED	Q11
									0	0.0000	DEF
		DEF	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	DEF1
									0	0.0000	D2
		D2	COLCT	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
			GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	DF1
									0	NOT RELEASED	DF3
									0	NOT RELEASED	DF4
									0	NOT RELEASED	DF5
									0	0.0000	DF2
		DF2	COLCT	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
			COLCT	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	0.0000	DEND
		DEND	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		DEF1	TERM	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
0.2100E+02			GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
		Q12	QUEUE	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
0.2300E+02			GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			

									2	12.0000	
									0	NOT RELEASED	E3
									0	NOT RELEASED	DEF
									0	0.0000	Q12
		Q12	QUEUE	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
0.2400E+02	1			0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									1	8.0000	
0.3000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		Q11	QUEUE	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
0.3200E+02			GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									1	1.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	0.0000	EY
		EY	GOON	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	NOT RELEASED	M2
									0	NOT RELEASED	M3
									0	NOT RELEASED	M4
									0	NOT RELEASED	M5
									0	NOT RELEASED	M6
									0	NOT RELEASED	M7
									0	0.0000	M8
		M8	GOON	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
		Q21	QUEUE	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
									8	10.0000	
0.3300E+02			GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			

									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
		Q12	QUEUE	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
0.3500E+02			GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									2	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q12
									0	0.0000	E3
		E3	EVENT	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	NOT RELEASED	M2
									0	0.0000	M3
		M3	GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
		Q13	QUEUE	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
									3	5.0000	
0.3600E+02	1			0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		Q11	QUEUE	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									1	8.0000	
0.4000E+02	1			0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.1000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		EY	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			

	Q11	QUEUE	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
		GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
								0	NOT RELEASED	E36
								0	NOT RELEASED	DEF
								0	0.0000	Q13
	Q13	QUEUE	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
								3	5.0000	
0.4200E+02		GOON	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
								0	NOT RELEASED	DEF
								0	NOT RELEASED	Q21
								0	0.0000	E8
	E8	EVENT	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
								0	0.0000	EY
	EY	GOON	0.2000E+01	0.6000E+01	0.2200E+02	0.3000E+01	0.8000E+00			
								0	NOT RELEASED	LNEW
								0	NOT RELEASED	M1
								0	NOT RELEASED	M2
								0	NOT RELEASED	M3
								0	NOT RELEASED	M4
								0	NOT RELEASED	M5
								0	NOT RELEASED	M6
								0	NOT RELEASED	M7
								0	NOT RELEASED	M8
								0	0.0000	M9
	M9	GOON	0.2000E+01	0.6000E+01	0.2200E+02	0.3000E+01	0.8000E+00			
	Q22	QUEUE	0.2000E+01	0.6000E+01	0.2200E+02	0.3000E+01	0.8000E+00			
								9	6.0000	
0.4400E+02		GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
								1	1.0000	
								0	NOT RELEASED	DEF
								0	NOT RELEASED	Q11
								0	0.0000	E2
	E2	EVENT	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
								0	0.0000	EY
	EY	GOON	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
								0	NOT RELEASED	LNEW
								0	NOT RELEASED	M1
								0	NOT RELEASED	M2
								0	NOT RELEASED	M3
								0	NOT RELEASED	M4
								0	NOT RELEASED	M5
								0	NOT RELEASED	M6
								0	NOT RELEASED	M7
								0	0.0000	M8
	M8	GOON	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
	Q21	QUEUE	0.2000E+01	0.1000E+02	0.2100E+02	0.2000E+01	0.1000E+01			
								8	10.0000	
0.4500E+02		GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
								0	NOT RELEASED	DEF
								0	NOT RELEASED	Q13
								0	0.0000	E36

		E36	EVENT	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.3000E+01	0.1400E+02	0.4000E+01	0.1000E+01			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	NOT RELEASED	M2
									0	NOT RELEASED	M3
									0	0.0000	M4
		M4	GOON	0.1000E+01	0.3000E+01	0.1400E+02	0.4000E+01	0.1000E+01			
		Q14	QUEUE	0.1000E+01	0.3000E+01	0.1400E+02	0.4000E+01	0.1000E+01			
									4	3.0000	
			GOON	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q11
									0	0.0000	E2
		E2	EVENT	0.1000E+01	0.1000E+01	0.1100E+02	0.1000E+01	0.1000E+01			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	0.0000	M2
		M2	GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
		Q12	QUEUE	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
0.4700E+02			GOON	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									2	12.0000	
									0	NOT RELEASED	DEF
									0	NOT RELEASED	Q12
									0	0.0000	E3
		E3	EVENT	0.1000E+01	0.1200E+02	0.1200E+02	0.2000E+01	0.8000E+00			
									0	0.0000	EY
		EY	GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	NOT RELEASED	M1
									0	NOT RELEASED	M2
									0	0.0000	M3
		M3	GOON	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
		Q13	QUEUE	0.1000E+01	0.5000E+01	0.1300E+02	0.3000E+01	0.9000E+00			
									3	5.0000	
0.4800E+02	1			0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E1	ENTER	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E5	EVENT	0.2000E+01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00			
		E6	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
		EY	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			
									0	NOT RELEASED	LNEW
									0	0.0000	M1
		M1	GOON	0.2000E+01	0.8000E+01	0.1100E+02	0.1000E+01	0.9000E+00			

```
Q11  QUEUE     0.2000E+01,  0.8000E+01  0.1100E+02   0.1000E+01  0.9000E+00
                                                                                1        8.0000
     GOON      0.2000E+01   0.6000E+01  0.2200E+02   0.3000E+01  0.8000E+00
                                                                                0  NOT RELEASED   DEF
                                                                                0  NOT RELEASED   Q22
                                                                                0        0.0000   E9
E9   EVENT     0.2000E+01   0.6000E+01  0.2200E+02   0.3000E+01  0.8000E+00
                                                                                0        0.0000   EY
EY   GOON      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0        0.0000   LNEW
LNEW GOON      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0  NOT RELEASED   LA2
                                                                                0        0.0000   D1
D1   COLCT     0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0  NOT RELEASED   SP1
                                                                                0  NOT RELEASED   SP3
                                                                                0  NOT RELEASED   SP4
                                                                                0  NOT RELEASED   SP5
                                                                                0        0.0000   SP2
SP2  COLCT     0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0        0.0000   TEND
TEND GOON      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
LA2  GOON      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0  NOT RELEASED   LAS1
                                                                                0        0.0000   D3
D3   COLCT     0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0  NOT RELEASED   BE1
                                                                                0  NOT RELEASED   BE3
                                                                                0  NOT RELEASED   BE4
                                                                                0  NOT RELEASED   BE5
                                                                                0        0.0000   BE2
BE2  COLCT     0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
                                                                                0        0.0000   LAS1
LAS1 GOON      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
LAST TERM      0.2000E+01   0.6000E+01  0.2200E+02  -0.1000E+01  0.8000E+00
     GOON      0.1000E+01   0.3000E+01  0.1400E+02   0.4000E+01  0.1000E+01
                                                                                0  NOT RELEASED   DEF
                                                                                0  NOT RELEASED   Q14
                                                                                0        0.0000   E4
E4   EVENT     0.1000E+01   0.3000E+01  0.1400E+02   0.4000E+01  0.1000E+01
                                                                                0        0.0000   EY
EY   GOON      0.1000E+01   0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01
                                                                                0        0.0000   LNEW
LNEW GOON      0.1000E+01   0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01
                                                                                0  NOT RELEASED   LA2
                                                                                0        0.0000   D1
D1   COLCT     0.1000E+01   0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01
                                                                                0  NOT RELEASED   SP?
```

```
                                                                                    0  NOT RELEASED     SP3
                                                                                    0  NOT RELEASED     SP4
                                                                                    0  NOT RELEASED     SP5
                                                                                    0        0.0000     SP1
              SP1  COLCT     0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

                                                                                    0        0.0000     TEND
              TEND GOON      0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

              LA2  GOON      0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

                                                                                    0  NOT RELEASED     LAS1
                                                                                    0        0.0000     D3
              D3   COLCT     0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

                                                                                    0  NOT RELEASED     BE2
                                                                                    0  NOT RELEASED     BE3
                                                                                    0  NOT RELEASED     BE4
                                                                                    0  NOT RELEASED     BE5
                                                                                    0        0.0000     BE1
              BE1  COLCT     0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

                                                                                    0        0.0000     LAS1
              LAS1 GOON      0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

              LAST TERM      0.1000E+01  0.3000E+01  0.1400E+02  -0.1000E+01  0.1000E+01

0.5000E+02     END OF TRACE
```

APPENDIX H1

EXAMPLE 4

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

GENERAL INFORMATION

DATE : 03-08-87

PROJECT # : TEST 4

SIMULATION REFERENCE : 4

DEPARTMENT : PRODUCTION

OPERATOR NAME : IMED JAMOUSSI

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 100.00

NUMBER OF FACILITIES : 8

NUMBER OF PRODUCTS : 2

PRODUCT # 1

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 10.0000 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 1.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 2 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 12.0000 0.0000

PERCENTAGE OF REWORK : 0.200

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 13. IN THE 3 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 5.0000 0.0000

PERCENTAGE OF REWORK : 0.090

PERCENTAGE OF SCRAP : 0.010

GOES TO FACILITY 14. IN THE 4 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 3.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

PRODUCT # 2

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 12.0000 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 8.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.100

GOES TO FACILITY 21. IN THE 2 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 10.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 22. IN THE 3 PLACE

PROCESSING DISTRIBUTION : CONSTANT

PARAMETER(S) : 6.0000 0.0000

PERCENTAGE OF REWORK : 0.100

PERCENTAGE OF SCRAP : 0.100

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT IGPSSMP   BY IMED JAMOUSSI

DATE 4/12/1987   RUN NUMBER 1 OF 1

CURRENT TIME 0.1000E+03
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
PT DEF TIME	0.8000E+01	0.0000E+00	0.0000E+00	0.8000E+01	0.8000E+01	2
P1 DEF TIME		NO VALUES RECORDED				
P1 BET DEF		NO VALUES RECORDED				
P2 DEF TIME	0.8000E+01	0.0000E+00	0.0000E+00	0.8000E+01	0.8000E+01	2
P2 BET DEF	0.4800E+02	0.0000E+00	0.0000E+00	0.4800E+02	0.4800E+02	1
P3 DEF TIME		NO VALUES RECORDED				
P3 BET DEF		NO VALUES RECORDED				
P4 DEF TIME		NO VALUES RECORDED				
P4 BET DEF		NO VALUES RECORDED				
P5 DEF TIME		NO VALUES RECORDED				
P5 BET DEF		NO VALUES RECORDED				
T SYSTEM TIME	0.3178E+02	0.8628E+01	0.2715E+00	0.2400E+02	0.4500E+02	9
P1 SYSTEM TIME	0.3800E+02	0.6325E+01	0.1664E+00	0.2800E+02	0.4500E+02	5
P2 SYSTEM TIME	0.2400E+02	0.0000E+00	0.0000E+00	0.2400E+02	0.2400E+02	4
P3 SYSTEM TIME		NO VALUES RECORDED				
P4 SYSTEM TIME		NO VALUES RECORDED				
P5 SYSTEM TIME		NO VALUES RECORDED				
PT BET COMP	0.6000E+01	0.3338E+01	0.5563E+00	0.0000E+00	0.1200E+02	8
P1 BET COMP	0.1075E+02	0.2500E+01	0.2326E+00	0.7000E+01	0.1200E+02	4
P2 BET COMP	0.1600E+02	0.6928E+01	0.4330E+00	0.1200E+02	0.2400E+02	3
P3 BET COMP		NO VALUES RECORDED				
P4 BET COMP		NO VALUES RECORDED				
P5 BET COMP		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE		AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11	QUEUE	0.2200	0.4142	1	0	1.2941
2	Q12	QUEUE	1.2500	0.8874	3	3	11.3636
3	Q13	QUEUE	0.0000	0.0000	0	0	0.0000
4	Q14	QUEUE	0.0000	0.0000	0	0	0.0000
5	Q15	QUEUE	0.0000	0.0000	0	0	0.0000
6	Q16	QUEUE	0.0000	0.0000	0	0	0.0000
7	Q17	QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21	QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22	QUEUE	0.0000	0.0000	0	0	0.0000
10	Q23	QUEUE	0.0000	0.0000	0	0	0.0000
11	Q24	QUEUE	0.0000	0.0000	0	0	0.0000
12	Q25	QUEUE	0.0000	0.0000	0	0	0.0000

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	5.2500	1.1522	9	6	1.5130

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
1	FACILITY 11	1	0.6900	0.4625	1	0.0000	10.0000	9.0000	16
2	FACILITY 12	1	0.8900	0.3129	1	0.0000	11.0000	89.0000	7
3	FACILITY 13	1	0.3000	0.4583	0	0.0000	35.0000	5.0000	6
4	FACILITY 14	1	0.1500	0.3571	0	0.0000	45.0000	3.0000	5
5	FACILITY 15	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
6	FACILITY 16	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
8	FACILITY 21	1	0.4800	0.4996	1	0.0000	32.0000	10.0000	4
9	FACILITY 22	1	0.2400	0.4271	0	0.0000	42.0000	6.0000	4
10	FACILITY 23	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	100.0000	0.0000	0

APPENDIX I1

SCHRIBER PRODUCTION SHOP EXAMPLE

STATEMENT OF THE PROBLEM

A production shop is comprised of six different groups of machines. Each group consists of a number of identical machines of a given kind as indicated below.

Machine Group Number	Kind of Machines in Group	Number of Machines in Group
1	Casting units	10
2	Lathes	4
3	Planers	3
4	Drill presses	4
5	Shapers	12
6	Polishing machines	3

Three different types of jobs move through the production shop. These job-types are designated as Type 1, Type 2, and Type 3. Each job-type requires that operations be performed at specified kinds of machines in a specific sequence. All operation times are exponentially distributed. The visitation sequences and average operation times are shown in the next page.

Visitation Sequences and Mean Operation Times for the Three Types of Jobs

Job Type	Total Number of Machines to be Visited	Machine Visitation Sequence	Mean Operation Time(Minutes)
		Casting Unit	125
		Planer	35
1	4	Lathe	20
		Polishing machine	60
		Shaper	105
2	3	Drill Press	90
		Lathe	65
		Casting Unit	235
		Shaper	250
3	5	Drill press	50
		Planer	30
		Polishing machine	25

Jobs arrive at the shop with exponential interarrival times with a mean of 9.6 minutes. Twenty-four percent of the jobs in this stream are of Type 1, 44 percent are of Type 2, and the rest are of Type 3. The type of arriving job is independent of the job type of the preceding arrival. Simulate this situation.

Note: This statement, taken from Pritsker book, differ slightly from the one that Schriber has.

APPENDIX J1

SOLUTION OF PRITSKER

S L A M   S U M M A R Y   R E P O R T

SIMULATION PROJECT PROBLEM 8.12 BY ROLSTON

DATE 8/ 1/1980 RUN NUMBER 1 OF 1

CURRENT TIME 0.2400E+04
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
TIME 1	0.3117E+03	0.2057E+03	0.6601E+00	0.2828E+02	0.1054E+04	55
TIME 2	0.2927E+03	0.1588E+03	0.5424E+00	0.4860E+02	0.8232E+03	105
TIME 3	0.5351E+03	0.2243E+03	0.4191E+00	0.1744E+03	0.1371E+04	61

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
CAST	0.1137E+02	0.3036E+01	0.0000E+00	0.1400E+02	0.2400E+04	0.1300E+02
LATHES	0.3099E+01	0.1728E+01	0.0000E+00	0.5000E+01	0.2400E+04	0.2000E+01
PLANERS	0.1990E+01	0.1459E+01	0.0000E+00	0.4000E+01	0.2400E+04	0.4000E+01
DRILL PRESSES	0.5317E+01	0.2264E+01	0.0000E+00	0.8000E+01	0.2400E+04	0.4000E+01
SHAPERS	0.1166E+02	0.3591E+01	0.0000E+00	0.1600E+02	0.2400E+04	0.6000E+01
POLISHERS	0.2016E+01	0.1353E+01	0.0000E+00	0.4000E+01	0.2400E+04	0.4000E+01

FILE STATISTICS

FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1		1.7017	2.7950	12	0	58.3430
2		3.2226	5.1101	19	0	99.1561
3		0.2887	0.9207	5	0	19.2478
4		0.2314	0.6835	5	0	12.0744
5		0.0632	0.3221	3	0	12.6328
6		0.8670	3.0411	16	16	46.2394
7		37.4528	9.4861	52	35	72.4893

APPENDIX K1

SIMULATION TO DETERMINE ARRIVAL TIMES

```
GEN,IMED JAMOUSSI,EXAMPLE,4/12/1987,1;                                  00000070
LIMITS,4,4,100;                                                         00000080
NETWORK;                                                                00000090
        CREATE,EXPON(9.6),,,,1;                                         00000100
        GOON;                                                           00000110
        ACT,,.24,E1;                                                    00000120
        ACT,,.44,E2;                                                    00000130
        ACT,,.32,E3;                                                    00000140
E1      COLCT,BET,P1 BET ARR;                                           00000150
        TERM;                                                           00000160
E2      COLCT,BET,P2 BET ARR;                                           00000170
        TERM;                                                           00000180
E3      COLCT,BET,P3 BET ARR;                                           00000190
        TERM;                                                           00000200
        END;                                                            00000210
;TIME OF SIMULATION                                                     00000220
INIT,0,1000;                                                            00000230
FIN;                                                                    00000240
```

APPENDIX L1

ARRIVAL TIMES

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT EXAMPLE   BY IMED JAMOUSSI

DATE 4/12/1987   RUN NUMBER 1 OF 1

CURRENT TIME 0.1000E+04
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P1 BET ARR	0.3471E+02	0.4729E+02	0.1362E+01	0.8149E+00	0.2461E+03	28
P2 BET ARR	0.2324E+02	0.3633E+02	0.1563E+01	0.3212E+01	0.2378E+03	42
P3 BET ARR	0.2266E+02	0.2128E+02	0.9390E+00	0.6128E+00	0.9183E+02	42

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1		0.0000	0.0000	0	0	0.0000
2		0.0000	0.0000	0	0	0.0000
3		0.0000	0.0000	0	0	0.0000
4		0.0000	0.0000	0	0	0.0000
5	CALENDAR	1.0000	0.0000	2	1	4.3290

APPENDIX M1

RESULTS FROM MDMSS FOR SCHRIBER EXAMPLE

MDMSS

MENU-DRIVEN MANUFACTURING SYSTEM
SIMULATOR

VERSION 1.0

RELEASED APRIL 1987

DEVELOPED BY :
IMED JAMOUSSI

UNDER THE SUPERVISION OF :
DR. JOE H. MIZE

AT

OKLAHOMA STATE UNIVERSITY
DEPARTMENT OF INDUSTRIAL ENGINEERING

FOR MORE INFORMATION CALL (405)-624-6055

GENERAL INFORMATION

DATE : 03-29-87

PROJECT # : SCEXM1

SIMULATION REFERENCE : E1

DEPARTMENT : TESTING

OPERATOR NAME : IMED JAMOUSSI

LEVEL OF SIMULATION : 4

DURATION OF SIMULATION : 2400.00

NUMBER OF FACILITIES : 6

NUMBER OF PRODUCTS : 3

PRODUCT # 1

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 34.7100 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 125.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 13. IN THE 2 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 35.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 3 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 20.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 16. IN THE 4 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 60.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

PRODUCT # 2

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 23.2400 0.0000

GOES TO FACILITY 15. IN THE 1 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 105.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 14. IN THE 2 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 90.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 12. IN THE 3 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 65.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

PRODUCT # 3

ARRIVAL DISTRIBUTION : CONSTANT
PARAMETER(S) : 22.6600 0.0000

GOES TO FACILITY 11. IN THE 1 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 235.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 15. IN THE 2 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 250.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 14. IN THE 3 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 50.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 13. IN THE 4 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 30.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

GOES TO FACILITY 16. IN THE 5 PLACE

PROCESSING DISTRIBUTION : EXPONENTIAL

PARAMETER(S) : 25.0000 0.0000

PERCENTAGE OF REWORK : 0.000

PERCENTAGE OF SCRAP : 0.000

S L A M I I S U M M A R Y R E P O R T

SIMULATION PROJECT IGPSSMP BY IMED JAMOUSSI

DATE 4/12/1987 RUN NUMBER 1 OF 1

CURRENT TIME 0.2400E+04
STATISTICAL ARRAYS CLEARED AT TIME 0.0000E+00

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
P? DEF TIME		NO VALUES RECORDED				
P1 DEF TIME		NO VALUES RECORDED				
P1 BET DEF		NO VALUES RECORDED				
P2 DEF TIME		NO VALUES RECORDED				
P2 BET DEF		NO VALUES RECORDED				
P3 DEF TIME		NO VALUES RECORDED				
P3 BET DEF		NO VALUES RECORDED				
P4 DEF TIME		NO VALUES RECORDED				
P4 BET DEF		NO VALUES RECORDED				
P5 DEF TIME		NO VALUES RECORDED				
P5 BET DEF		NO VALUES RECORDED				
T SYSTEM TIME	0.5984E+03	0.3585E+03	0.5990E+00	0.7816E+02	0.1686E+04	161
P1 SYSTEM TIME	0.3712E+03	0.2083E+03	0.5612E+00	0.7816E+02	0.8829E+03	49
P2 SYSTEM TIME	0.5315E+03	0.2286E+03	0.4302E+00	0.8319E+02	0.9854E+03	67
P3 SYSTEM TIME	0.9453E+03	0.3925E+03	0.4152E+00	0.1458E+03	0.1686E+04	45
P4 SYSTEM TIME		NO VALUES RECORDED				
P5 SYSTEM TIME		NO VALUES RECORDED				
P? BET COMP	0.1390E+02	0.1368E+02	0.9843E+00	0.4614E-01	0.7012E+02	160
P1 BET COMP	0.4520E+02	0.6381E+02	0.1412E+01	0.1059E+01	0.3878E+03	48
P2 BET COMP	0.3346E+02	0.3067E+02	0.9167E+00	0.1606E+00	0.1135E+03	66
P3 BET COMP	0.4301E+02	0.3487E+02	0.8108E+00	0.1729E+01	0.1343E+03	44
P4 BET COMP		NO VALUES RECORDED				
P5 BET COMP		NO VALUES RECORDED				

FILE STATISTICS

FILE NUMBER	ASSOC NODE LABEL/TYPE		AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	Q11	QUEUE	13.6987	11.4838	33	32	188.9473
2	Q12	QUEUE	0.1205	0.5018	4	0	2.4306
3	Q13	QUEUE	0.0661	0.3923	4	0	1.6363
4	Q14	QUEUE	18.5653	13.9281	44	30	303.1074
5	Q15	QUEUE	5.6005	5.8634	24	24	73.4489
6	Q16	QUEUE	0.4210	0.9134	4	0	10.5255
7	Q17	QUEUE	0.0000	0.0000	0	0	0.0000
8	Q21	QUEUE	0.0000	0.0000	0	0	0.0000
9	Q22	QUEUE	0.0000	0.0000	0	0	0.0000
10	Q23	QUEUE	0.0000	0.0000	0	0	0.0000
11	Q24	QUEUE	0.0000	0.0000	0	0	0.0000
12	Q25	QUEUE	0.0000	0.0000	0	0	0.0000

13	Q26	QUEUE	0.0000	0.0000	0	0	0.0000
14	Q27	QUEUE	0.0000	0.0000	0	0	0.0000
15	Q31	QUEUE	0.0000	0.0000	0	0	0.0000
16	Q32	QUEUE	0.0000	0.0000	0	0	0.0000
17	Q33	QUEUE	0.0000	0.0000	0	0	0.0000
18	Q34	QUEUE	0.0000	0.0000	0	0	0.0000
19	Q35	QUEUE	0.0000	0.0000	0	0	0.0000
20	Q36	QUEUE	0.0000	0.0000	0	0	0.0000
21	Q37	QUEUE	0.0000	0.0000	0	0	0.0000
22	Q41	QUEUE	0.0000	0.0000	0	0	0.0000
23	Q42	QUEUE	0.0000	0.0000	0	0	0.0000
24	Q43	QUEUE	0.0000	0.0000	0	0	0.0000
25	Q44	QUEUE	0.0000	0.0000	0	0	0.0000
26	Q45	QUEUE	0.0000	0.0000	0	0	0.0000
27	Q46	QUEUE	0.0000	0.0000	0	0	0.0000
28	Q47	QUEUE	0.0000	0.0000	0	0	0.0000
29	Q51	QUEUE	0.0000	0.0000	0	0	0.0000
30	Q52	QUEUE	0.0000	0.0000	0	0	0.0000
31	Q53	QUEUE	0.0000	0.0000	0	0	0.0000
32	Q54	QUEUE	0.0000	0.0000	0	0	0.0000
33	Q55	QUEUE	0.0000	0.0000	0	0	0.0000
34	Q56	QUEUE	0.0000	0.0000	0	0	0.0000
35	Q57	QUEUE	0.0000	0.0000	0	0	0.0000
36		CALENDAR	31.5418	6.6023	39	33	12.5727

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
1	FACILITY 11	10	9.4176	1.6536	10	0.0000	10.0000	10.0000	132
2	FACILITY 12	4	2.1297	1.3164	1	0.0000	4.0000	4.0000	118
3	FACILITY 13	3	0.9989	1.0094	1	0.0000	3.0000	3.0000	96
4	FACILITY 14	4	3.6548	0.9568	4	0.0000	4.0000	4.0000	113
5	FACILITY 15	12	10.7723	2.9860	12	0.0000	12.0000	12.0000	147
6	FACILITY 16	3	1.5724	1.1815	2	0.0000	3.0000	3.0000	94
7	FACILITY 17	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
8	FACILITY 21	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
9	FACILITY 22	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
10	FACILITY 23	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
11	FACILITY 24	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
12	FACILITY 25	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
13	FACILITY 26	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
14	FACILITY 27	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
15	FACILITY 31	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
16	FACILITY 32	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
17	FACILITY 33	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
18	FACILITY 34	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
19	FACILITY 35	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
20	FACILITY 36	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
21	FACILITY 37	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
22	FACILITY 41	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
23	FACILITY 42	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
24	FACILITY 43	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
25	FACILITY 44	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
26	FACILITY 45	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
27	FACILITY 46	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
28	FACILITY 47	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
29	FACILITY 51	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
30	FACILITY 52	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
31	FACILITY 53	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
32	FACILITY 54	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
33	FACILITY 55	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0
34	FACILITY 56	1	0.0000	0.0000	0	0.0000	2400.0000	0.0000	0

VITA

Imed H. Jamoussi

Candidate for the Degree of

Master of Science

Thesis: MENU-DRIVEN MANUFACTURING SYSTEM SIMULATOR

Major Field: Industrial Engineering

Biographical:

Personal Data: Born in Sfax, Tunisia, April 12, 1962, the son of Hassouna M. and Naziha Jamoussi.

Education: Graduated from Lycee de Garcon de Sfax, now Lycee Heidi Chaker, with a Baccalauriate Degree in June 1981; was offered a scholarship from the Tunisian government to continue the higher Education in the United State of America; joined Oklahoma State University the fall of 1982; received Bachelor of Science Degree in Industrial Engineering from Oklahoma State University in July, 1985; completed requirements for the Master of Science degree at Oklahoma State University in May, 1987.

Organizations: Member of:
1- TAU BETA PI
2- ALPHA PI MU

Professional Experience: Summer job at Energy Power Product, a division of W. W. Williams, Summer, 1982; Teaching Assistant, Department of Industrial Engineering, Oklahoma State University, August, 1985, to December, 1986.

Permanent Address:
107 Avenue Farhat Hached
Sfax 3000
Tunisia
Phone: (216)-(04)-20358 or (216)-(04)-41057