THE DESIGN AND IMPLEMENTATION
OF AN
ELEMENTARY SIMULATION SYSTEM

A Thesis

Presented to

The Faculty of Graduate Studies and Research

The University of Manitoba

In Partial Fulfilment

of the Requirements for the Degree

Master of Science in Computing Science

by

Jack E. Peterson

October, 1971

ABSTRACT

The concept of a discrete simulation, representing the actual flow of transactions through a real physical system, has become a reality. Specialized simulation-oriented computer programming languages have been developed and improved to the point where complex physical phenomena can be translated into computer machine language. Through these languages physical processes normally spanning days, months or years require only seconds or minutes in a computer. Assuming that the basic rationale of simulation is founded and its inherent benefits are obtainable, the justification of which are beyond the scope of this work, then it is the purpose of this thesis to suggest that these benefits can be more easily understood, demonstrated and achieved through the use of non-complex, non-programming notations and translators such as the one described in this paper. The simulation notation to be presented requires no prior knowledge of computer programming languages and introduces the concepts of discrete simulation at an elementary level. Although, introductory in its

approach, the notation does provide the opportunity to incorporate more complex concepts such as random variables, distribution functions and mathematical equations into basic models. In summary, it is the hypothesis of this presentation that the instruction and comprehension of simulation concepts, the preparation of non-complex computer executable models and the introduction to advanced simulations are facilitated substantially by notations such as this one.

ACKNOWLEGEMENTS

The author is indebted to all those who have contributed or assisted in making this thesis possible.

Sincere appreciation is directed to messers J. M. Wells, thesis advisor, and R. L. MacMillan, fellow student, for their valued friendship, advice and encouragement.

To Mrs. Bonnie Pinkerton, typist, this completed work is a testament to her patience and ability under challenging circumstances. I am extremely grateful.

The completion of this thesis would not have been possible were it not for theunfaltering patience and understanding of the author's wife.

List of Figures

Glossary

Accuracy	Closeness of a model's response to response observed in the real world.
Assumption	A fact or statement taken for granted.
Deterministic Variable	A variable whose value is uniquely determined when the factors to which it is related are determined.
Digital Simulation	A simulation executed by a digital computer.
Discrete-Change Model	A simulation model in which state changes occur only at discrete points in time.
Dynamic System	A system whose state changes with time during its normal operation.
Dynamic Entity	An entity which has the ability to move through a system.
Entity	Any distinguished item, being, or processing unit within a model.
Event	An instant in simulated time at which a change to a new system state can take place.
Event Time Advance	A method of time advance where time is incremented from one event time to the next, which may be an increment of several units of simulated time.
Model	A representation of an existing or proposed system.

Parameter	A characteristic value of a transaction.
Process	An instance of an activity during a simulation that can last for a period of time.
Physical Construction	The counter part of the entity in the real world system being modeled. An actual physical phenomenon.
Queue	In general, a waiting line of transactions. In the elementary simulation system the transactions obey the first-in, first out discipline.
Random Numbers	Numbers, usually uniformly distributed between 0 and 1, that occur in such a way as to be completely unpredictable.
Simulation	The manipulation of a system's model to reproduce its operations as it moves through time.
Simulation Clock	A counter used in a simulation model that acts as a clock as simulation time advances.
Simulation Language	A formal terminology and set of programming statements that can be used for, and that facilitates, the construction of a simulation model and computer program.
Simulation Program	A computer program, representing a specific simulation model or a class of models.

System Boundaries	Those parts of the system to be modeled which separate the model section to be simulated from the external larger system.
Stochastic Variable	A random variable, i.e. a variable whose value is not uniquely determined by factors to which it is related but that varies in some statistical way.
Transaction	A dynamic entity.

TABLE of CONTENTS

CHAPTER I

INTRODUCTION

1.1 Objectives of the Thesis

The simulation system described in this presentation represents an attempt to facilitate the simulation of non-complex discrete models. This system incorporates a simple notation used in the model definition stage of the simulation. Once defined the model characteristics are described to the simulation programs by model definition statements. These statements are then processed by the system's programs and the simulation experiment is then executed. The system's output displays relevant information regarding the performance of the model's entities.

The system is a general purpose simulation notation insofar as it does not incorporate concepts unique to any given industry or class of applications. It is restrictive, however, in that only situations capable of being defined as discrete models can be simulated. The basic concept of such models is that of transactions flowing through a network of queues, storages and facilities, each of these representing an actual physical phenomenon. Therefore, traffic models, consisting of automobiles trucks, buses, trains etc. moving through streets, highways, bridges and other transportation entities are valid candidates for this system. The vast number of situations where people are involved with waiting lines, and movement through buildings, houses, apartments, etc are included in the range of this system. The flow of raw, semi-finished and finished materials from primary, and through secondary industry to the consummer can also be modeled and simulated by this simulation system.

Although such transportation, population and logistical situations contain the basic concepts capable of being modeled by this system, this system does not attempt to facilitate the representation of large and complex models.It is intended that elementary situations and sub-systems of larger networks be simulated by this system. More specifically, models presented as part of the initial teaching material of discrete simulations can be represented easily by this system. In addition, the very general, relatively simple models often used to examine the feasibility of more complicated models are within the scope of this system.

The primary design concept of this simulation system is its ease of application. It is intended as a much simpler alternative to the more powerful simulation programming languages such as GPSS, FORTRAN, SIMSCRIPT and the like. It does, however, contain certain necessary features of these languages where such concepts are easily understood and essential to the model building and simulation processes. By retaining its simplicity the system can be used by non-programmers. Thus it opens up the realm of simulation and its benefits to business and industry without requiring them to invest in an expensive educational program such as is required for the more complex languages. Similarily the system can be taught to business and engineering students further extending the potential benefits of simulation to the organizations which will employ these students.

1.2 Comparison With Other Simulation Languages

Without exception, all simulation experiments, discrete or otherwise require a detailed system analysis phase. The result of this step is a complete description of the system to be simulated. This description usually consists of one or more flow charts of progressively increasing complexity and detail as well as the corresponding narratives. The system's physical entities and the logic rules governing the interaction of these entities will be the subject matter of the system description.

In addition, common to all simulations, is an objective of the experiment. This objective will, in some fashion, be a desire to, initially, isolate the critical factors in the performance of the system. The isolation of ineffeciencies, high and low equipment utilizations, bottlenecks, and restricted flows through the system, are a few of the problems which can be explained from a system simulation. Once known, the objective becomes one of minimizing problems within the system's processes. The ineffeciencies are studied, solutions proposed and the altered system again simulated for the purpose of evaluating these solutions. New problems may present themselves and the cycle of alteration and simulation is repeated until a conclusion is reached. This conclusion may be either an acceptance of a series of modifications to the physical system to increase efficiency or it may be a realization that modifications to the system alone are not sufficient to solve the problems. In the latter case, the boundaries of the system may have to be extended to include factors which govern the performance of the previously defined smaller system.

The purpose of a simulation language is to facilitate the definition of the system to the computer and then to allow modification of the described system with a minimum of effort. To date there have been two basic approaches to the design of simulation languages. One approach, which is subsequently referred to as the "transaction method" allows the system to be described primarily as a network of queues, storages, facilities and decision blocks through which transactions flow. A transaction may be any physical entity which, in the actual system, flows through the network. A system defined by the transaction method becomes a system in duplicate, the simulated version a model of the real one. If vehicles flow through streets in the original system then transactions representing vehicles flow through facilities, queues and storages representing streets and other traffic features found in the actual system. The I B M General Purpose Simulation System and its successors are based on the transaction method. The main concept of this approach is its correspondence with a flow chart of the physical network through which the real world transactions flow.

The sample G P S S program provided in Figure 1.1 will illustrate the concept of a transaction flowing through statements which represent actual physical entities. The equivalent statements from the elementary simulation system are also presented. The comparison is further discussed in Section 4.17.

Figure 1.1

Sample GPSS Program

```
      statement                          comments
      GENERATE   1,0,,,,,20              GENERATE A TRANSACTION.
      ASSIGN     1,K100                  ASSIGN TO PARAMETER 1
*                                        THE VALUE 100.
      QUEUE      Q1                      TRANSACTION ENTERS
*                                        QUEUE NO. 1.
      SEIZE      FAC1                    TRANSACTION ATTEMPS TO
*                                        SEIZE FACILITY 1,THEN
      DEPART     Q1                      DEPARTS FROM QUEUE 1
*                                        AND SPENDS TEN TIME
      ADVANCE    10                      UNITS IN FACILITY 1,
*                                        AFTER WHICH THE
*                                        TRANSACTION ATTEMPS
      ENTER      S1                      TO ENTER STORAGE S1.
*                                        HAVING DONE SO
      RELEASE    FAC1                    FACILITY 1 IS RELEASED
*                                        AND BECOMES AVAILABLE
*                                        FOR SEIZING BY ANOTHER TRANS*
      . . .
```

Equivalent Elementary Simulation System Statements

```
CGEN    100,1,(1,1),0,0,1000,(1,101)

CASGN   101,7,(1,1),0,(1,100),(1,-10)

QUE     -10,(1,1)

FAC     1,(1,10),(1,-1)

STOR    -1,1000,0,(1,1),(1,1),(1,300)

. . .
```

Figure 1.2

SAMPLE SIMSCRIPT II SIMULATION PROGRAM
A JOB SHOP SIMULATION

```
PREAMBLE
NORMALLY MODE IS INTEGER AND DIMENSION IS 0
PERMANENT ENTITIES....
     EVERY PRODUCT HAS A SALES.FREQUENCY AND A NAME AND OWNS A STRUCTURE
          DEFINE SALES.FREQUENCY AS A REAL RANDOM LINEAR VARIABLE
          DEFINE NAME AS AN ALPHA VARIABLE
     EVERY PRODUCT,PRODUCT HAS A PRODUCT.SALES(*/2)
     EVERY PRODUCTION.CENTER HAS A (MAX.IN.QUEUE(1/2), MAX.QUEUE(2/2)) IN ARRAY 1
        A (WNUM(1/2), MNUM(2/2)) IN ARRAY 2, A WSUM, A MSUM, A NUMBER.IDLE
        AND OWNS A QUEUE
          DEFINE NUMBER.IDLE AS A VARIABLE MONITORED ON THE LEFT
TEMPORARY ENTITIES....
     EVERY JOB HAS A VALUE IN WORD 2, A DUE.DATE, AN ARRIVAL.TIME,
        AN EXPEDITE.FACTOR FUNCTION, MAY BELONG TO A QUEUE, OWNS A ROUTING
        AND MAY BELONG TO A WAITING.SET
          DEFINE EXPEDITE.FACTOR AS A REAL FUNCTION
          DEFINE VALUE, DUE.DATE AND ARRIVAL.TIME AS REAL VARIABLES
          DEFINE ROUTING AS A FIFO SET WITHOUT P AND N ATTRIBUTES
          DEFINE QUEUE AS A SET RANKED BY HIGH VALUE
     EVERY OPERATION HAS A (CODE(1/2) MACHINE.DESTINED(2/2)) IN WORD 1
        AND A PROCESS.TIME AND BELONGS TO A STRUCTURE AND A ROUTING
          DEFINE STRUCTURE AS A SET RANKED BY LOW CODE WITHOUT M ATTRIBUTE
                     AND WITHOUT R ROUTINES
          DEFINE PROCESS.TIME AS A REAL VARIABLE
EVENT NOTICES INCLUDE WEEKLY.REPORT
     EVERY SALE HAS A PRODUCT.TYPE, A PRICE AND A PRIORITY
          DEFINE PRICE AS A REAL VARIABLE
     EVERY END.OF.PROCESS HAS AN ITEM AND A PRODUCER
BREAK SALE TIES BY HIGH PRICE THEN BY LOW PRIORITY
EXTERNAL EVENTS ARE END.OF.SIMULATION AND SALE
EXTERNAL EVENT UNITS ARE LOCAL.SALES AND IMPORT.SALES
PRIORITY ORDER IS END.OF.PROCESS, SALE, WEEKLY.REPORT AND END.OF.SIMULATION
BEFORE FILING AND REMOVING FROM QUEUE CALL QUEUE.CHECK
BEFORE DESTROYING JOB, CALL STAY.TIME
        DEFINE STAY AS A REAL DUMMY VARIABLE
TALLY AVG.STAY AS THE WEEKLY MEAN, VAR.STAY AS THE WEEKLY VARIANCE, SUM.STAY AS
          THE WEEKLY SUM, SUM.SQUARES.STAY AS THE WEEKLY SUM.OF.SQUARES, AND
          NUM.STAY AS THE WEEKLY NUMBER OF STAY
ACCUMULATE WSUM AS THE WEEKLY SUM, WNUM AS THE WEEKLY NUMBER, AVG.QUEUE AS THE
          WEEKLY MEAN, MAX.QUEUE AS THE WEEKLY MAXIMUM AND FREQ(0 TO 25 BY 1)
          AS THE WEEKLY HISTOGRAM OF N.QUEUE
ACCUMULATE MSUM AS THE MONTHLY SUM, WNUM AS THE MONTHLY NUMBER, AVG.IN.QUEUE AS
          THE MONTHLY MEAN, MAX.IN.QUEUE AS THE MONTHLY MAXIMUM OF N.QUEUE
THE SYSTEM OWNS A FINISHED.GOODS.INVENTORY
DEFINE FINISHED.GOODS.INVENTORY AS A SET RANKED BY DUE.DATE
DEFINE LOCAL TO MEAN DEFINE I,J,K,L,M AND N AS SAVED INTEGER VARIABLES
DEFINE WEEK TO MEAN *HOURS.V*7 HOURS
DEFINE PRIORITY.FREQUENCY AS A 2-DIMENSIONAL ARRAY
DEFINE TITLE AS A TEXT VARIABLE
DEFINE WEEK.COUNTER AND TAPE.FLAG AS INTEGER VARIABLES
DEFINE AVERAGE AS A REAL FUNCTION WITH 1 ARGUMENT
END
```

The second approach, as exemplified by SIMSCRIPT, SPURT (1), FORTRAN and other high level general purposes languages, rather then being oriented towards a flow chart of the physical system rely heavily on the narrative description of the system. Every event, operation, facility, storage, queue and transaction is defined by program statements. Hereafter, referred to as the "programmed method" this approach results in a program consisting of statements which in certain languages such as SIMSCRIPT closely resembles the actual narrative description of the real world system.

Figure 1.2 is a sample of a SIMSCRIPT program taken from a paper presented by P.J. Kiviat at the Second Conference on Applications of Simulation, December 1968, entitled " Introduction To The Simscript II Programming Language.", Digest of the Second Conference On Applications of Simulation

From this example the correspondence between the system narrative description and the program statements can be analysed.

Phrases such as

"EVERY JOB HAS A VALUE IN WORD 2, A DUE. DATE, AN ARRIVAL TIME,..." ,

"EVERY OPERATION HAS A ..." ,

"DEFINE PROCESS. TIME AS A REAL VARIABLE", and

"EVENT NOTICES INCLUDE WEEKLY, REPORT "

clearly illustrate the narrative nature of this language.

FORTRAN, SPURT which is a set of simulation oriented FORTRAN subprograms, PL-I and other general purpose languages require the representation of the physical system entities in more or less meaningful program statements. However,the"programmed method " is retained.

It is not the purpose of this presentation to compare in detail the relative merits of either the programmed method or the transaction method. It is, however, within the scope of this thesis to propose the elementary simulation notation described herein as a less complex alternative to GPSS or the programmed method languages for the simulation of elementary systems. Like GPSS this simulation system is based on the transaction method. In addition many of the concepts implemented in GPSS have also been included in this system. Chapter 4 discusses in detail those simularities. It is the purpose of chapters 2,and 3 to present the elements of this system and to demonstrate the case of defining non complex physical systems.

Chapter 2 contains the complete specifications for the use of all elements of the language. The process of model building using this notation is described. The basic concepts of transactions, facilities, storages and queues are discussed. For the most part this chapter describes the system notation and model definition statements. Consideration is also given to the procedure for executing simulation experiments, error messages and correction suggestions as well as simulation output interpretation.

Chapter 3 presents examples of situations modeled and simulated with this system. The process from initial model formulation, to modification of the model to evaluate alternate model characteristics as well as final conclusions is presented.

Chapter 4 presents a complete system description. The basic system flow and procedures of each program and sub-program are presented. Program design concepts are discussed in general and in detail. The method of implementing each element of the language is described. Operational considerations and methods of expanding the language are also considered.

Chapter 5 summarizes the language and its relative merits. There are eight appendices; C, which is a language reference sheet used instead of the more lengthy chapter 2 when building models; A, & B are lists of error messages and the remainder are listings of the simulation system programs with sample output.

CHAPTER II

SIMULATION SYSTEM SPECIFICATIONS

2.1 General Description

The elementary simulation system is a general purpose simulation language. The term general purpose applies because the system has not been designed specifically for a particular class of discrete simulations. Rather, generality of terminology and implementation has been maintained. The system may be applied to any class of discrete, transaction oriented simulation experiments.

Although it is possible to describe relatively complex systems with this language, it has not been designed for this purpose. The version described in the following pages has been implemented with certain limitations in the sizes of the simulation allowed. These will be described in subsequent sections. By the intentional implementation of these restrictions and the avoidance of complex language features this simulation notation requires no prior programming experience, and therefore may be taught at an introductory level to non-computer oriented students. In addition, production managers in the manufacturing industries, shipping and warehouse managers in the wholesale industry, traffic managers in the transportation industry and all supervisors involved with transaction oriented systems, without investing considerable time, can learn this notation and implement simulation experiments.

The application of simulation, lends itself to a relatively concise description. Initially, the system to be studied must be clearly understood. Defining the scope of the simulation is accomplished in part by defining the boundaries of the system to be considered. The interaction of the transactions and the physical constructions must be described. Then, the representation of the real world system, the model, must be defined. The model will usually incorporate various simplifications of events and conditions of the actual situation. The model must then be defined to the computer, and validated, after which simulation experiments can begin.

Although, it is not the purpose of this presentation to instruct the students in the techniques of model building and simulation, section 2.4 presents an approach to model building using this notation. This is appropriate to ensure that the benefits of this system are achieved. Once constructed, the model is easily described in the notation of this system.

Prior to the discussion of model building the basic concepts of the transaction and the physical entities are described. Following the model construction section attention is given to relevant aspects of the completed model. The method of defining the model using the system's notation is presented followed by the detail specification of each of the model definition statements. Finally, the simulation execution, errors in the model and output interpretation are discussed.

2.2 Transaction Concept

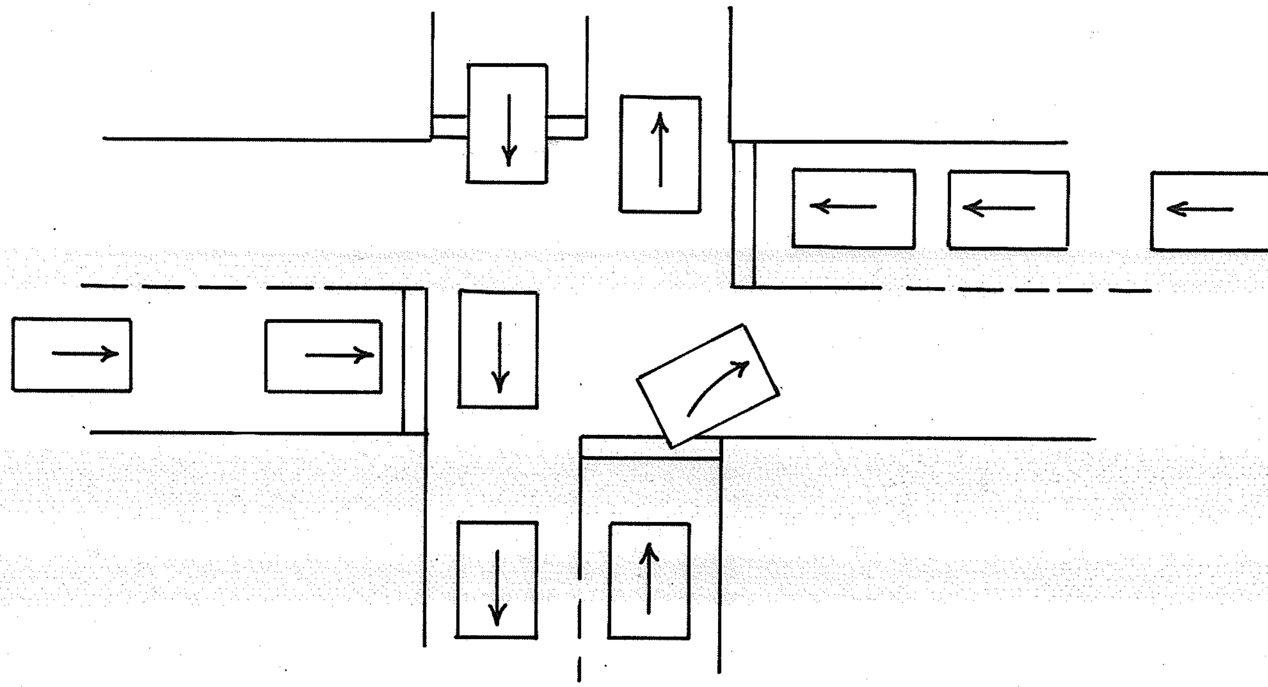
2.2.1 The Transaction

The basic concept of the elementary simulation system is that of the transaction. In the context of this system a transaction is a representation of a dynamic entity from the real world system being simulated. The main characteristic of such an entity is its ability to move through the physical system.The network of physical constructions which constitute a part of the real system are meaningless until materials, resources, vehicles, messages and the like begin to flow through and become modified by these constructions. Similarily, within the simulation, until transactions begin to flow through the simulation representation of the actual physical constructions the simulation has not begun. Therefore the materials, resources, vehicles and messages of the real system are represented by different transaction types in the simulated system. A maximum of one hundred active transactions are allowed at any one time.

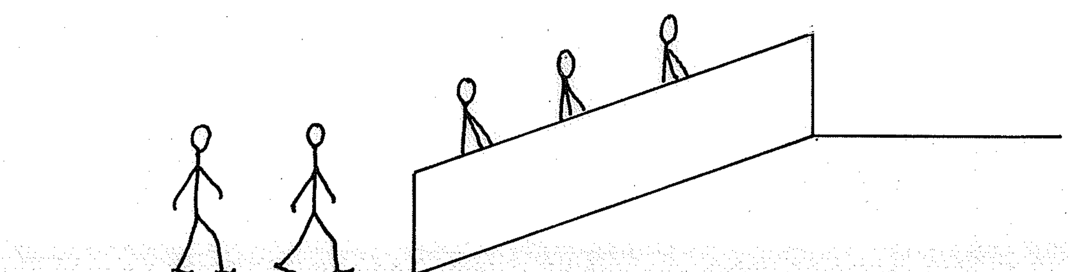
The system of a traffic intersection possesses the dynamic entities of vehicles and pedestrians. A simulation of such a system would consist of one transaction representing each pedestrian and one transaction representing each vehicle. If one hundred vehicles pass through the system then one hundred transactions representing vehicles would exist in the simulation. The parallel also exists for the number of pedestrians. Figure 2.1 is a pictorial representation of the transaction concept.

2.2.2 Transaction Parameters

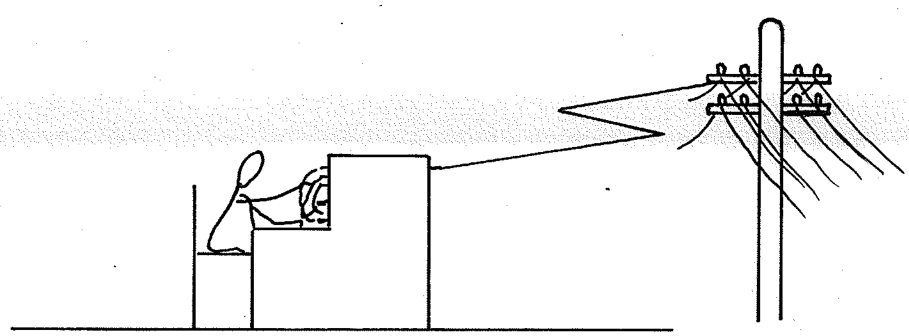
In the previous section (2.2.1) the term transaction was used to mean both vehicles, pedestrians and in a broader sense materials, resources, messages and any other representation of a dynamic physical entity. In a simulation the problem exists of distinguishing between transactions which are representing different physical entities such as vehicles, and pedestrians. In fact it may be desirable to identify different types of vehicles as well as various pedestrian types. In the elementary simulation system this distinction is accomplished in two ways. Transactions representing completely different concepts which do not use the same physical construction in the real system will not likely use the same representation of these physical



Cars at an intersection



People riding an escalator



Telephone calls at a switchboard

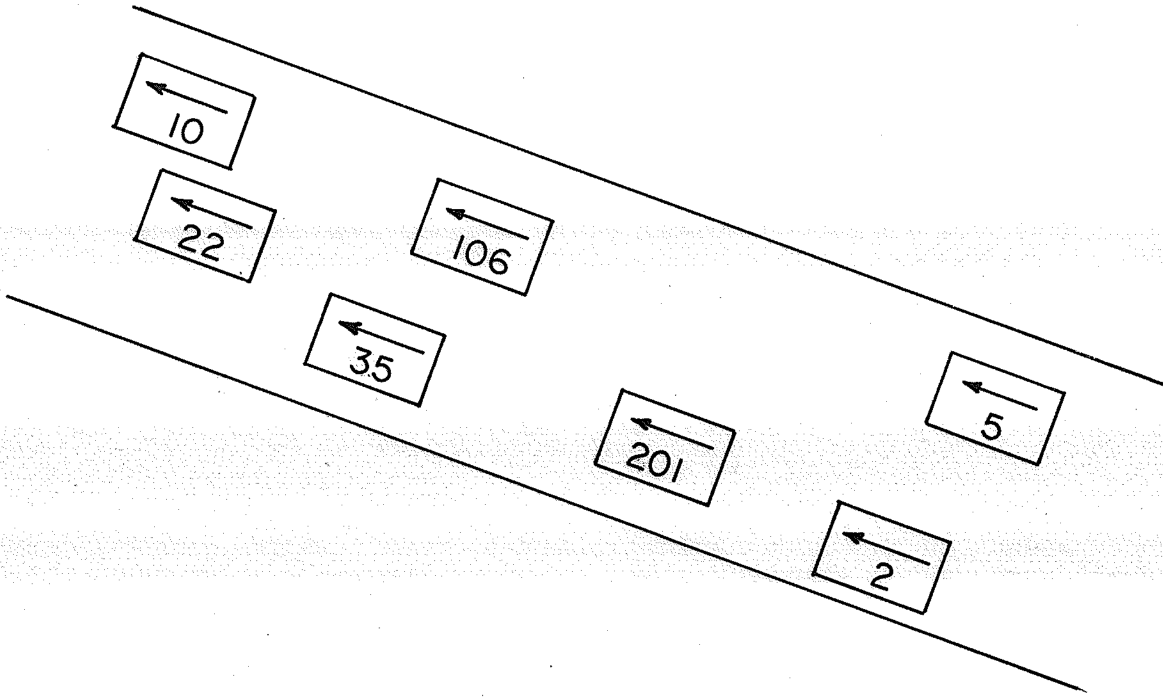
Figure 2.1

constructions in the simulation. Therefore, one separation of transaction types is inherent in the nature of the transaction itself and the separate flow paths available to each transaction.

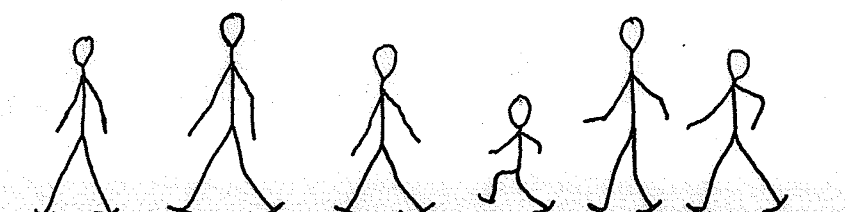
The second method of distinction which applies when differing representations are desired within the same or closely related transaction types is accomplished by means of five parameters attached to each transaction. These parameters are part of the transaction and move whenever the transaction moves. Each of the five parameters can be assigned a numeric integer value from $-(2^{31})$ to $(2^{31} - 1)$.

The transactions representing vehicles and pedestrians could have parameter one defined as positive for pedestrian and negative for vehicle. Different pedestrian types would be assigned different positive integer values while different vehicle types would be assigned unique negative integer values. Parameters two to five might be used to describe vehicle colour or make,age, condition, engine type for vehicles or age, direction, sex, weight, and height for pedestrian differences. Figure 2.2 illustrates the transaction parameter concept. Parameteric values are assigned and modified by means of the control block "assign" described in section 2.12.7.

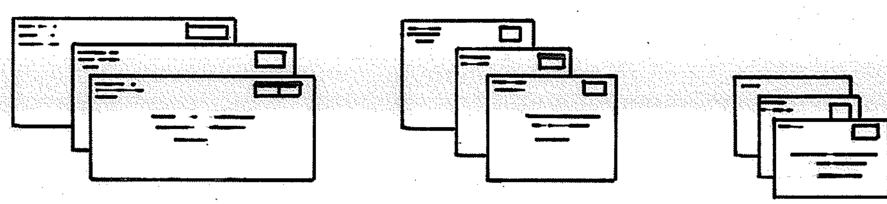
Transaction Parameter Concept



Racing cars with different numbers, speeds, etc.



People of different height, weight, age, sex, race, etc.



Letters to different cities from different cities

Figure 2.2

Although it is not the purpose of this simulation system to provide features required for large simulations, where necessary, improvisations can be made to extend the system. One situation where the extension may be required is in the representation of more than five transaction characteristics using only five parameters. Consider the representation of the human attributes of height and weight as an example. The range for height is approximately from twenty-four inches to ninety inches, while for weight it varies from ten pounds to four hundred pounds. By multiplying height by a factor of one thousand and adding it to the weight both can be represented by one number.

e.g. ;

height- 61 inches

weight- 208 pounds

becomes 61 X 1000 + 208 = 61,208

To decode this value divide by 1000 for the height and subtract 1000 X the integer value of the height for the weight ie: height $= \frac{61,208}{1000} = 61$

weight $= 61,208 - 61 \times 1000 = 208$

If the height is required to one decimal place a factor of ten thousand can be used.

2.3 The Physical Entities : Facilities, Storages, Queues

2.3.1 The Facility

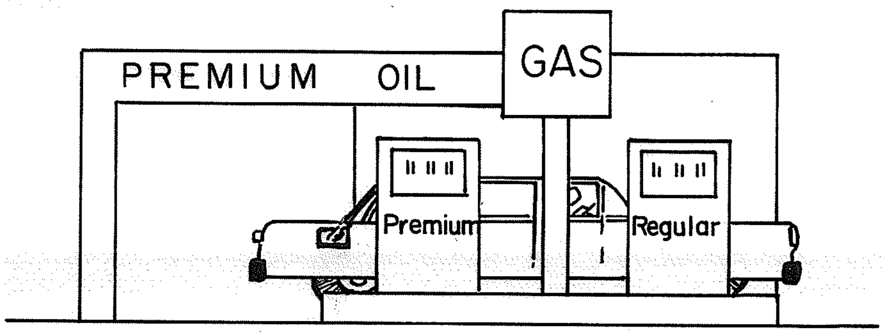
The term facility, as used in the context of this simulation system, refers to a physical construction which acts upon transactions, perhaps to alter their characteristics,direction,etc. Usually implied in the processing of a transaction by such a physical construction is the passage of time.A machine in a manufacturing plant which converts semi-finished materials to a higher degree of completion is a facility. A conveyer belt transporting parcels, an intersection through which vehicles pass; an elevator or escalator carrying people, or a computer processing a payroll,are examples of facilities. Figure 2.3 contains example of facilities.

2.3.2 The Storage

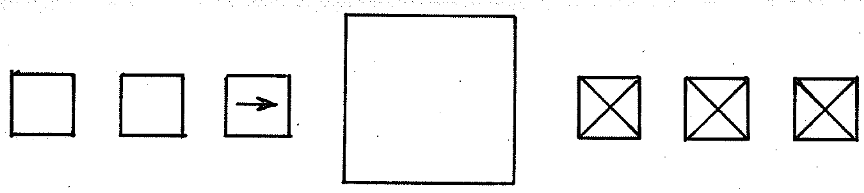
A storage in a real world system usually has the capability of holding, retaining or detaining one or more transactions. A storage in a simulation similarily has the capability of containing transactions. These transactions may leave while new transactions may enter. Each storage has a maximum capacity.

Some examples of storages which may be represented similarily in this simulation system are: a water reservoir an oil tank, a wholesale warehouse, a grain storage bin, a series of counter shelves, a frozen

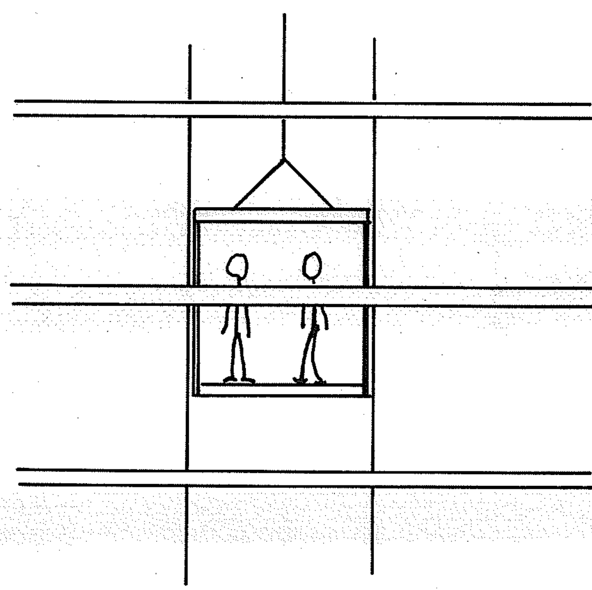
Facility Concept



A service station is a facility



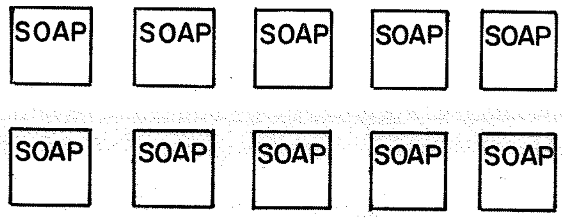
A manufacturing process is a facility



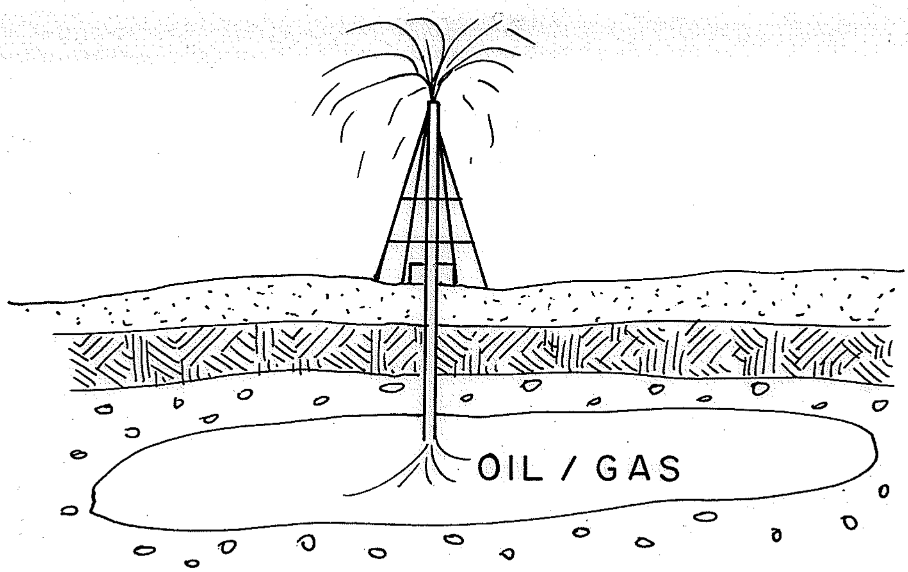
A building elevator is a facility

Figure 2.3

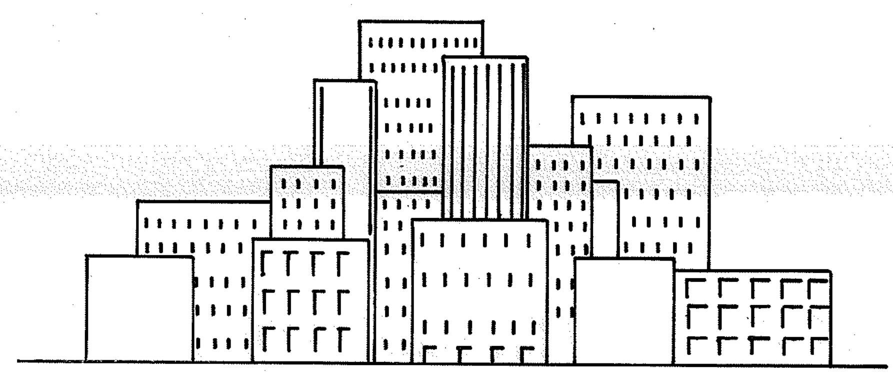
Storage Concept



A warehouse containing cartons of soap is a storage



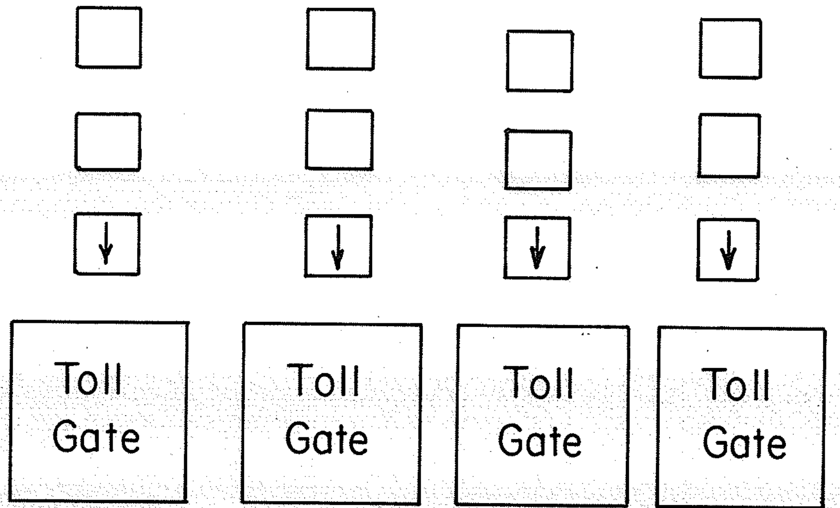
The earth is a storage place for minerals



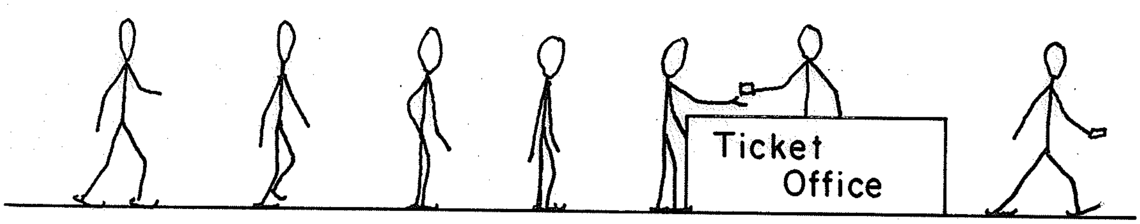
A city is a storage for cars, trucks, people, trains, etc.

Figure 2.4

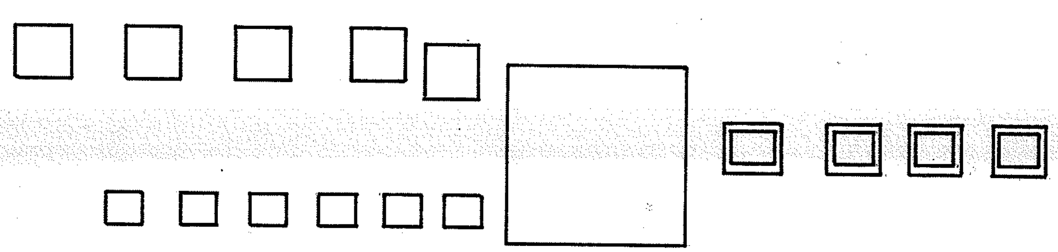
Queue Concept



Cars waiting to pass through a toll gate constitute a queue



People waiting to buy tickets constitute a queue



Sub-assemblies entering the assembly shop wait in queus

Figure 2.5

foods freezer, a classroom of students, an entire office building or a parking lot. Figure 2.4 illustrates storages.

2.3.3 The Queue

Any ordered waiting line is considered a queue in the context of this simulation system. The prime consideration is the term "ordered". A queue exists when transactions join a waiting line and leave that line in the order they arrived. A queue need not have a maximum length. Figure 2.5 illustrates the queue concept. Other examples of queues are: jobs waiting for access to the computer, cars waiting for a green light, airplanes waiting to take off, cars waiting for gas at a service station and people waiting to enter a theatre.

2.4 Model Building Concepts

2.4.1 The Modeling Process.

A model, simply stated, is a representation of a proposed or existing system. The process of building the model is called modeling. Simulation models are constructed for the analysis of systems. A system is any bounded section of reality. Once a system is defined, the purpose of the simulation study determines what a model of the system will look like. Simulation is the manipulation of a model to reproduce the operations of a system through time. The modeling process must consider the purposes of the model, the precision required of the output, the corresponding detail to achieve this degree of accuracy, the assumption necessitated at the system boundaries and the availability of data, to name a few. (2)

Modeling as a process is iterative. Initially the model will likely appear as a gross over-simplification of the real system. Gradually, as the processes and interactions of the system are studied from different perspectives, the model will change. Assumptions, approximations, decision rules, stochastic and deterministic representations and the like will become progessively evident as the model in each of its stages is compared with the original system. P.J. Kiviat (2) has defined the modeling process as follows :

Stage 1 : Statement of a problem in general system terms.
Definition of gross system boundaries.
Statement of output (s) needed to solve the problem.

Stage 2 : Statement of (initial) assumptions.
Definition of static and dynamic system structure.
Construction of minimal system model.
Assessment of assumption is the light of Stage 1 goals.

Stage 3 : Determination of input data requirements and availability.

If input data required are not available, modify assumptions and model structure by returning to Stage 2.

Stage 4 : Determination of output possibilities.

If output is insufficient, modify assumptions and model structure by returning to Stage 2.

Stage 5 : Prepare precise specifications for final model.

Select a modeling and programming language.

Reassess the implications of all assumptions for the future. Prepare a detailed plan for use of the model.

Reference will be made to these five stages, and the achievement of each, within the context of the simulation notation described in this presentation.

2.4.2 Modeling With The Elementary Simulation System Notation

The final process of specifying the model to the computer, using this system's notation is dependent, almost entirely, on a flowchart of the system model. The prime requirement of this flow chart is that it be transaction oriented. That is, the flowchart should represent a network of physical entities joined by directed lines indicating the paths that transactions can follow. Where appropriate, such as the definition of the model boundaries, conditional situations and the like, transaction oriented control blocks are inserted into the paths joining the physical entities. Figure 2.6 illustrates a simple configuration of the physical entities as might be defined in a model flowchart. Figure 2.7 is the same configuration of the physical entities with the addition of a control block to generate transactions into the system and a termination block to remove them. Figure 2.7 , with both system boundaries defined, is a valid model configuration while figure 2.6 is a model subset. In this fashion, models are constructed, using the elememtary simulation notation.

The correspondence between the physical configuration of the actual system and the proposed model, using this notation, should be almost direct, excepting the inclusion of control blocks to define the system's boundaries and it's decision rules. Therefore the problem of constructing an accurate model becomes one of defining the decision rules and representing the system data.Data representation, both stochastic and deterministic is accomplished using the "FUNCTION " and "VARIABLE " model definition statements. In addition, fourteen control blocks, sometimes used in conjunction with the FUNCTION and VARIABLE statements, are available to assist the modeling of the system's interactions and decision rules. For the most part these control blocks are verbs which describe an action on one or more transactions. Some representative control blocks are: TEST a condition; ASSEMBLE two or more transactions into one before proceeding;GATHER two or more transactions then proceed; ASSIGN a value to a parameter; GENERATE a transaction into the system; REMOVE a transaction from the system; SPLIT one transaction into two or more , etc.

Figure 2.6

Sample queue, facility, storage configuration.

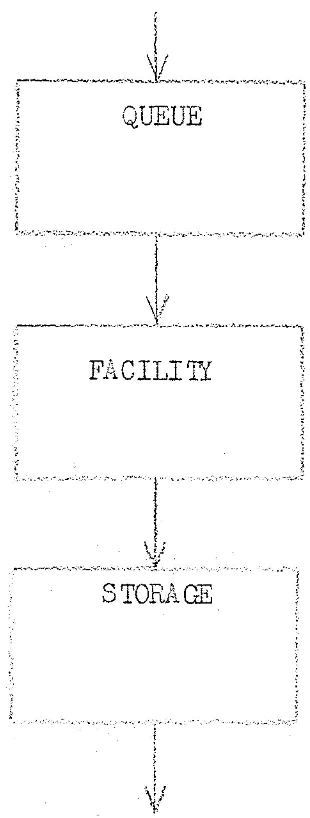
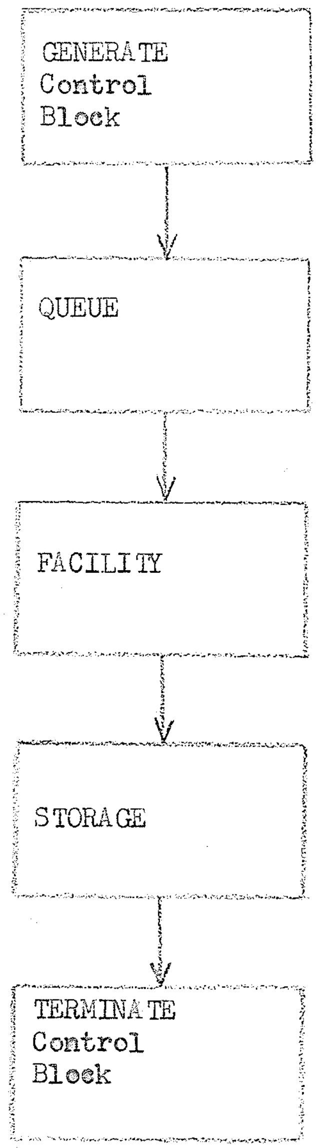


Figure 2.7

Generation and Termination Control Blocks.



2.5 Model Attributes

2.5.1 Definition and Description

A model attribute is a quantity, a measure, a characteristic or a parameter pertaining to any element of a model. Each attribute in some manner illustrates a quantitative characteristic of the model. In the process of a simulation experiment an attribute may remain constant or it may vary. The model attributes available in this system can be divided into two types: those which can be modified by the model builder and those which cannot be modified directly. Both types are available to be inspected or in some manner used by the model builder.

Certain model attributes such as queue contents, storage contents and the availability of a facility are often tested in the model to determine the path a transaction will take. Other characteristics of these physical entities are more useful to evaluate the overall system performance. The average contents of a queue, the average time in the queue per transaction, and the utilization of a facility or a storage are examples of model characteristics which illustrate the system's performance. The attributes just described are of the type which are accessible to the model builder but whose values are determined by the processes being simulated.

The second type of model attributes are those whose values the model builder may vary at any time within the simulation execution. The transaction parameters are this type. The initial and all subsequent values of each parameter are assigned by model definition statements. Other attributes of this type are the system wide counters which, unlike the parameter, are not unique to a given transaction. Functions representing points on a graph, and mathematical equations called variables must be defined by the model builder. Common scalars or constants in certain circumstances are considered in this group as well. Figure 2.8 is a table illustrating all the model attributes available to the model builder. The allowable attribute types are: direct, implying their direct control within the model; accessible indicating these as being accessible only, and output meaning that these attributes are provided for inspection as general simulation statistics.

Figure 2.8

Model Attribute Types

Name	Type
1. scalar	direct,output,accessible
2. function	direct,output,accessible
3. variable	direct,output,accessible
4. parameter	direct,output,accessible
5. counter	direct,output,accessible
6. queue contents	output,accessible
7. storage contents	output,accessible
8. facility availability	output,accessible
9. transaction entries (facility,storage,queue)	output
10. average utilization (facility,storage)	output
11. average time per transaction (facility,storage,queue)	output
12. average contents (storage,queue)	output
13. maximum contents (storage,queue)	output
14. minimum contents (storage,queue)	output
15. queue zero entries	output

2.5.2 Specification of Direct Model Attributes

The direct model attributes are the parameter, variable, function, counter and scalar. To specify these or to reference the value represented by one of these attributes, two integer values are required. The first of this pair defines the type while the second identifies the number within the type. Figure 2.9 indicates the direct attributes, the type, valid number ranges within type and an example or a reference to one.

These direct model attributes are used in several situations by the model builder. The concept of the parameter was discussed in section 2.2.2. A counter like a parameter is used to store integer values. Unlike a parameter a counter is not associated with only one transaction. Counters can be referenced by any transaction in the model. Counters are described in section 2.12.8. A scalar is any integer which is evaluated whenever the variable is referenced (Section 2.14). A function is a two dimensional graph which when referenced yields the value of the dependent argument corresponding to the value of the independent argument defined in the function (Section 2.15).

2.6 Use of the Model Attributes

The model attributes describe in section 2.5 are used in a variety of ways in the definition of the model.

2.6.1 Next Entity Specification

In the discussion on model building the importance of the transaction oriented flow chart was emphasized. Each

Figure 2.9

Specification of Direct Model Attributes

Name	Type	Range of Numbers	Example
scalar	1	integers	(1,-233)
function	2	1 to 20	(2,3)
variable	3	1 to 19	(3,8) ;V8=LN(P1*C1/256)
parameter	4	1 to 5	(4,5)
counter	5	1 to 25	(5,25)

of the physical entities and the control blocks included in this model diagram are numbered accordingly to the rules:

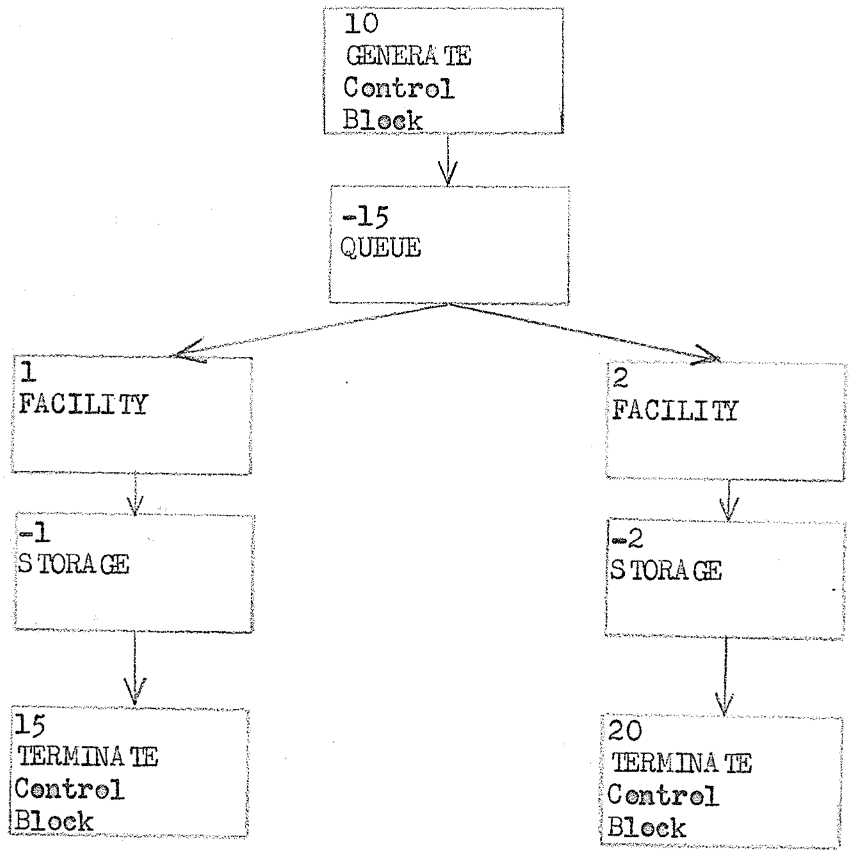
1. Assign a positive integer to each facility,
2. Assign a higher positive integer to each control block,
3. Assign a negative integer to each storage,
4. Assign a lower negative integer to each queue,
5. The range of integer values used for one entity type, need not be inclusive, but must not overlap the integer range of any other entity type. These rules will be reviewed in section 2.7.

Figure 2.10 illustrates a sample model diagram with each of the entities numbered. In this fashion each entity is identified to the simulation system programs.

Each numbered model entity is defined to the simulation system by means of the model definition statements. A separate statement type exists for each entity type ie. facility, storage, queue, control block. When a transaction is generated the next entity as specified in the corresponding model definition statement is entered by the transaction. In figure 2.10 the second entity is queue "-15". In the queue model definition statement the path to be taken upon leaving is stated. From this diagram there are two possible next entities;

Figure 2.10

Sample Model Diagram with Numbered Entities



"facility 1" or "facility 2". The choice of which to enter might be based on several factors. Some possible factors might be: availability of both facilities, a parameter of the transactions, a model counter value, a randomization scheme, or a mathematical equation. The method of choosing the branch (left or right) is indicated by a pair of integers (recall section 2.5).If the decision of which facility to enter was dependent on function number two then the next entity arguments in the queue model definition statement would contain the integer pair (2.2).The first 2 indicates that a function gives the number of the entity, the second 2 indicates which function.

The facility model definition statement for "facility 1" would contain the integer pair (1,-1) because the next entity is always the one with the number -1.Wherever a next entity specification is required an integer pair defining an attribute type and its number must be defined.

2.6.2 Definition of Entity Block Characteristics

Many block characteristics are defined by pairs of integers in the same manner as the choice of next entity.These are listed below.

1. Facility processing time per transaction

2. Units entering a storage per transaction

3. Units leaving a storage per transaction
4. Parameter identity in several control blocks
5. Counter identy in several control blocks
6. Simulation termination count

The meaning of each term will be discussed in the section on the appropriate model definition statement.

2.7 Specifying the Model : Notation

The specification of the completed model to the computer for execution is accomplished by means of a relatively simple system of notation. The notation consists of model definition statements. Each such statement describes either a facility, a queue, a storage, a control block, a variable equation or a function. Together all the model definition statements define the transaction paths through the model, the model's decision rules and all relevant data.

The general form of the model definition statement is:

Model Definition Statement Format

$$La\ldots ad\ldots dx_1d\ldots dx_2d\ldots dx_3d\ldots dx_{n_1}$$

where L is a key letter(s) indicating the statement type,

a...a is a string of alphanumeric characters,

d...d is a string of delimeter characters,

x_i, i= 1,2,3,...n are numeric values defining the statement characteristics,

n_1 is the number of numeric values to be specified for statement type "L".

Model definition statements are usually specified one per line.

Figure 2.11 specifies the valid delimeter characters. Figure 2.12 indicates the key letters of the statement identifies and the number of numeric arguments. Figure 2.13 illustrates some sample model definition statements.

The remainder of this chapter, excepting the last three sections, is devoted to the definition and illustration of each model definition statement type.

Figure 2.11

Table of Delimeters

Character	Description	Character	Description
blank	blank or space	>	"greater than"sign
,	comma	<	"less than"sign
/	slash or oblique	=	equal sign
*	asterisk	$	dollar sign
(	left bracket	&	ampersand
)	right bracket	|	vertical line
:	colon	@	"at" sign
;	semi-colon		

Figure 2.12

Model Definition Statement Summary

Statement Type	Statement Identifying Character(s)	Number of Arguments
Facility	F	5
Storage	S	9
Queue	Q	3
Variable	V	1
Function	FU or FN	variable
Control Block	C	4 to 9
Decision Rule	D	variable

Figure 2.13

Sample Model Definition Statements

```
Q1  -10 ( 1, 4)
Q2  -10/1/3/
Q   -11//1//3
STORAGE  1:(8000,0),(1,1),(1,2),(1,25)
V2= 200+V1-C3
FAC  20,(5,22),(1,100)
FUN1  1,2,2,5(0,0/1,2/2,3/,3,4/4,5)
FN2  2,2,1,3(0,0/.5,6/1.0,7)
```

2.8 Entity Numbering Rules

In section 2.6.1 "Next Entity Specification" the rules of numbering the entities of the model were briefly introduced. These rules are restated in this section. In the many examples which follow in this chapter the use of these rules is thoroughly illustrated. Figure 2.14 contain the numbering rules. Figure 2.15 illustrate these rules as applied to an elementary model.

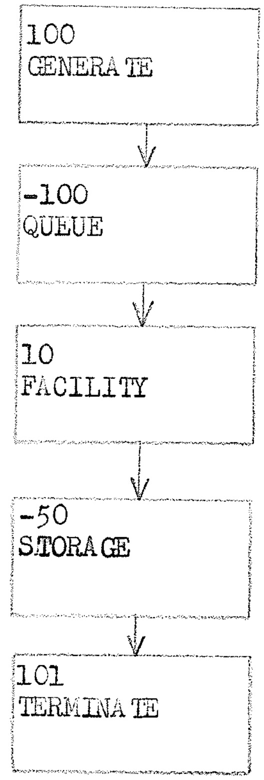
Figure 2.14

Entity Numbering Rules

Entity Type	Valid Numbering Range
Facility	- numbers greater than zero,
Control Block	- numbers higher than the highest facility number,
Storage	- numbers less than zero,
Queue	- numbers lower than the lowest storage number.

Figure 2.15

Sample Model Diagram with
Numbered Entities



2.9 The Facility Statement

The facility statement defines the characteristics of the facility.The general form of this statement is

$$Fa..ax_1d..dx_2d..dx_3d..dx_4d..dx_5$$

where "F" begins in any column,

a-are alphanumeric characters

d-are delimeter characters

x_1- the facility number

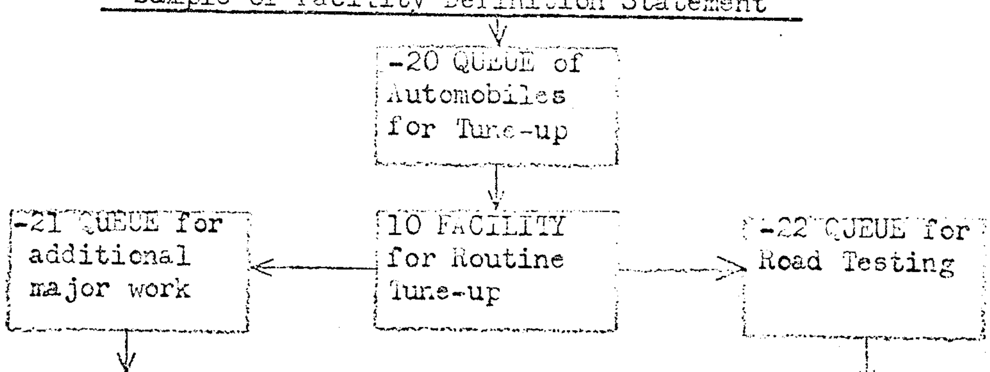
x_2, x_3 -the processing time

x_4,x_5- the next entity

Figures 2.16 and 2.17 illustrate the use of this statement.

Figure 2.16

Sample of Facility Definition Statement



In the model section illustrated in the diagram automobiles are entering a large auto repair and maintenance centre. Each transaction represents one vehicle. The autos entering this section of the model have the following parameter definitions:

Parameter Number	Definition	Range of Values
1	engine type	11-4 cylinder
		12-6 cylinder
		13-8 cylinder
2	performance type	1-economy
	(compression and	2-medium performance
	carberation type)	3-high performance

The servicing time per auto in the "routine tune-up" facility is defined as follows:

Service time = (engine type) * 2 + (performance type) * 2

= (Parameter 1) * 2 + (Parameter 2) * 2

This definition of the service time is accomplished by the varible equation

$$V1 = P1 * 2 + P2 * 2$$

$$\text{or } V1 = (P1 + P2) * 2$$

Thus variable number 1 defines the service time for any automobile entering this facility.

After the auto has been serviced there is a probability that additional major work is required.

Applying this probability to the choice of the next entity 25 % of the time an auto will require additional work and thus enter queue number "-21". The remainder of the vehicles will enter the road testing section indicated by queue "22". This relationship, ie 25% to queue -21 and 75 % to queue -22, is defined by a function. The function which accomplishes this relationship is defined as function number 2.

FUNCTION 2,2,3,2,3 (0.0,0.0),(.25,-21),(1.0,-22)

The interpretation of this statement is left to section 2.14.

From the above diagram and additional information the facility definition statement can be defined.

FTUNEUP 10 , (3, 1) (2 , 2)

X_1 - facility number

X_2 - service time type

X_3 - service time attribute #

X_4 - next entity attribute type

X_5 - next entity attribute #

Figure 2.17 Facility Definition Statements

In figure 2.17 the specification of the facility definition statement was illustrated in the context of the total model building process. In this figure only the actual facility statement specification is illustrated.

FAC 1 , (1, 20) , (1, 30)

X_1 - facility number

X_2 - service time attribute type

X_3 - service time attribute number

X_4 - next entity attribute type

X_5 - next entity attribute number

The pair (X_2, X_3) = (1, 20) implies that the service time is the constant 20 time units for every transaction. The pair (X_4, X_5) = (1, 30) implies that the next entity is the constant, entity number 30 for all transactions.

FACILITY 2 , (4 , 3) ,(5 , 5)

X_1 - facility number 2

X_2 - service time attribute type

X_3 - service time attribute number

X_4 - next entity attribute type

X_5 - next entity attribute number

where (X_2, X_3) = (4,3) implies that the service time is given by the parameter 3 for the transaction being processed, and

where (X_4 ,X_5) implies that the next entity is given by the fifth counter.

2.10 The Storage Statement

The characteristics of each storage included in a model are defined in storage statements. The general form of this statement is:

$$Sa\ldots ad\ldots dx_1d\ldots dx_2d\ldots dx_3d\ldots dx_4d\ldots dx_5d\ldots dx_6d\ldots dx_7d\ldots dx_8d\ldots dx_9$$

where

S- begins in any column

a- is any alphanumeric character

d- is any delimiter

x_1-is the storage number

x_2-is the maximum contents

x_3-is the initial contents

x_4 and x_5 are the model attribute type and attribute number defining the number of units added to the storage contents each time a transaction enters this storage,

x_6 and x_7 are the model attribute type and attribute number defining the number of units subtracted from the storage contents each time a transaction leaves this storage,

x_8 and x_9 are the attribute type and attribute number defining the next entity for the leaving transaction.

Figures 2.18 and 2.19 illustrate the storage statement.

Notes on the Storage Concept

1. The capacity of a storage need not be measured in transactions. The model builder chooses the storage unit, and expresses the capacity, the initial contents, the addition to the contents and the subtraction from the contents in these units.

2. If a storage is full when a transaction attempts to enter it and add to its contents that transaction will be held infront of that storage until the contents are decreased sufficiently to allow it's entry.

3. If the storage capacity will be exceeded by the entry of a transaction with the corresponding addition to its contents the transaction is held in front of the storage as in #2.

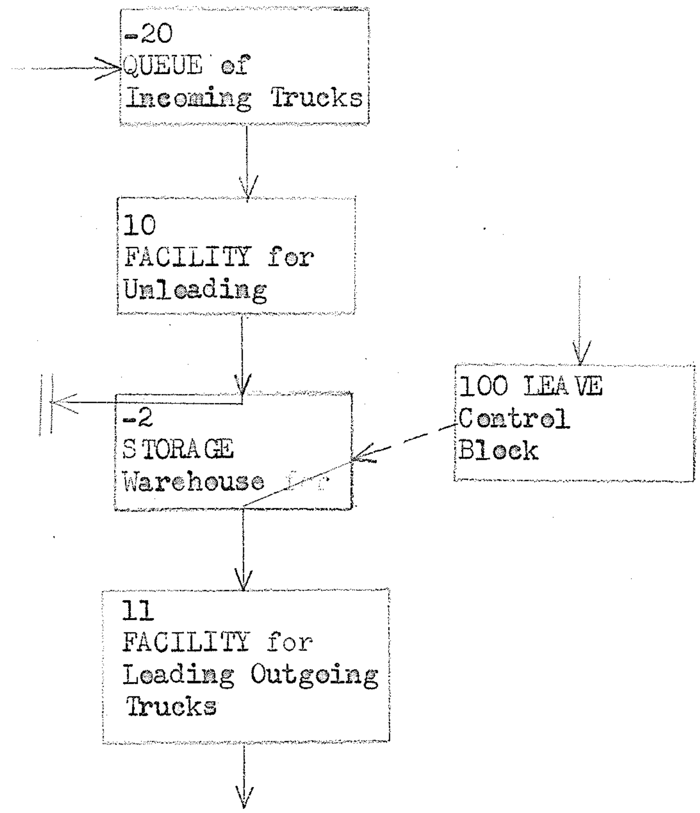
4. If a transaction which is leaving a storage attempts to subtract more units than exist in the storage at that time the storage contents are reduced to zero and the following message is printed

" A TRANSACTION ATTEMPTED TO REMOVE M UNITS FROM
STORAGE # N.
THE STORAGE CONTAINED X UNITS.
THE CLOCK TIME IS Y UNITS.
THE STORAGE IS NOW EMPTY."

After the transaction has emptied the storage it will continue on its normal path.

Figure 2.18

Sample Storage Definition Statement



A transaction which enters a storage block will add to the total storage contents. This transaction is then removed from the system. A transaction which enters a "LEAVE" control block subtracts units from the total storage contents.

The model segment illustrated in the diagram depicts a situation of incoming delivery trucks joining a queue. The trucks are unloaded with their packages being stored in a warehouse. From another section of the model delivery trucks arrive at the warehouse to pickup items from storage (Control Block "LEAVE").

Transactions representing trucks join the queue, then the facility. The same transactions enter the storage. After entering a storage the system removes the entering transaction from the system automatically. No transactions will leave a storage unless a "LEAVE" control block with the appropriate storage number is entered by a transaction. When a transaction enters a "LEAVE" control block the storage number defined in the control block's arguments is evaluated. The transaction is automatically moved to the referenced storage, it then leaves the storage,reducing the storage contents by the value specified in the storage definition statement. This transaction then enters the next entity specified in the storage definition statement.

In this particular example the value of the addition to the storage contents, i.e. the number of storage units represented by the entering transaction, is given by parameter 5 of the entering transaction. The value of the subtraction of the storage contents, i.e. the number of storage units represented by the packages which are removed by the leaving transaction, is given by the parameter 4 of the outgoing truck i.e. the transaction which initiates the "LEAVE" from the storage. The initial contents of the storage is 125 storage units. The capacity of the storage in terms of useable storage units is 400. These specifications are defined in the storage definition statement as follows:

```
SWAREHOUSE      -2, 400, 125, (4, 5 ), ( 4, 4 ), (1, 11 )
```

X_1 - storage #

X_2 - capacity

X_3 - initial contents

X_4 - storage contents addition attribute type

X_5 - storage contents addition attribute #

X_6 - storage contents subtraction attribute type

X_7 - storage contents subtraction attribute #

X_8 - next entity attribute type

X_9 - next entity attribute #

where

$(X_4, X_5) = (4,5)$ is the value contained in parameter 5, of the entering transaction,

$(X_6, X_7) = (4,4)$ is the value contained in parameter 4 of the leaving transaction, i.e. the one coming from the "LEAVE" control block, and

$(X_8, X_9) = (1, 11)$ is the constant 11.

Figure 2.19 Storage Definition Statement Examples

In figure 2.19 the specification of the storage definition statement was illustrated in the context of the total model building process. In these examples only the actual storage statement specification is illustrated.

STOR -1, 300, 300 , (1 , 6),(1 , 12),(1,200)

X_1 - storage # -1

X_2 - capacity

X_3 - initial contents

X_4 - storage addition per transaction

X_5 - the constant 6 units

X_6 - storage subtraction per transaction

X_7 the constant 12 units

X_8 - the next entity

X_9 the constant, entity # 200

STORAGE 100 -100, 1000,0, (5 25),(2,10), (3,3)

X_1 - storage #

X_2 - capacity

X_3 - initial contents = 0

X_4 - storage addition per transaction

X_5 counter #25

X_6 storage subtraction per transaction

X_7 function #10

X_8 next entity,

X_9 the variable #3

2.11 The Queue Statement

The general form of the queue statement is:

$Qa..ad..dx_1d..dx_2d...dx_3$

Q - begins in any column

a - is any alphanumeric character,

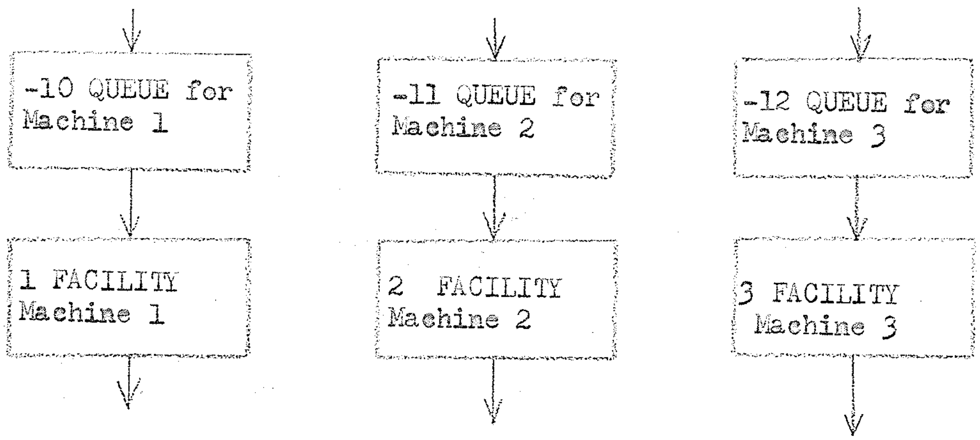
d - is any delimiter,

x_1 - is the queue entity number

x_2 and x_3 are the model attribute type and attribute number defining the next entity a departing transaction will enter.

Figures 2.20 and 2.21 illustrate the use of this statement. The queue is the simplest of entities. It is simply a waiting line of transactions which leave in the order they entered. Situations in which waiting lines exist are usually related to facilities which can process only one transaction at a time, while several may be requiring the service simultaneously.

Figure 2.20 Sample of Queue Definition Statement



The above diagram illustrates a model segment which is intended to compare the performance of three machines 1,2, and 3. Some of the most descriptive statistics relating to facility effeciency are those describing the average length of the queue, and the average time a transaction spent waiting to be serviced. Only be having transactions first join queues can such statistics be obtained. For the above three queues the following queue definition statements are defined.

QMACH1 - 10, (1,1)

x_1- queue # -10

x_2- next entity specification

x_3- the entity 1

QMACH2 -11, (1,2)

x_1- queue # -11

x_2- next entity specification

x_3- the entity 2

QMACH3 -12, (1,3)

x_1- queue # -12

x_2- next entity specification

x_3- the entity 3

Figure 2.21 Queue Definition Statement Examples

In figure 2.20 the specification of the queue definition statement was illustrated in the context of the total model building process. In these examples only the actual statement specification is considered.

QUE -100, (2,3)

x_1- queue # -100

x_2- next entity specification

x_3- the function # 3

Q -50, (3,2)

x_1- queue # -50

x_2- next entity specification

x_3- the variable # 2

2.12 Control Block Statements

The following sectors, 2.12.1 to 2.12.14 describe and illustrate the 14 control block statements.

2.12.1 Generate Control Block

The generate statements defines the system boundaries through which transactions enter the system being modeled. The general form of this statement is:

Ca..a x_1d.dx_2d..dx_3d..dx_4d..dx_5d..dx_6d..dx_7d..dx_8

x_1- control block number

x_2- 1 (generate control block)

x_3 and x_4 are the model attribute type and attribute number defining the transaction inter-arrival time,

x_5- is the inter-arrival modifier,

x_6- is the generation offset,

x_7- is the number of transactions to create, if zero transactions are created indefinitely

$x_8$and x_9- is the specification of the next entity.

Figure 2.22 illustrates the generation control block's use in a model.

Notes on the generation control block:

1. Arguments x_3 and x_4, the inter-arrival time specification can only refer to scalars or functions.

2. The inter-arrival time of the first transaction to leave a generate block is determined by evaluating the inter-arrival specification before the actual simulation begins. Therefore if the inter-arrival time is specified by a function whose independent variable is a parameter or counter, and since the initial values of all parameters and counters are zero, the functional value of zero will result as the inter-arrival time for the first transaction. If the interarrival time is zero it is charged to 1 for the first transaction.

3. If the inter-arrival specification (x_3 & x_4) is a function then a non-zero value of x_5, the inter-arrival modifier, will be multiplied by the functional evaluation, the result being the inter-arrival time.

4. All inter-arrival times are integer. However, if the situation in note 3 exists and the functional evaluation is a decimal number, transaction does not occur until after the multiplication of the modifier by the functional.

5. If the inter-arrival time (x_3 & x_4) is a constant and the inter-arrival modifier x_5 is non-zero then the distribution of inter-arrival times will be rectangular

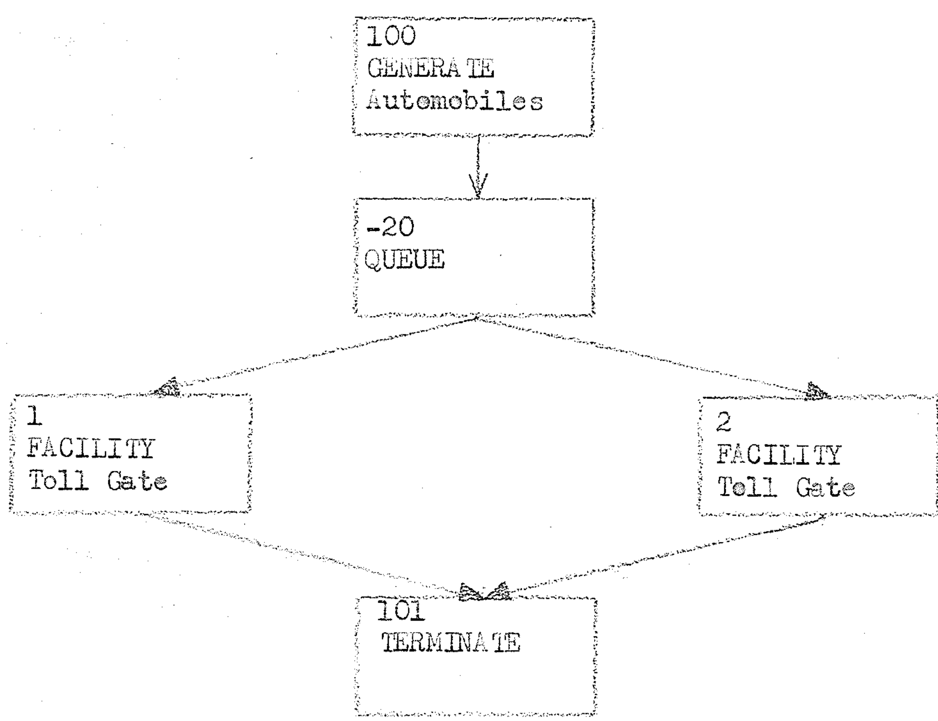
with mean x_4 and range to x_5. Since negative inter-arrival times are invalid x_4 must be greater than x_5. See figure 2.24. Example 2.

6. The inter-arrival time for a transaction is evaluated at the generation of the previous transaction.

7. The sixth argument (x_6) the generation offset determines the base time for the first transaction generation. Therefore if the evaluation of the inter-arrival time for the first transaction is 3 units and the generation offset is 10 units then the first transaction will be generated at 13 units of time.

Figure 2.22 Examples of Generate Control Block (#1)

Example 1



CGENERATE 100, 1, (2,9),0,10,225,1,-20

X_1 - control block # = 100

X_2 - control block type = 1

X_3 - transaction inter-arrival

X_4 - generation time= Function 9

X_5 - inter-arrival modifier = 0

X_6 - time of first generation from 0

X_7 - number of transactions to generate

X_8 - next entity

X_9 - specification

Example 2

Purpose: To illustrate the application of argument x_5, to the inter-arrival modifier, with arguments x_3 and x_4, the transaction inter-arrival generation time.

Consider the generate statement;

CGENERATE 90,1,(1,5),0 etc.

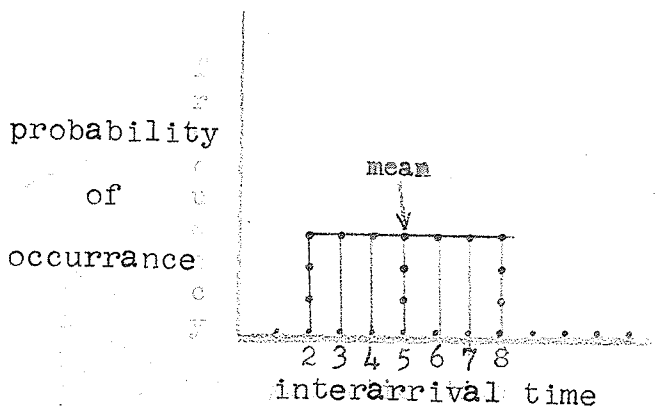
The inter-arrival specification (1,5) indicates that transactions will be generated every 5 minutes i.e. (1,5) = constant 5. The inter-arrival modifier, equal to zero has no effect on the inter-arrival times.

Consider the generate statement with x_5 not zero:

CGENERATE 90,1, (1,5), 3, etc.

This specification (1,5) and 3 indicates that the mean inter-arrival time is 5 minutes but the range of these times is 3 units on either side of the mean.

Thus the inter-arrival times will vary between 2 and 8 with mean 5. The specification further indicates, by convention, that the times within the range 2 to 8 have equal probability. The following diagram illustrates the distribution of such generation times.



2.12.2 Split Control Block

Transactions entering the split control block initiate the creation of one or more duplicate transactions called offspring transactions. The original or parent transactions and the offspring transaction may be directed to different next entities. The general form of this statement is:

Ca...xd..dx$_1$d..dx$_2$d..dx$_3$d..dx$_4$d..dx$_5$d..dx$_6$d..dx$_7$

where

X_1- is the control block number

X_2- = 2 the split control block

X_3 and X_4 are the model attribute type and attribute number defining the parent transaction next entity,

X_5- is the number of offspring to create

X_6 and X_7 are the model attribute type and attribute number defining the offspring transactions next entity.

Figure 2.23 illustrates the split control block incorporated into a model.

Notes on the split control block:

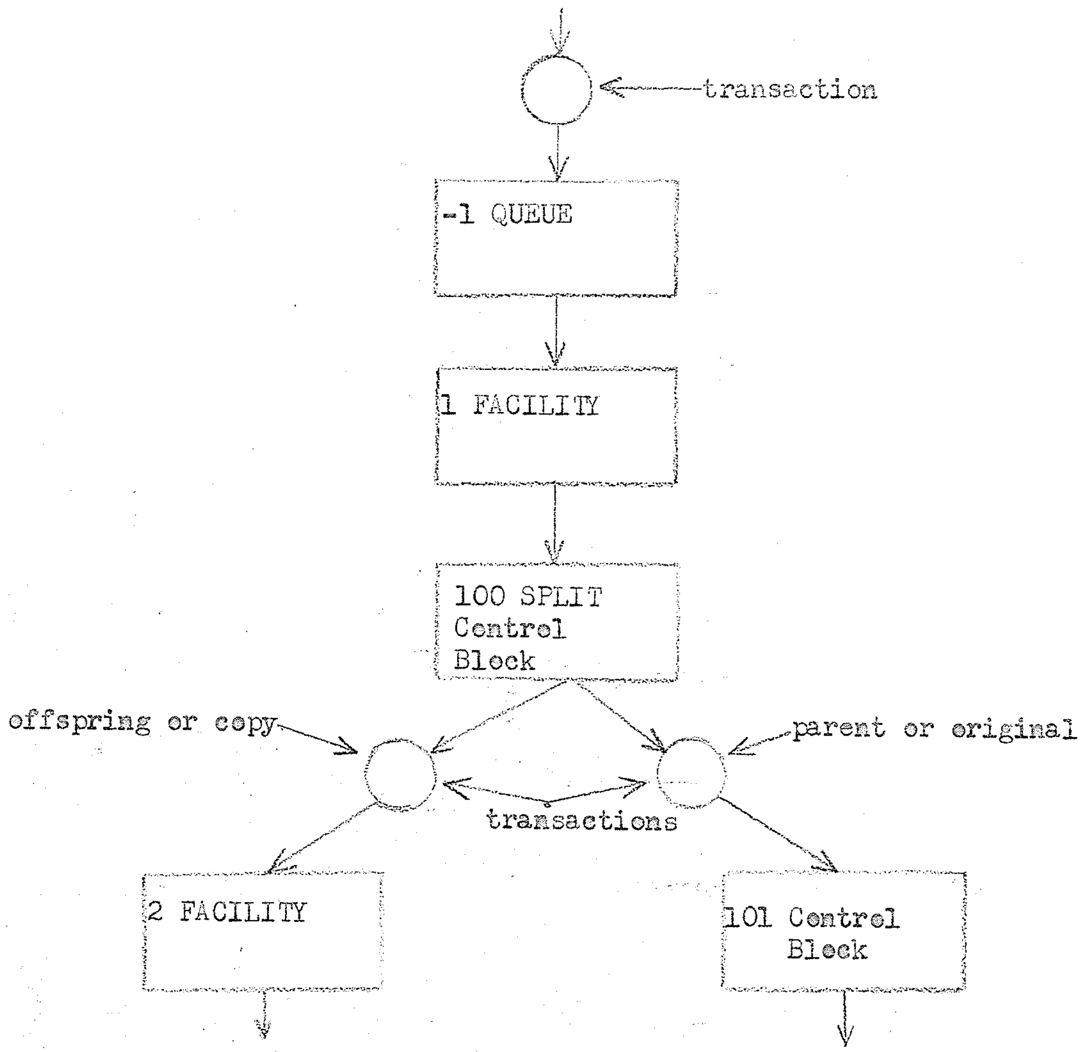
1. When a transaction enters a "split" control block the number of copies or offspring as specified in argument x_5 are created. The next entity for the parameter or original transaction is evaluated from arguments x_3 and x_4 and the original transaction directed to that entity.

2. When the parent transaction reaches a point in the model where it is on a hold position or if it has been removed from the simulation, the offspring transactions are processed.

3. The offspring processing is initiated by the evaluation of the offspring next entity for the first copy. This copy is then directed to this next entity and will continue through the model until it is either removed or placed in a hold status. In either case, when the processing of the first copy is stopped (permenantly or temporarily) the second copy is processed.

4. The processing of the second and each subsequent copy, up to x_5 copies, begins by the re-evaluation of the offspring next entity specification, and the movement of that copy through the model until its processing is stopped. The next copy is then treated in the same fashion.

Figure 2.23 Split Control Block



CSPLIT 100,2, (1,101) ,1, (1,2)

X_1 - control block #

X_2 - control block type

X_3 - parent next entity

X_4 - specification

X_5 - number of offspring to create

X_6 - offspring next entity

X_7 - specification

2.12.3 Assemble Control Block

The assemble control block performs the opposite function of the split control block. This block joins, unites or assembles separate transactions into one. The one transaction which continues when the specified number have been joined possesses the characteristics of the last transaction to enter the assemble block.

The general form of this statement is:

$Ca.ad.dx_1d...dx_2d...dx_3d...dx_4d...dx_5d...dx_6d$

x_1 - is the control block number,

$x_2 = 3$, the assemble control block,

x_3 and x_4 are the model attribute type and attribute number defining the number of transactions to combine.

x_5 and x_6 are the next entity specifications.

Figure 2.24 illustrates the assemble control block in a model.

Notes on the assemble control block:

1. When a transaction enters an "assemble " control block that control block may be in one of three states:

i) no transactions are in the block,

ii) less than n-1 transactions are in the block, where n is the number of transactions to assemble (x_3 and x_4)

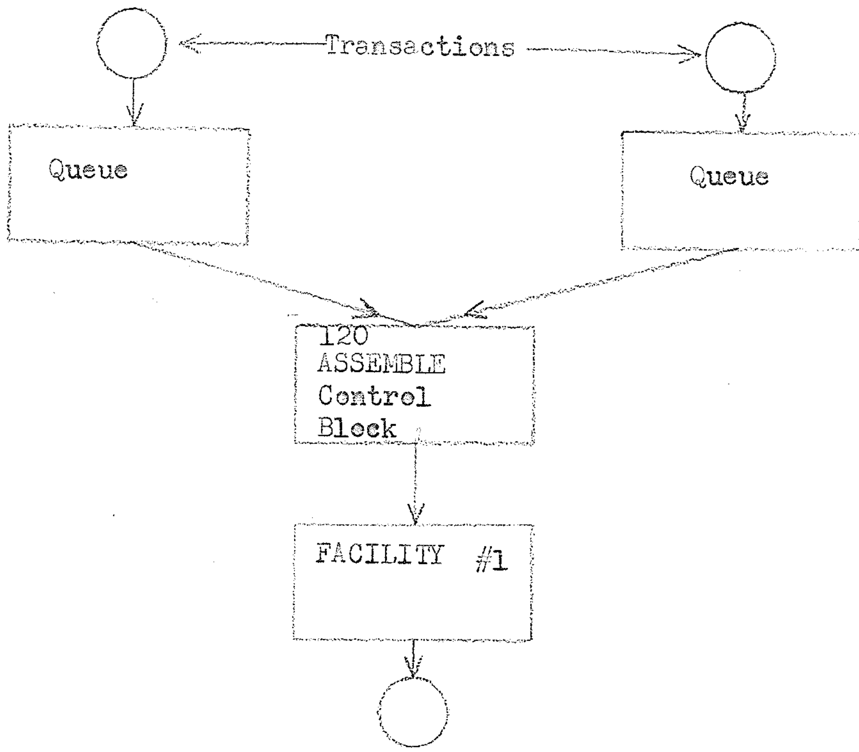
iii) there are n-1 transactions in the block.

2. If there are no transactions in the block, arguments x_3 and x_4, the number to assemble are evaluated and stored internally. This transaction is then removed from the system and the counter is reduced by 1.

3. If there have been less than n-1 transaction entries into this block the counter is reduced by one and the transaction removed from the system.

4. If there have been n-1 transaction entries then the current transaction represents the nth entry. The counter is reduced to zero, the next entity specification is evaluated and this transaction is directed to the evaluated next entity.

Figure 2.24 Assemble Control Block



CASSEMBLE 120,3, (1,2), (1,1)

X_1 - control block #

X_2 - control block type

X_3 - number of transactions to

X_4 - assemble-specification = 2

X_5 - next entity

X_6 - specification

2.12.4 The Facility Availability Control Block

This control block examines the facility specified. If the facility is available a "0" is placed in the designated counter. If unavailable a "1" is placed in the designated counter.

The format of this statement is:

$Ca..ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6$

x_1-is the control block number.

x_2= 4 the facility availability control block

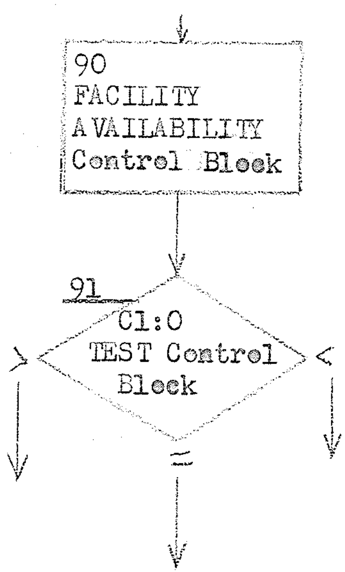
x_3= facility number to be examined,

x_4= counter number into which statue of facility will be placed,

x_5 and x_6 are the next entity specifications.

Figure 2.25 illustrates the use of this block.

Figure 2.25 Example of Facility Availability Control Block



CFACAVAIL 90,4,2,1,(1,91)

X_1 - control block # = 90

X_2 - control block type = 4

X_3 - facility to be examined = 2

X_4 - place status in counter # 1

X_5 - next entity specification

X_6 - entity # 91

2.12.5 The Gather Control Block

The gather control block is similiar to the assemble control block in the sense that several transactions enter this block before any leave. In this block when the designated number of transactions have entered the block all the transactions leave in the order they entered. Transactions are held in this block until the required number have arrived.

The general form of this block is:

$$Ca.ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6$$

x_1 - is the control block number

x_2 = 5

x_3 and x_4 are the model attribute type and attribute defining the number of transactions to gather,

x_5 and x_6 are the next entity specifications.

Figure 2.26 illustrates the use of this block.

Notes on the gather control block:

1. When a transaction enters a "gather" control block that control block will be in one of three status:

   i) no transactions are in the block,

   ii) less than n-1 transactions are in the block, where n is the number to be gathered,(x_3 and x_4 evaluation),

   iii) there are n-1 transactions in the block.

2. If there are no transactions in the block,arguments x_3 and x_4 are evaluated to yield the number of transactions to be gathered. This value is stored internally. This counter is reduced by one and the transaction is held in this block.

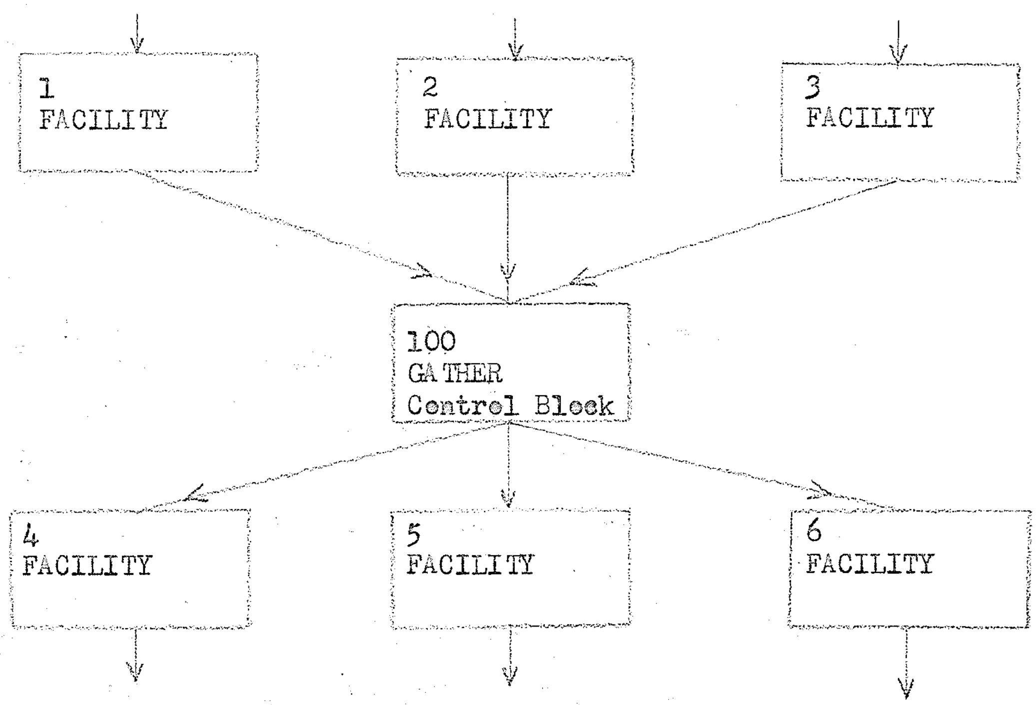
3. If there are less than n-1 transactions in the block the internal counter is reduced by one and this transaction joins the others being held in this block.

4. If there are n-1 transactions in the block then the entering transaction represents the nth entry. This transaction reduces the counter to zero and is placed with the other n-1 transactions. The hold state is now changed and the first transaction to enter the block is directed to the evaluated next entity.

5. The first transaction is then processed until it can proceed no further or is removed from the system.

6. The second and each subsequent transaction which entered the gather block is now processed. The next entity specification is re-evaluated and this transaction directed through the model until it is either stopped or removed, at which time the next transaction is processed in the same manner.

Figure 2.26 Example of the Gather control block



CGATHER 100, 5, (1,3), (4,2)

X_1 - control block # 100

X_2 - control block type

X_3 - number of transactions to gather;

X_4 - the constant = 3

X_5 - next entity specification;

X_6 - the value of parameter 2

Before any transaction will leave control block "100 ", three transactions will enter the "gather" block. When three have entered they will leave in the order they entered each being directed to the entity number in the transaction's second parameter.

2.12.6 The Queue Contents Control Block

This control block examines the desired queue and places its contents in the specified counter.

The general form of this statement is;

$$Ca..ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6$$

x_1 - is the control block number

x_2 = 6, the queue contents control block

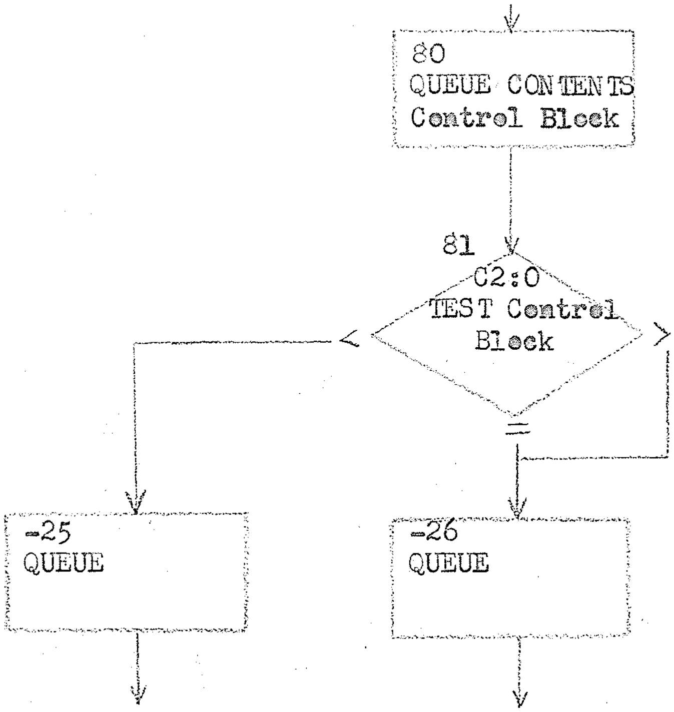
x_3 - is the queue number to be examined

x_4 - is the counter into which the queue contents will be placed,

x_5 and x_6 are the next entity specification.

Figure 2.27 illustrates the use of this block.

Figure 2.27 Example of Queue Contents Control Block



CQUECONT 80 , 6 , -25 , 2 , (1, 81)

X_1- control block #

X_2- control block type

X_3- queue to be examined

X_4- place contents in counter #

X_5- next entity specification

X_6- entity 81

2.12.7 The Assign Control Block

The assign control block modifies the contents of a parameter of the entering transaction.

The general form of this block is;

$$Ca.ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6d..dx_7d..dx_8d..dx_9$$

x_1 - is the control block number

x_2 = 7, the assign control block,

x_3 and x_4 are the model attribute type and attribute number defining the parameter to be modified,

x_5 - is the type of modification

= -1 subtraction

= 0 replace

= 1 addition

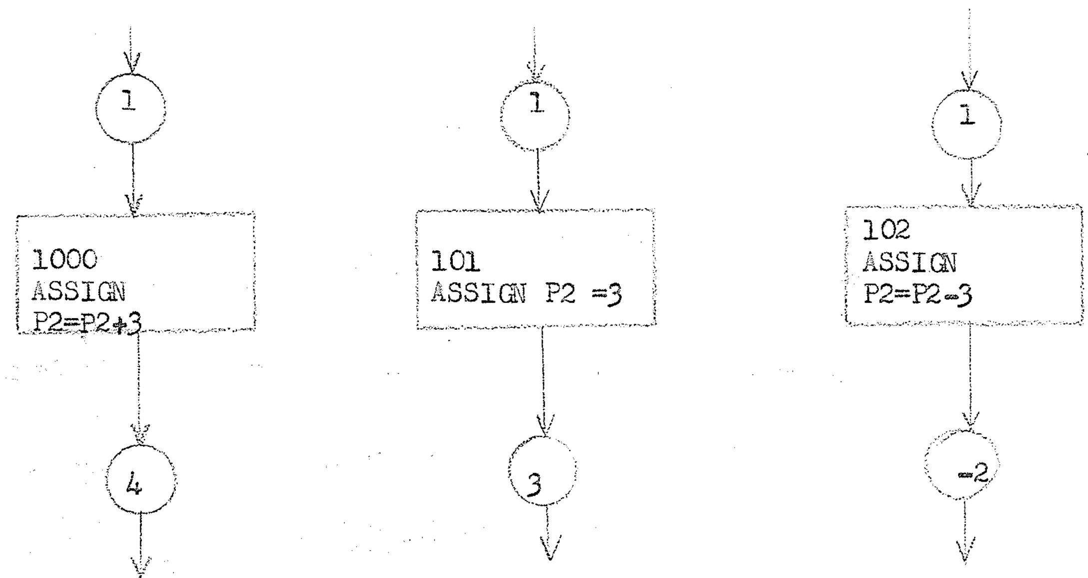
x_6 and x_7 are the model attribute type and attribute number defining the value of the modification.

x_8 and x_9 are the next entity specification.

Figure 2.28 illustrates the use of this block

Figure 2.28 Assign Control Block

Transactions
(present value of parameter 2 is 1)




```
CASGN        100, 7, ( 1,2 ), 1, (1,3 ), (1,-20)
CASGN        101, 7, ( 1,2 ), 0, (1,3 ), (1,-21)
CASGN        102, 7, ( 1,2 ),-1, (1,3 ), (1,-22)
```

X_1- control block #

X_2- control block type

X_3- parameter to be

X_4- altered

X_5- modification

X_6- value of modification

X_7- constant 3

X_8- next entity specification

X_9-

2.12.8 The Counter Control Block

The counter control block performs a modification on a counter.

The general form of this block is:

$$Ca..ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6d..dx_7d..dx_8d..dx_9$$

x_1 - is the control block number

x_2 = 8, the counter control block,

x_3 and x_4 are the model attribute type and attribute number defining the counter identity,

x_5 - is the type of modification,

= -1 subtraction

= 0 replace

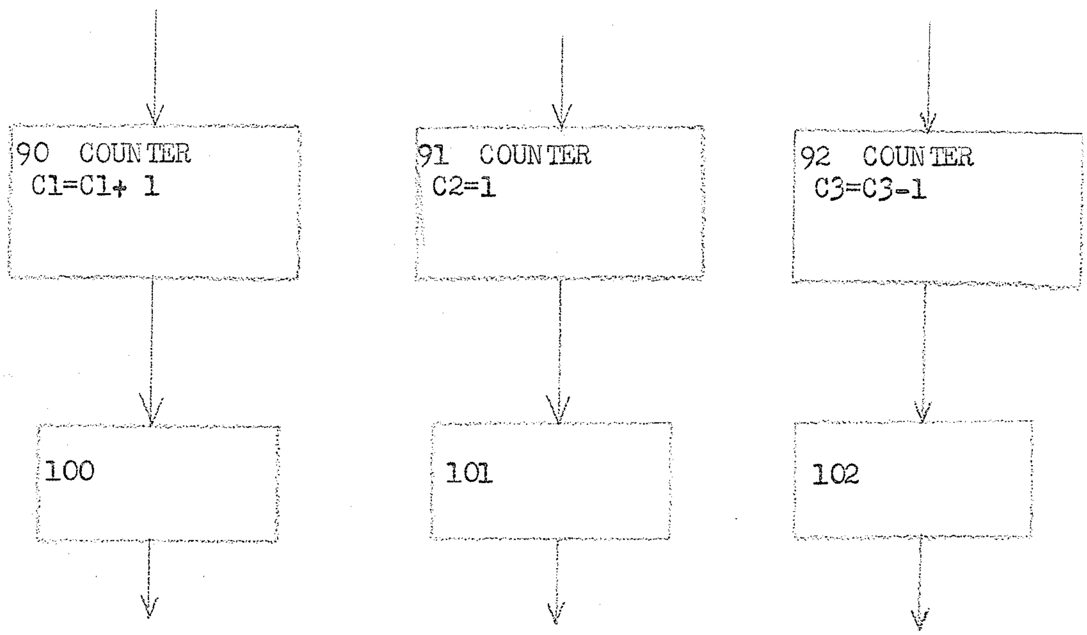
= 1 addition

x_6 and x_7 are the model attribute type and attribute number defining the modification value,

x_8 and x_9 are the next entity specification.

Figure 2.29 illustrates the use of this block.

Figure 2.29 Counter Control Block



Counter Number	Initial Value	Modified Value
1	5	6
2	5	1
3	5	4

COUNTER 90,8,(1,1),1,(1,1),(1,100)

COUNTER 91,8,(1,2),0,(1,1),(1,101)

COUNTER 92,8,(1,3),-1,(1,1),(1,102)

X_1 - control block #

X_2 - control block type

X_3 counter number

X_4 specification

X_5 - modification

X_6 value of modification

X_7 specification

X_8 next entity

X_9 specification

2.12.9 The Test Control Block

This block compares the value of two model attributes and directs the transaction to one of three next entities depending on the result of the comparison.

The form of this statement is:

$$Ca..ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6d..dx_7d..dx_8d..dx_9$$

x_1 - is the control block number

x_2 = 9 , the test control block

x_3 and x_4 are the model attribute type and attribute number defining argument A,

x_5 and x_6 are the model attribute type and attribute number defining argument B,

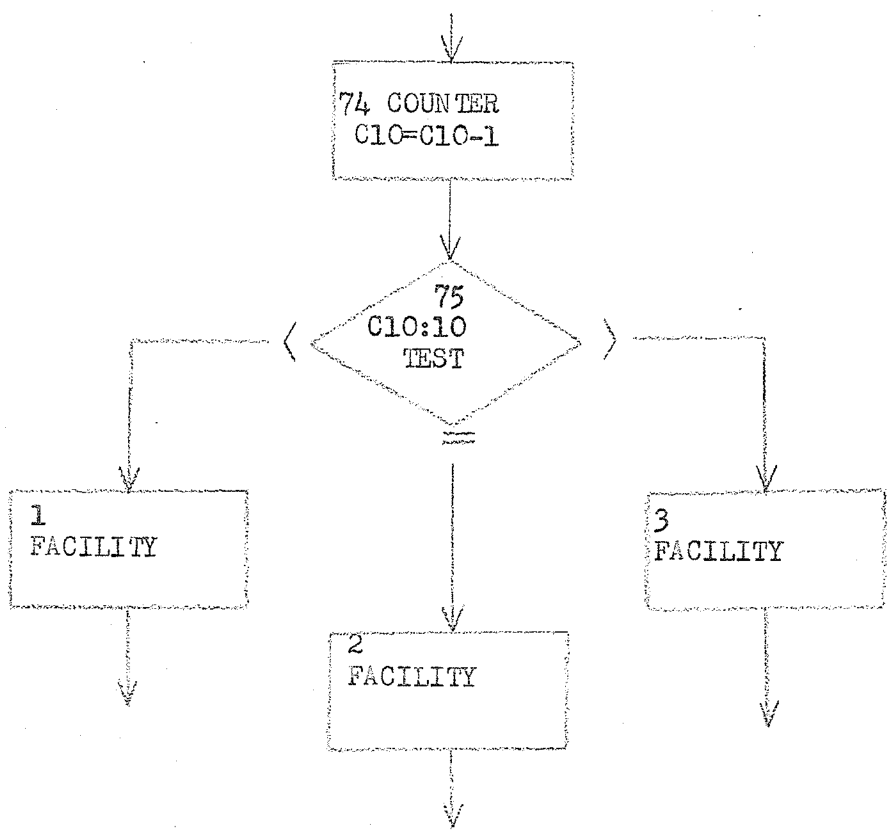
x_7- is the block number which the transaction will enter if $A < B$,

x_8- is the block number which the transaction will enter if $A = B$,

x_9- is the block number which the transaction will enter if $A > B$.

Figure 2.30 illustrates the use of this block.

Figure 2.30 Example of Test Control Block



CTEST 75,9,(5,10),(1,10),3,2,1

X_1 - control block # = 75

X_2 - control block type = 9

X_3 argument A

X_4 specification = C10

X_5 argument B

X_6 specification = 10

X_7 - next entity if $A < B$

X_8 - next entity if $A = B$

X_9 - next entity if $A > B$

2.12.10 The Leave Control Block

The leave control block initiates the subtraction of units from a storage. When a transaction enters this block it will immediately be transferred to the referenced storage. It will then continue on the appropriate path, the next entity being the one specified in the storage statement.

The form of this statement is:

Ca..ad..dx$_1$d..dx$_2$d..dx$_3$d..dx$_4$

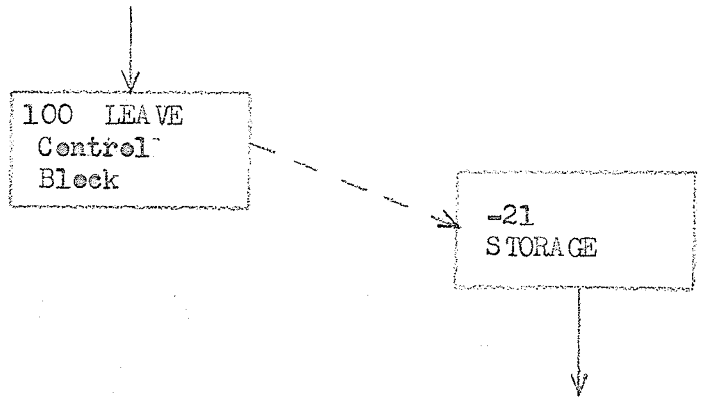
x_1 - is the control block number

x_2 = 10, the leave control block,

x_3 and x_4 are the attribute type and attribute number defining the storage from which units will be removed.

Figure 2.31 illustrates the use of this block.

Figure 2.31 Example of Leave Control Block



CLEAVE 100,10,(1,-21)

X_1 - control block #

X_2 - control block type

X_3 - storage specification

X_4 - the storage -21

2.12.11 The Remove Control Block

The remove control block defines a system boundary corresponding to the departure of transactions from the system. Transactions which enter this block are removed from the simulation.

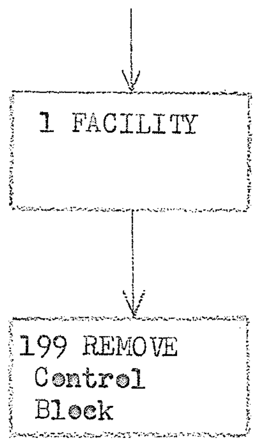
The general form of this statement is:

$Ca.ad..dx_1 d..dx_2$

x_1 - is the control block number.

x_2 = 11, the remove control block.

Figure 2.32 Example of Remove Control Block



CREMOVE 199, 11

X_1 - control block # 199

X_2 - control block type = 11

2.12.12 The Terminate Control Block

The terminate control block removes transactions in the same manner as the remove control block. In addition, however, a count is maintained of the number of transactions removed. The specified termination count is evaluated each time a transaction enters this block. If the number of transactions removed by this block is equal to or greater than the value of the termination count the simulation is terminated.

The form of this statement is:

$Ca..ad..dx_1d..dx_2d..dx_3d..dx_4$

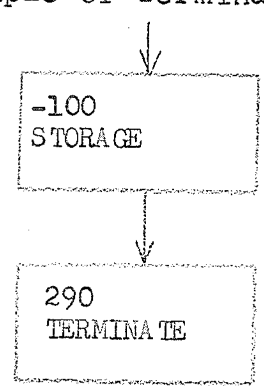
x_1- is the control block number

x_2= 12, the terminate control block

x_3 and x_4 are the model attribute type and attribute number defining the termination count.

Figure 2.33 illustrates the use of this block.

Figure 2.33 Example of Termination Control Block



CTERMINATE 290,12,(3,6)

X_1-control block #

X_2-control block type

X_3- termination count

X_4- = variable 6

Each transaction which enters the terminate block increases an internal counter. The "termination count ", x_3 and x_4 is then evaluated. If the value of variable 6 has been reached or surpassed by the counter then the simulation will be ended. Each terminate control block has its own internal counter.

2.12.13 The Print Control Block

This block initiates the printing of the five parameters, the twenty-five counters and the value of the clock when a transaction enters the block.

The form of this statement is:

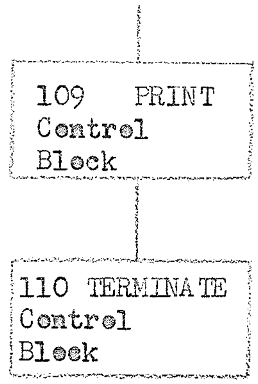
$Ca.ad..dx_1d..dx_2d..dx_3d..dx_4$

x_1- is control block number

x_2= 13

x_3 and x_4 are the model attribute type and attribute number defining the next entity.

Figure 2.34 Example of Print Control Block



CPRINT 109, 13, (1, 110)

X_1 - control block #

X_2 - control block type

X_3 next entity specification

X_4 the constant 110

2.12.14 The Storage Contents Control Block

This control block examines the desired storage and places its contents in the specified counter.

The form of this statements is:

$$Ca.ad..dx_1d..dx_2d..dx_3d..dx_4d..dx_5d..dx_6$$

x_1 - is the control block number

x_2 = 14, storage contents control block

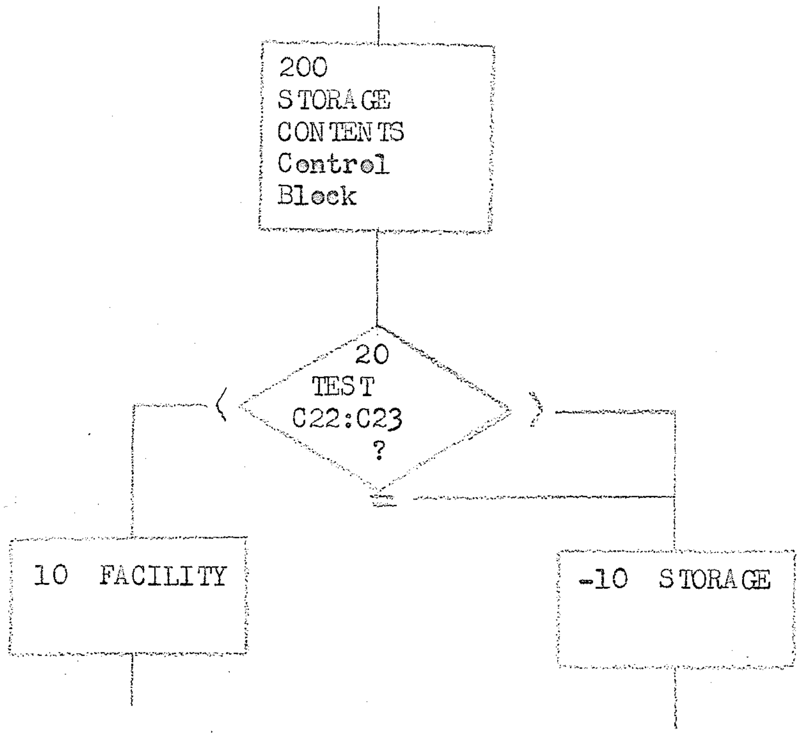
x_3 - is the storage number

x_4 - is the counter number

x_5 and x_6 are the next entity specifications.

Figure 2.35 illustrates the use of this block.

Figure 2.35 Example of Storage Contents Control Block



CSTØRCØNT 200 , 14 , -10 , 22 , (1 , 20)

X_1 - control block # = 200

X_2 - control block type

X_3 - storage to be monitored

X_4 - place contents in counter #

X_5 - next entity

X_6 - specification

2.13 Decision Statements

The decision statement is a special statement which allows the model builder to incorporate special types of next entity specifications. In section 2.6.1 "Next Entity Specification " the use of the model attributes in defining a next entity was illustrated. In this manner the specified model attribute type and its number are evaluated, the value being the number of the next entity.

The decision statement allows the next entity to be selected from several candidates, the choice being made accordingly to one of several predefined criteria.

The general form of this statement is:

$Da..ad..dx_1d..dx_2d..dx_3d..dy_1d..dy_2d..dy_k$

x_1- is the entity number at which the decision statement will be evaluated, this entity must be the same one which has specified its next entity values as x_2 and x_3 of the decision statement

x_2- specifies the entity type of the decision candidates

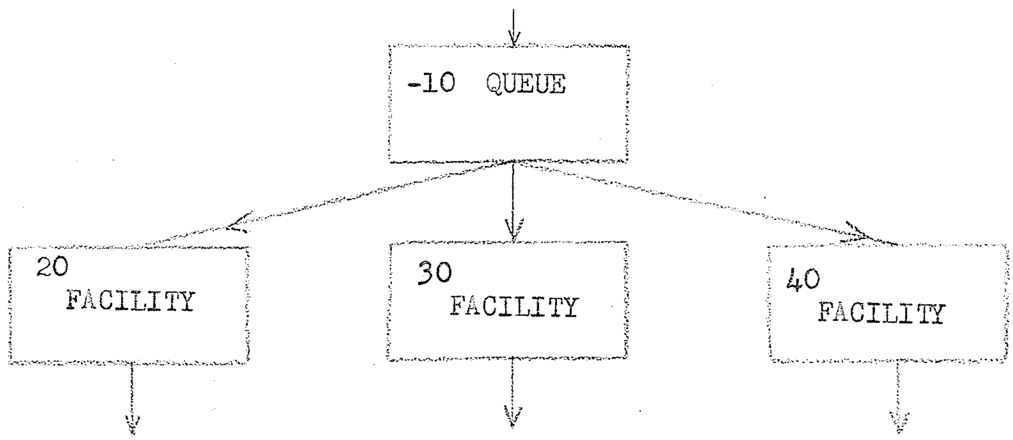
x_3- specifies the decision rule

y_1 to y_k are the "k" candidates to be analysed according to the rule specified in x_2 and x_3. All the candidates must be of the same entity type as specified in x_2. i.e. facility, storage, queue.

Figure 2.37 defines the decision rules for each entity type and their meaning.

Figure 2.36 illustrates the use of the decision statement in next entity specification.

Figure 2.36 Example of the Decision Rule for next entity specification



Consider the above model segment. Transactions leaving the queue may enter any one of the facilities 20, 30 or 40 depending on the next entity specification in argument x_2 and x_3 of the queue. By using the model attributes in the next entity specification a parameter, a function, a counter or a variable might be employed to vary the next entity over the values 20, 30 and 40.

In many instances, however, the model builder wishes the next entity decision to be based on the status of the alternative next entity candidates. For the facility two possible characteristics are provided. These are the facility utilization and the facility availability. Therefore the model builder can define the next entity choice to be one of three criteria; namely:

1. choose the first available facility.
2. choose the least utilized facility.
3. choose the highest utilized facility.

To incorporate one of these concepts into a model the next entity specification of the entity at which the decision must be made is defined to be the desired rule type. Then in a separate "decision rule" model definition statement the decision rule is defined. Each of the three facility criteria are defined below including the appropriate queue definition statement.

1. Choose the first available facility

QUE -10, (6,1)

X_1 - queue #

X_2 - next entity type

X_3 - next entity number

DRULE -10,(6,1),(20,30,40)

X_1 - entity at which choice is to be made

X_2 - decision rule

X_3 - type and number

Y_1, Y_2, Y_3 - next entity candidates

2. Choose least utilized facility

QUE -10, (6,2)

X_1 - queue #

X_2 - next entity

X_3 - specification

DRULE -10,(6,2),(20,30,40)

X_1 - entity at which choice is to be made

X_2 - decision rule

X_3 - type and number

Y_1 -

Y_2 - next entity candidates

Y_3 -

3. Choose the highest utilized facility

QUE -10, (6,3)

DRULE -10, (6,3), (20,30,40)

Figure 2.37 Next Entity Selection by Decision Statement

Entity type (x_2)	Rule No. (x_3)	Rule Description
6 - Facility	1	first available facility
	2	lowest utilized facility
	3	highest utilized facility
7 - Storage	1	lowest utilized storage
	2	highest utilized storage
	3	least full storage
	4	most full storage
8 - Queue	1	shortest queue
	2	longest queue

Note 1.

For the specification 6,1, choose the first available facility, the facilities are analysed in the order of specification in the decision statement. If none of the facilities are available the first one will be occupied when it becomes available.

Note 2.

If the analysis of the candidates results in two or more entities being equally valid for selection the one specified to the left of the others will be chosen.

2.14 Variable Statements

The variable statement is used to define a mathematical relationship between certain model attributes. The general form of this statement is:

V_i = expression

i: is the variable number 1 to 20 (variables must be numbered consecutively beginning at 1).

V can begin in any column,

= must occur immediately after the variable number.

expression is a mathematical expression composed of valid characters arranged according to a defined syntox.

Note: the occurance of a blank in any position to the right of the "V_i =" indicates the end of the variable definition.

The expression has the following elements:

a) number eg. 123.4, -231 , .67

b) mneumonic attribute names

P	= 1,2,...5	parameter
V	= 1,2,...20	variables
C	= 1,2,...25	counters

c) operators

*	multiplication	+	addition
**	power	-	subtraction

/ division

d) brackets:

(open bracket

) closed bracket

e) special functions

LN	-- natural log
EXP	-- exponential
SIN	-- sine
TAN	-- tangent
COS	-- cosine
LOG	-- log to the base 10
SQRT	-- square root
TANH	-- hyperbolic tangent
SINH	-- hyperbolic sine
COSH	-- hyperbolic cosine
ATAN	-- arctangent
ARSIN	-- arcsine
ARCOS	-- arccosine
COTAN	-- cotangent

The following rules apply in the formation of expressions.

1. Embedded blanks in the expression are invalid

   V1 = 23 * C2   valid

   V1 = 2 3 * C 2   invalid

2. Mnemonics cannot contain embedded blanks.

   valid C2 P1 V2

   invalid C 2 P 1 V 2

3. The number of open brackets must equal the number of closed brackets.

4. The first bracket used in an expression must be an open bracket.

5. The last bracket used in an expression must be a closed bracket.

6. Special functions must have their argument enclosed in brackets.

   valid V1 = LN (argument)

   invalid V2 = EXP argument

7. The argument of a special function may be any valid expression.

8. Mnemonics , constants and special functions must be separated by eperators.

```
valid    V10 =C2*23
         V10 =C2*C3
         V10 =C2*LN(C5)
invalid  V11 =C2C4
         V12 =234C1
         V13 =LN(C5)C6
```

Some valid variable definition statements are:

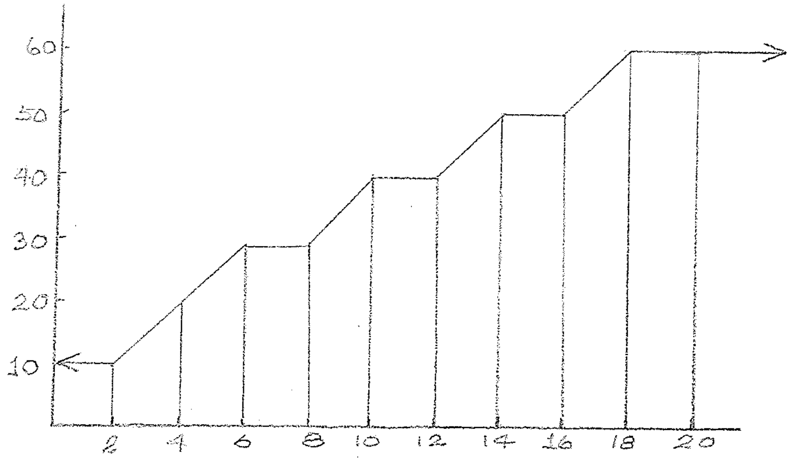
```
V1 = (LN(C2/C3)*P2**C05(P3/632.)
V2 = C4 + C5 + C6/(EXP(P1))
V3 = V2 + V1 *10
V4 = V3-V2 + (COS(EXP(C2) + P2)
V5 = ATAN(P2)
V6 = LOG((P2/P3)*P5)
```

2.15 The Function Statement

The function statement is used to define a relationship between two variables, one dependent the other independent. A function is used when the relationship between these two quantities is known graphically. The x and y points from the two dimensional graph are input to the simulation system by means of the FUNCTION model definition statement. The independent variable is also part of the definition of this statement.

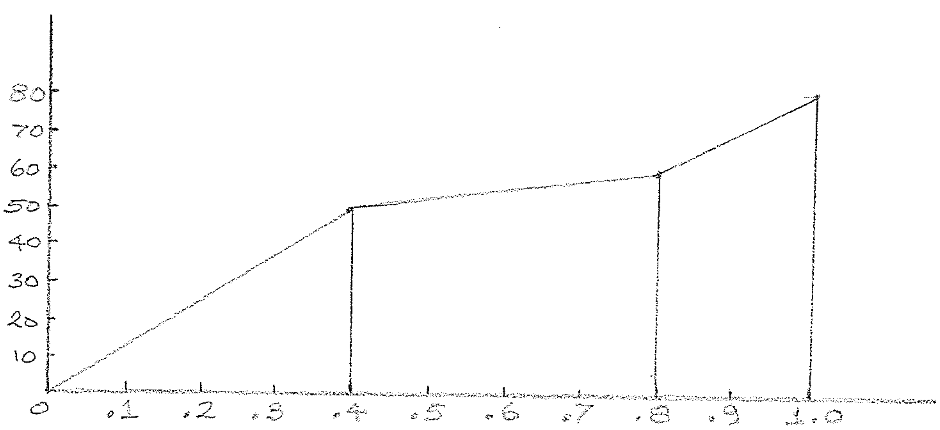
The independent variables allowable for functions are random numbers, counters and parameters. Depending on the relationship between the variables the defined points may be considered as part of a continuous curve or a discrete relationship. This property is also part of the definition statement. Figures 2.38 to 2.40 illustrate the formation and interpretation of continuous and discrete functions.

Figure 2.38 A Continuous Function



independent axis value	functional value
0	10
1	10
2	10
3	15
4	20
5	25
6	30
7	30
8	30
9	35
10	40
12	40
15	50
20	60
25	60

Figure 2.39 A Continuous Function



Function is defined by the points

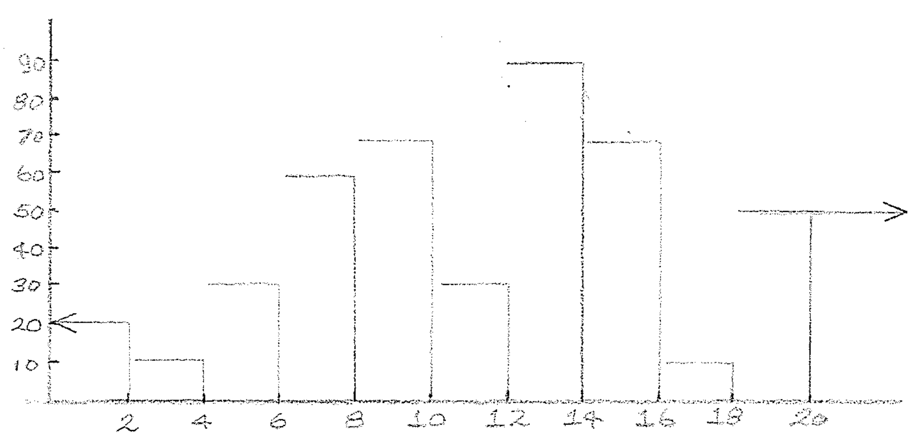
(0,0) , (.4, 50) , (.8,60) , (1.0,80)

Random Variable Value	functional value
.1	12.5
.2	25
.4	50
.5	52.5
.6	55
.7	57.5
.8	60
.9	70
1.0	80

Note: Since the independent variable is uniformly distributed for the interval [o,1] then functional values less than fifty will occur 40% of the time, functional values between 50 and 60 will occur 40% of the time and functional values between 60 and 80 will occur 20% of the time.

Figure 2.40 A Discrete Function

Figure 2.42 illustrates the nature of a discrete function. A table of x values and the corresponding functional values is presented also.



The function definition points are (2,20),(4,10), (6,30) , (8,60) , (10,70) , (12,30) , (14,90) , (16,70) , (18,10) , (20,50)

Functional values

f(1)= 20, f(2)= 20, f(3)= 10, f(4)= 10, f(5)= 30, f(6)= 30, f(7)= 60, f(8)= 60, f(9)= 70, f(10)= 70, f(11)= 30, f(12)= 30, f(13)= 90, f(14)= 90, f(15)= 70, f(16)= 50, f(17)= 10, f(18)= 10, f(19)= 50, f(20)= 50, f(21)= 50, f(22)= 50, The general form of the Function definition statement is:

$$\left\{\begin{matrix} FU \\ FN \end{matrix}\right\} \ a..ad..dn_1d..dn_2d..dn_3d..dn_4dkd..dx_1d..dy_1d.. \\ dx_2d..dy_2d..dx_kd..dy_k$$

where

FU or FN as the first two letters of the statement,

a- is any alphanumeric character,

d- is any delemeter,

n_1-is the function number less than 20.

n_2-is the independent argument type

n_2= 1 implies parameter

n_2= 2 implies random number

n_2= 3 implies counter

n_2= 4 implies clock time

n_3-is the identity of independent argument specified by n_2.

n_2= 1 implies $1 \leq n_3 \leq 5$

n_2= 2 implies $1 \leq n_3 \leq 8$

n_2= 3 implies $1 \leq n_3 \leq 25$

n_2= 4 implies $-1000 \leq n_3 \leq 1000$

if n_2= 4 then n_3 is not used.

n_4- is the function type

n_4= 1 implies a continuous function

n_4= 2 implies a discrete function

k- is the number of $x_i y_i$ points to be defined.

x_i, y_i $i = 1, k$ are the values representating the graphical points. x_i and y_i are real numbers.

Notes: on the Function

1. n_1 to n_4 and k are integer values.

2. Functions are referenced by the normal integer pair which is used for all model attributes, example (2,2) refers to function 2.

3. When a function is referenced in a model definition statement that function is evaluated at that time and this value passed to the appropriate model definition statement.

Figures 2.41 illustrates some function definition statement examples.

Figure 2.41 Function definition statements

FN1 1,3,1,2,5,(0.0,0.0/1.5,6/2.3,7/3.1,2/10,10)

N_1 - function number

N_2 - independent argument type (counter)

N_3 - counter number

N_4 - function type

K - number of X Y points

$X_i Y_i$ - points from graph

2.16 Student Considerations

This section is written for the purpose of cautioning the student with simulation. Although this system is elementary in nature and relatively limited in its scope it is possible to use it for difficult simulation problems. The novice, without realizing it, may easily include such complex situations in a model since on the surface they may appear straight forward. One class of situations often found in a model is the occurance of large stochastic elements. Consultation with an statistican is suggested on all models that the introductory student builds.

CHAPTER III

SIMULATION EXPERIMENTS

3.1 Introduction

In this chapter three sample simulations are presented. These descriptions include the statement of the problem, a description of the model building procedures, preparation of the model definition statements, simulation statistics and an interpretation of the results. Each simulation will be discussed under the following headings.

1. Statement of the problem,
2. Model Building Procedure,
3. Simulation Output,
4. Interpretation of Results.

The "Model Building Procedure" will follow the five stages of model building defined in section 2.41 and redefined here:

The Model Building Process

Stage 1: Statement of a problem in general terms .
Definition of gross system boundaries.
Statement of output needed to solve the problem.

Stage 2: Statement of initial assumptions.
Definition of static and dynamic system structures.

Construction of minimal system model.

Assessment of assumptions in the light of Stage 1 goals.

Stage 3:Determination of input data requirements and availability. If input data required are not available, modify assumptions and model structure by returning to Stage 2.

Stage 4: Determination of output possibilities.

If output is insuffucuent, modify assumptions and model structure by returning to Stage 2.

Stage 5: Prepare precise specifications for final model.

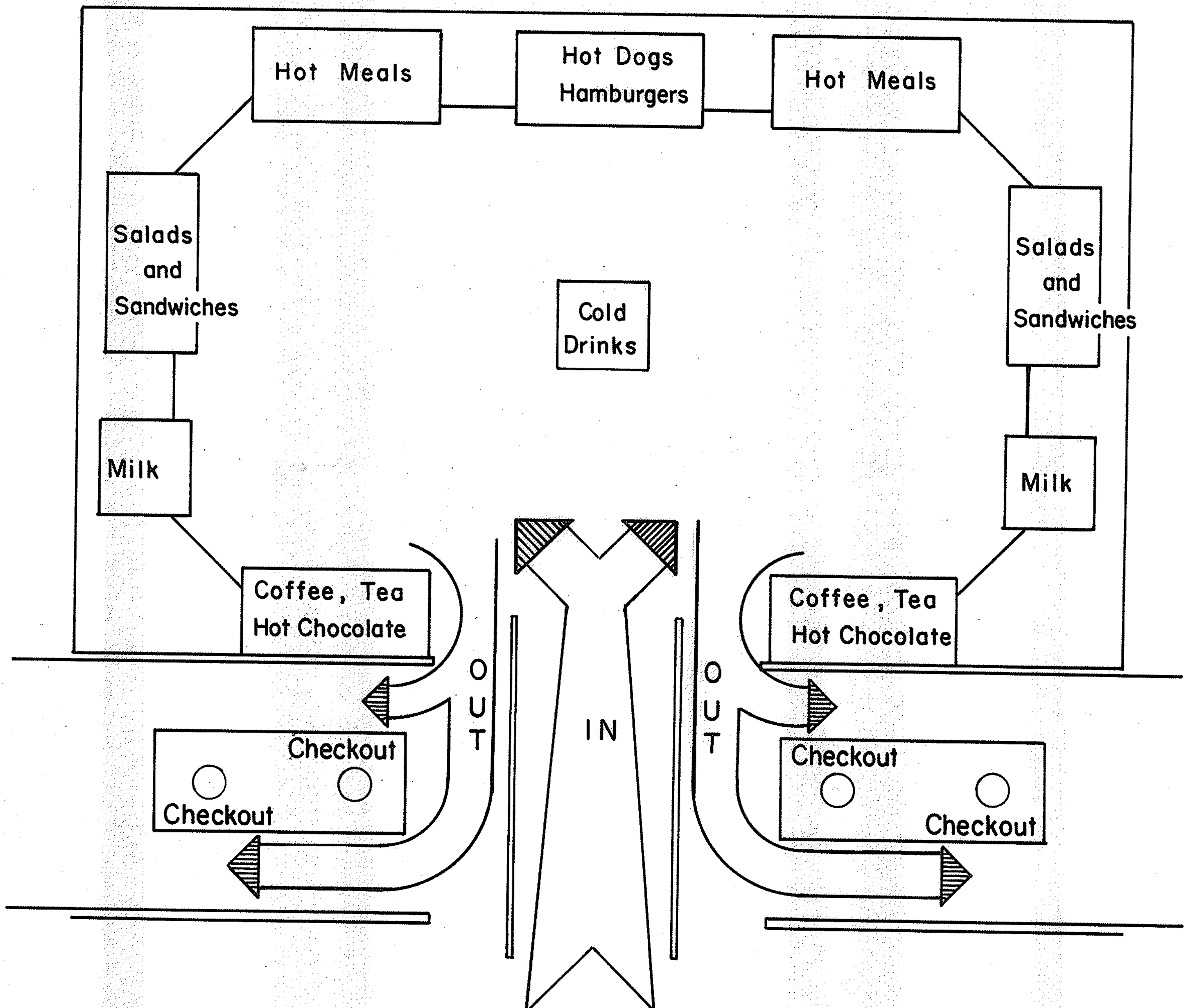
Select a modeling and programming language.

Reassess the implications of all assumptions for the future. Prepare a detailed plan for use of the model.

3.2 Example 1

3.2.1 Statement of the Problem

In this example the situation analysed is that of a cafeteria. Figure 3.1 illustrates the position of the various food counters, the checkout counters and the queues leading to these facilities. The purpose given to this simulation is simply that of studying the performance of this system. Hypothetically, it may be considered that this cafeteria is in the design stage and this simulation is part of the analysis of its feasibility.



Physical Cafeteria
Figure 3.1

3.2.2 Model Building Procedure

Stage 1

The problem was stated generally in section 3.1.1 .

The system boundaries may be defined in the following manner: The system begins where customers enter the first waiting line, it ends where a customer leaves one of the checkout counters.

The output required to solve the problem are counters' utilizations, queue lengths, and the total number of customers in the system.

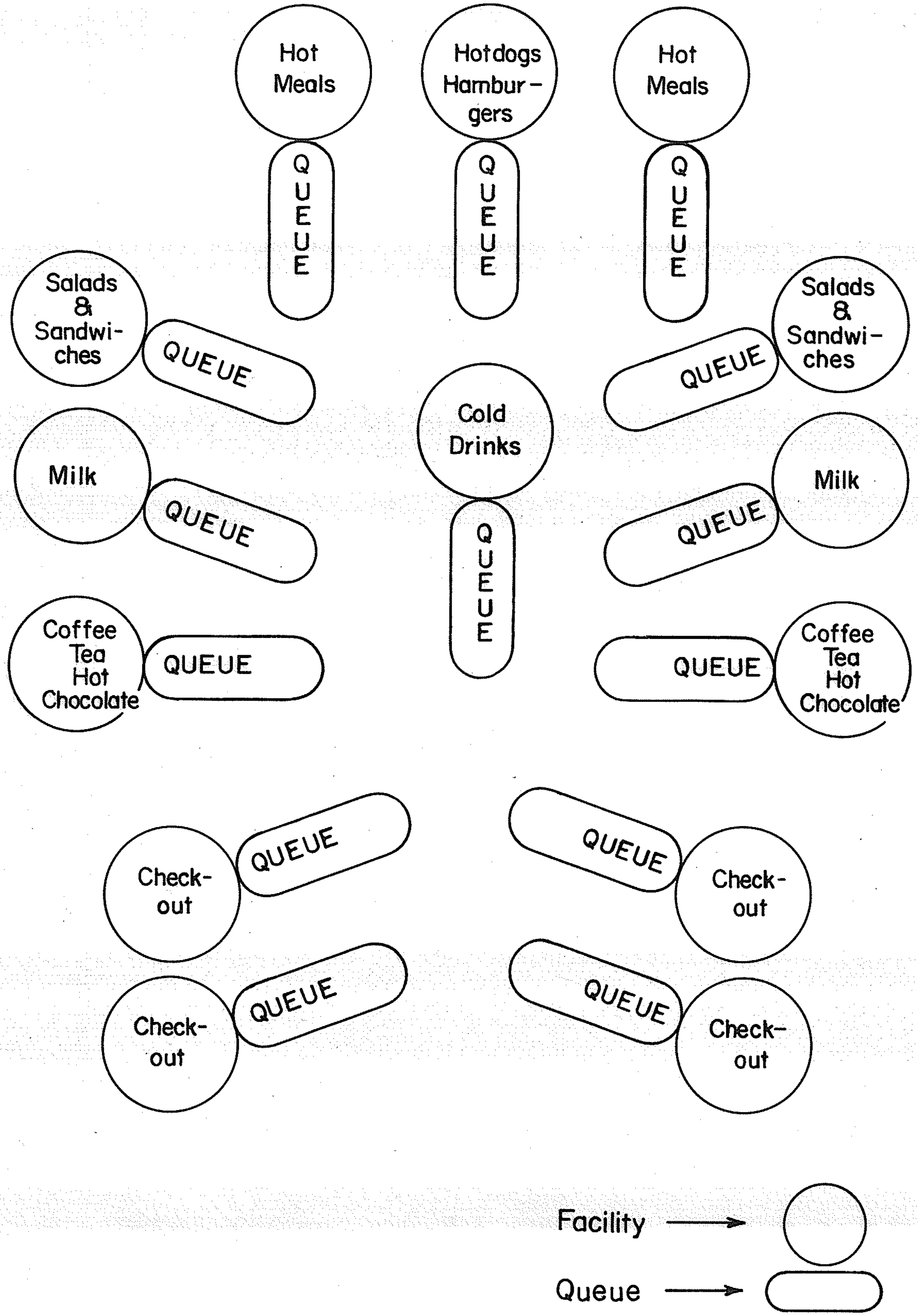
Stage 2

One of the first assumptions made for this situation is that the physical structure of figure 3.1 becomes the representation of queues and facilities defined in figure 3.2.

The dynamic structures or transactions in this system are the customers entering, moving from counter to counter then leaving the system.

The static structures in this system are the various food counters and, the checkout isles. These become facilities in figure 3.2

In the construction of the minimal system model the model builder attempts to represent the real-world system with the tools at his disposal. Since it is



Cafeteria Model I

Figure 3.2

the purpose of this presentation to illustrate the use of the simulation system described herein, its elements and statements are the so called tools with which the model is constructed. Initially the model builder may find the problem of representing the elements of the real world system by elements of an imaginary building system, challenging. As the building system becomes more familiar, as these examples are studied and as simple real systems are modelledan intuition develops. This intuition facilitates the process of constructing the elements of the modeling system to represent the system being simulated. Another point to note is that there exist an infinite number of combinations and configurations of the elements of a modeling system, all which adequately would simulate the same system. One such model for the cafeteria system is presented in the following paragraphs.

Figure 3.2 being the first approximation of the cafeteria has the following assumptions:

1. Each of the food counters is represented by a facility which is capable of serving one customer at a time. In the actual system at some counters two or more customers may serve themselves within a certain time interval. To reconcile this difference the processing time of the facility must be such that in a given interval of time the same number of customers are served in the model as in the actual system.

2. The groups of customers waiting at any counter generally cluster around the counter in the actual system. In such a cluster the first to join the waiting line may not necessarily be the first to be served. However, in the model the queues representing these waiting lines do follow the first in, first served pattern. Since the required outputs do not include the analysis of individual customers through the system but rather average customer movement this appears to be a valid approximation.

Another aspect of the actual system yet to be modeled is that of the movement of customers from counter to counter and then eventually to the checkout isles. Observation of the cafeteria has led to the following conclusions:

1. When a customer enters the system he will choose any of the six food types with probability.
2. Having made his first selection the probability that he will choose any of the six food types or that he will leave is equal.
3. If after the second selection if he is still in the system he repeats the actions of # 2.
4. The number of customers which go to four or more counters before exiting is negligible. Thus the fourth selection is always one of the checkout isles.

5. The checkout isle chosen is that with the shortest lineup or if there are no lineups then the closest one is chosen.

To represent these movement patterns within the system the flowchart described in figure 3.3 was defined.

The combination of figure 3.2 and 3.3 represent the minimal system model.

The assumptions were assessed in the light of Stage 1 goals and were found to be reasonable.

Stage 3

The determination of the input data requirements led to the following requirements:

1. Mean interarrival time for customers entering the system
2. Facility processing time for each food facility and checkout isle.

Stage 4

Determination of the output possibilities led to the following conclusions:

1. Facility utilization would be available
2. Queue lengths and averages would be available
3. The number of customers in the system would not be available.

To reflect the number of customers in the system as incremental counter was added to the model flowchart. This counter would be incremented by one when a transaction was generated and decremented by one when a transaction enters the checkout area.

Figure 3.3

Transaction Movement

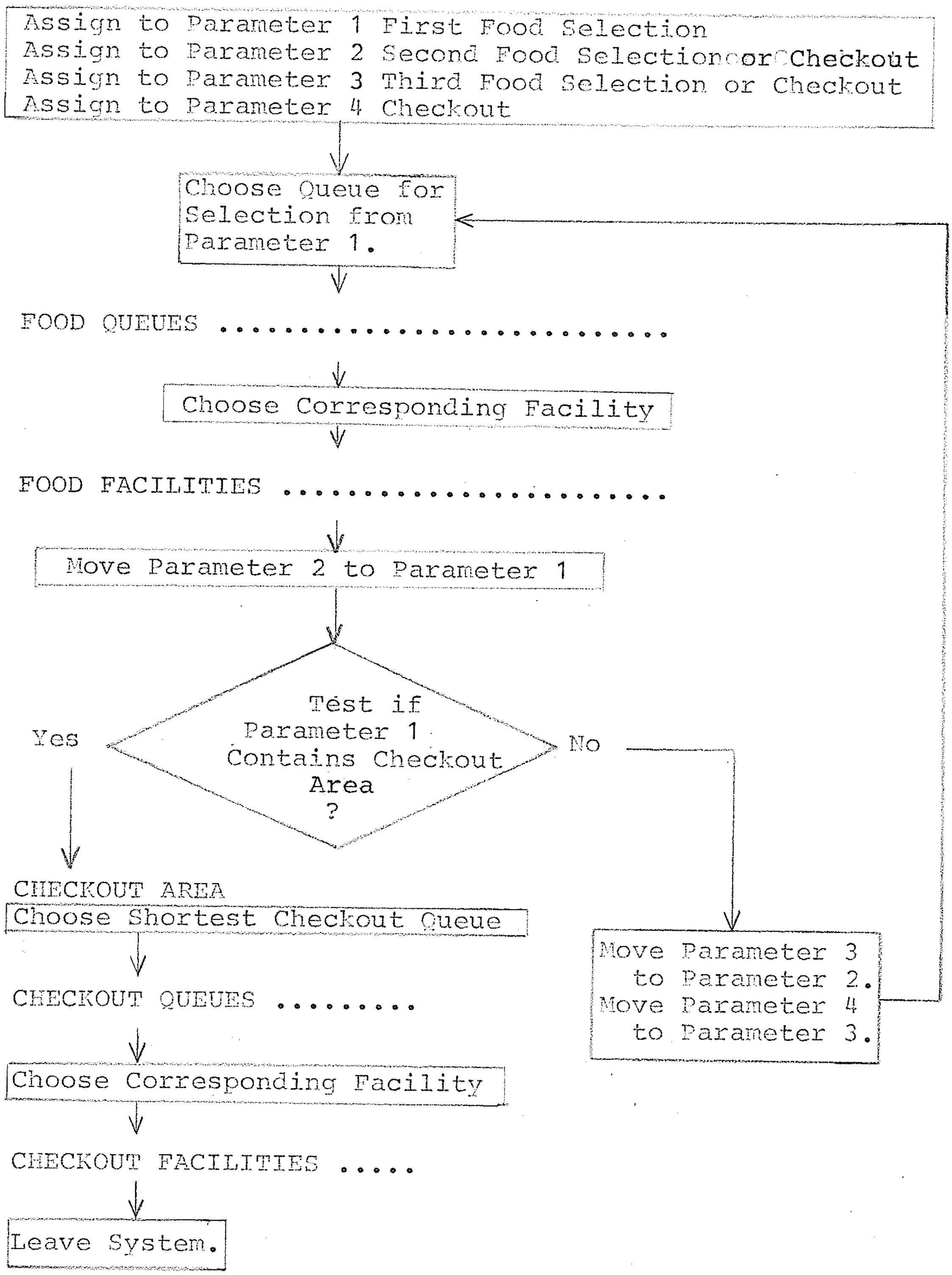


Figure 3.4

Generation and Facility Processing Times

-Facility Name	Mean Processing Time	Facility Numbers
coffee,tea, hot chocolate	20 seconds=2 units	10,11
milk	20 seconds=2 units	12,13
cold drinks	20 seconds=2 units	14
salads, sandwiches	20 seconds=2 units	15,16
hot dogs, hamburgers	30 seconds=3 units	17
hot meals	50 seconds=5 units	18,19
checkout isles	40 seconds=4 units	20,21,22,23

Generation Interarrival Time = 20 seconds ≑ 2 units.

Note:
The unit of time for this simulation is ten seconds.

Figure 3.5

Model Diagram

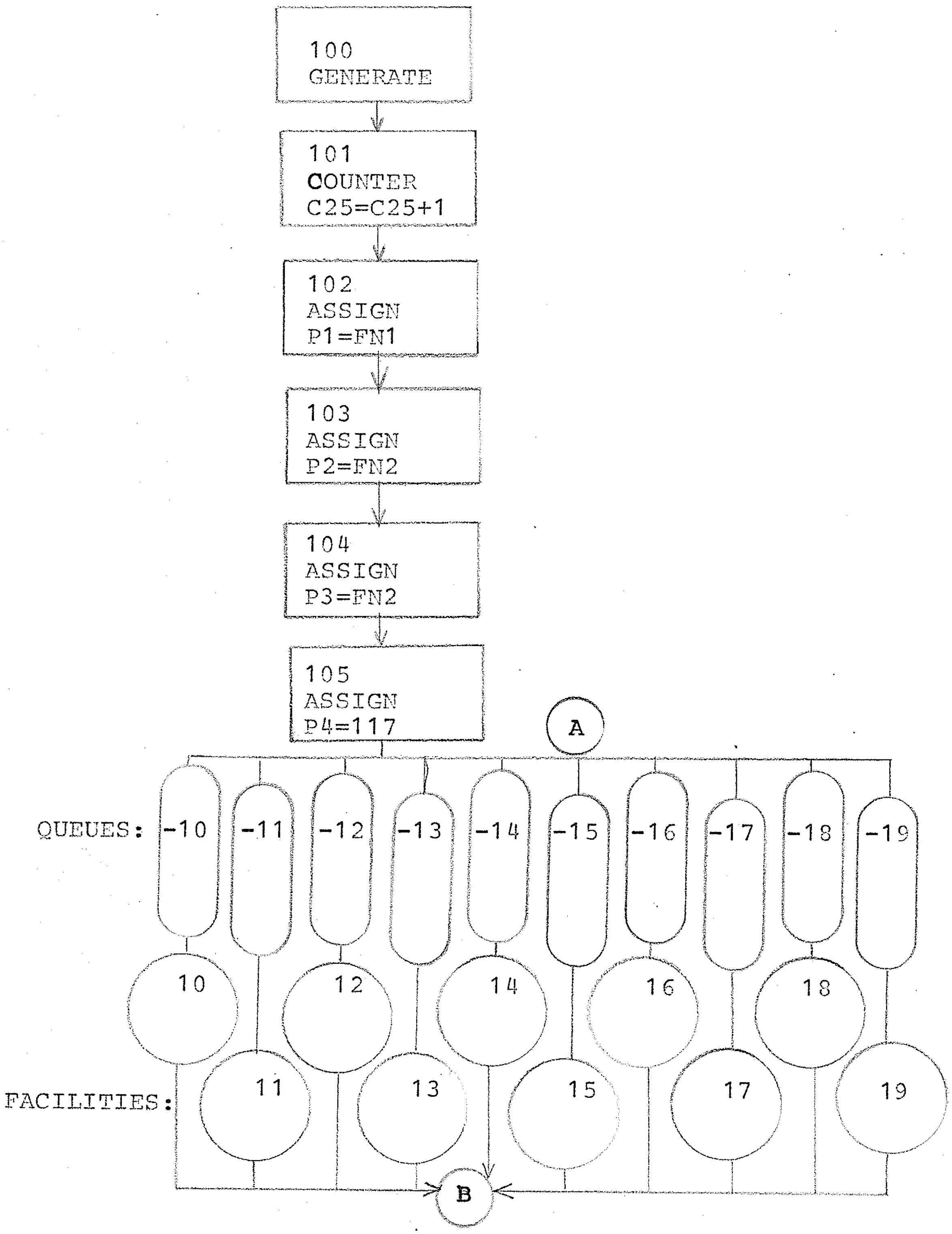


Figure 3.5 continued

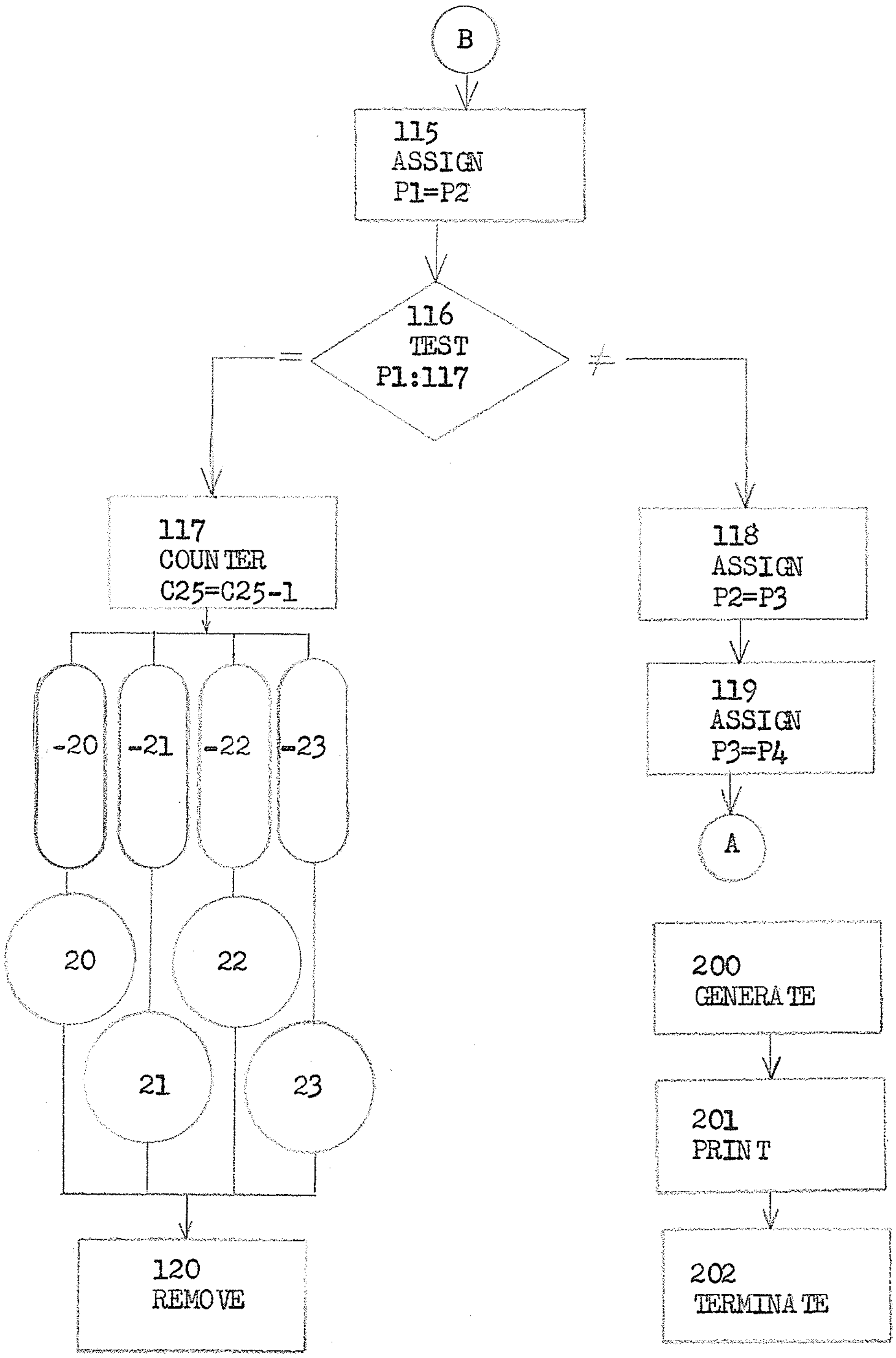


Figure 3.6 Example 1: Model Definition Statements

```
CGEN 100,1,(1,2),0,0,1000,(1,101)
  COUNTR 101,8,(1,25),1,(1,1),(1,102)
  CASGN 102,7,(1,1),0,(2,1),(1,103)
  CASGN 103,7,(1,2),0,(2,2),(1,104)
  CASGN 104,7,(1,3),0,(2,2),(1,105)
  CASGN 105,7,(1,4),0,(1,117),(4,1)
Q10,-10,1,10    Q11 -11,1,11    Q12 -12,1,12    Q13 -13,1,13
Q14 -14 1 14    Q15 -15,1,15    Q16 -16,1,16    Q17 -17 1 17
Q18 -18 1 18    Q19 -19,(1,19)
  F10,10,(1,2),(1,115)       F11,11,(1,2),*1,115)
  F12 12,(1,2),(1,115)       F13 13,(1,2),(1,115)
  F14 14,(1,2),(1,115)       F15 15,(1,2),(1,115)
  F16 16,(1,2),(1,115)       F17 17,(1,3),(1,115)
  F18 18,(1,5),(1,115)       F19 19,(1,5),(1,115)
  CASGN 115,7,(1,1),0,(4,2),(1,116)
  CTEST 116,9,(4,1),(1,117),118,117,118
  CASGN 118,7,(1,2),0,(4,3),(1,119)
  CASGN 119,7,(1,3),0,(4,4),(4,1)
  COUNTR 117,8,(1,25),-1,(1,1),(8,1)
  Q20,-20,(1,20)      Q21 -21,(1,21)      Q22 -22,(1,22)
  Q23 -23,(1,23)
  F20 20,(1,4),(1,120)       F21 21,(1,4),(1,120)
```

Figure 3.6 Example 1 cont'd

```
    F22 22,(1,4),(1,120)                    F23 23,(1,4),(1,120)
CRMVE 120,11
 CGEN 200, 1,(1,10),0,0,10,(1,201)
 CPRT 201,13,(1,202)
 CTRM 202,12,(1,5)
    FN1 1,2,4,2,11,(0,0/.083,-10/.166,-11/.249,-12/.332,
     -13/.499,-14/.582,-15/ x .665,-16/.832,-17/.915,
     -18/1.0,-19)
FN2 2,2,5,2,12 (0,0/.072,-10/.143,-11/.215,-12/.286,
  -13/.429,-14/.501,-15/ x .572,-16/.715,-17/.787,-18/.858,
  -19/1.0,117)
DR 117,8,1 (-20,-21,-22,-23)
```

Stage 5

To prepare precise specifications for the final model data was collected as described in stage 3. The unit of time for the simulation was chosen to be ten seconds. Thus all other times are expressed as multiples of ten second units. Figure 3.4 specifies the generation and facility times. The remaining requirements before the simulation may begin are the expression of the model in the appropriate computer language and the preparation of a plan for the use of the model.

Recalling the instructions for using this simulation system a model diagram is required. Figure 3.5 is such a diagram. Each block has been numbered according to the numbering rules. The corresponding model definition statements are encoded from this diagram appear in figure 3.6.

The plan for using this model would take into consideration the periods of the day which are to be simulated, changing staffing patterns at the checkout isles and behind the food counters, and other factors pertinent to the purpose of the study.

Figure 3.7 is a detailed explanation of the statements in figure 3.6.

The last three statements in figure 3.6 are added for the purpose of, firstly printing the counter values every ten time units thus displaying the contents of counter 25 the number of transactions in the system and, secondly to control the duration of the simulation to five hundred seconds.

Figure 3.8 contains the output from the simulation.

Figure 3.7

Model Definition Statement Explanations

Generate control block 100

```
CGEN    100,1,( 1,2 ), 0,0,1000, ( 1,101 )
```

- block number
- block type
- inter-arrival time
- inter-arrival modifier
- generation offset
- number of transactions to be generated
- next entity specification

Counter block 101

```
COUNT    101,8,( 1,25 ), 1,( 1,1 ), ( 1,102 )
```

- block number
- block type
- counter number specification
- modification
- modification value specification
- next entity specification

figure 3.7 con't

Assign Blocks 102 to 105

```
CASGN        102,7, (1,1 ),0, (2,1 ), (1,103 )
CASGN        102,7, (1,2 ),0, (2,2 ), (1,104 )
CASGN        104,7, (1,3 ),0, (2,2 ), (1,105 )
CASGN        105,7, (1,4 ),0, (1,117), (4,1 )
```

- block number
- block type
- parameter number
- specification
- modification
- modification value
- specification
- nexy entity
- specification

Queues -10 to -19

```
Q10     -10, (1,10 )
Q11     -11, (1,11 )
Q12     -12, (1,12 )
Q13     -13, (1,13 )
Q14     -14, (1,14 )
Q15     -15, (1,15 )
Q16     -16, (1,16 )
Q17     -17, (1,17 )
```

figure 3.7 con't

Q18 -18, (1,18)

Q19 -19, (1,19)

- queue number
- next entity
- specification

Facilities 10 to 19

F10 10,(1,2), (1,115)

F11 11,(1,2), (1,115)

F12 12,(1,2), (1,115)

F13 13,(1,2), (1,115)

F14 14,(1,2), (1,115)

F15 15,(1,2), (1,115)

F16 16,(1,2), (1,115)

F17 17,(1,3), (1,115)

F18 18,(1,5), (1,115)

F19 19,(1,5), (1,115)

- facility number
- processing
- time specification
- next entity
- specification

Assign blocks 115,118,119 see above block 102, to 105

Test control block 116

figure 3.7 cont'd

CTEST 116,9, (4,1),(1,117), 118,117,118

- block number
- block type
- "A" argument
- specification
- "B" argument
- specification
- next entity if A< B
- next entity if A= B
- next entity if A> B

Counter block 117 same form as block number 101 (see above)

Queues -20 to -23 same form as queues -10 to -19 (see above)

Facilities 20 to 23 same form as facilities 20 to 23 (see above)

Remove block 120

CRMVE 120,11

- block number
- block type

Generate block 200 same form as block number 100 (see above)

Print block 201

CPRT 201,13, (1,202)

- block number
- block type
- next entity
- specification

figure 3.7 cont'd

Terminate block 202

```
   CTERM      202,12, (1,5 )
 - block number
 - block type
 - termination count
 - specification
```

3.2.3 Interpretation of Results

Figure 3.8 shows the output from the execution of this model. From this output the following relevant observations can be made:

1. The highest utilization of any of the food counters, facilities 10 to 19 is .579. (See page 132)
2. The highest utilized of the checkout counters is facility 23 .(See page 132).
3. The longest queue for the food counters contained two customers.(See page 133).
4. The longest average waiting time for any food counter queue was 7 units or 70 seconds at food counter 18, the hot meals counter. All other average waiting times are 20 seconds or less.
5. The longest checkout queue was 1 while the highest average waiting time was 30 seconds.

From these observations the following conclusions can be made:

1. From observations 1,3 and 4 the only food counter which shows minor signs of congestion is the hot meals counter.
2. From observations 2 and 5 the checkout counters show no signs of congestion.
3. As a general conclusion it would appear that this system operating under the simulated transaction load is under-utilized.

The execution output on pages 123 to 130 indicates the path which each transaction has taken through the model. This output should be checked during the initial stages of the model construction to verify the model configuration and the transaction flow through the model.

```
                         AN ELEMENTARY SIMULATION SYSTEM
THIS SIMULATION SYSTEM REQUIRES THAT THE PHYSICAL SYSTEM TO BE SIMULATED
BE DEFINE BY THE FOLLOWING MODEL DEFINITION STATEMENTS:-
      DECISION RULE (D).....NEXT ENTITY CANDIDATES & SPECIAL DEC. RULES FOR CHOICE
      FACILITY (F)       ...DESCRIPTION OF THE FACILITY
      QUEUE    (Q)       ...DESCRIPTION OF THE QUEUE
      STORAGE  (S)       ...DESCRIPTION OF THE STORAGE
      CONTROL BLOCK  (C) ...DESCRIPTION OF THE CONTROL BLOCK
      FUNCTION (FN)      ...DESCRIPTION OF THE FUNCTION
      VARIABLE EQUATION(V)..DESCRIPTION OF THE EQUATION
      MODEL DEFINITION INPUT SUMMARY AND EDIT
....5...10....5...20....5...30....5...40....5...50....5...60....5...70....5...80
 CGEN 100,1,(1,2),0,0,1000,(1,101)
   COUNTR 101,8,(1,25),1,(1,1),(1,102)
   CASGN 102,7,(1,1),0,(2,1),(1,103)
   CASGN 103,7,(1,2),0,(2,2),(1,104)
   CASGN 104,7,(1,3),0,(2,2),(1,105)
   CASGN 105,7,(1,4),0,(1,117),(4,1)
Q10,-10,1,10   Q11 -11,1,11   Q12 -12,1,12   Q13 -13,1,13
Q14 -14 1 14   Q15 -15,1,15   Q16 -16,1 16   Q17 -17 1 17
Q18  -18 1 18  Q19 -19,(1,19)
   F10,10,(1,2),(1,115)      F11,11,(1,2),(1,115)
   F12 12,(1,2),(1,115)      F13 13,(1,2),(1,115)
   F14 14,(1,2),(1,115)      F15 15,(1,2),(1,115)
   F16 16,(1,2),(1,115)      F17 17,(1,3),(1,115)
   F18 18,(1,5),(1,115)      F19 19,(1,5),(1,115)
   CASGN 115,7,(1,1),0,(4,2),(1,116)
   CTEST 116,9,(4,1),(1,117),119,117,118
   CASGN 118,7,(1,2),0,(4,3),(1,119)
   CASGN 119,7,(1,3),0,(4,4),(4,1)
   COUNTR 117,8,(1,25),-1,(1,1),(8,1)
   Q20,-20,(1,20)   Q21 -21,(1,21)   Q22 -22,(1,22)
   Q23 -23,(1,23)
   F20 20,(1,4),(1,120)      F21 21,(1,4),(1,120)
   F22 22,(1,4),(1,120)      F23 23,(1,4),(1,120)
CRMVE 120,11
 CGEN 200 1,(1,10),0,0,10,(1,201)
 CPRT 201,13,(1,202)
 CTRM 202,12,(1,5)
   FN1 1,2,4,2,11,(0,0/.083,-10/.166,-11/.249,-12/.332,-13/.499,-14/.582,-15/  X
   .665,-16/.832,-17/.915,-18/1.0,-19)
FN2 2,2,5,2,12 (0,0/.072,-10/.143,-11/.215,-12/.286,-13/.429,-14/.501,-15/     X
 .572,-16/.715,-17/.787,-18/.858,-19/1.0,117)
DR 117,8,1 (-20,-21,-22,-23)
```

Figure 3.8 Example 1:output

```
1125 THERE ARE    0 ERRORS IN THE EDIT PHASE.
          SIMULATION IS CONTINUED.
```

```
TRANSACTION #    1 IS LEAVING ENTITY #  100.
TRANSACTION #    1 IS LEAVING ENTITY #  101.
TRANSACTION #    1 IS LEAVING ENTITY #  102.
TRANSACTION #    1 IS LEAVING ENTITY #  103.
TRANSACTION #    1 IS LEAVING ENTITY #  104.
TRANSACTION #    1 IS LEAVING ENTITY #  105.
TRANSACTION #    1 IS LEAVING ENTITY #  -10.
TRANSACTION #    2 IS LEAVING ENTITY #  100.
TRANSACTION #    2 IS LEAVING ENTITY #  101.
TRANSACTION #    2 IS LEAVING ENTITY #  102.
TRANSACTION #    2 IS LEAVING ENTITY #  103.
TRANSACTION #    2 IS LEAVING ENTITY #  104.
TRANSACTION #    2 IS LEAVING ENTITY #  105.
TRANSACTION #    2 IS LEAVING ENTITY #  -15.
TRANSACTION #    1 IS LEAVING ENTITY #   10.
TRANSACTION #    1 IS LEAVING ENTITY #  115.
TRANSACTION #    1 IS LEAVING ENTITY #  116.
TRANSACTION #    1 IS LEAVING ENTITY #  118.
TRANSACTION #    1 IS LEAVING ENTITY #  119.
TRANSACTION #    1 IS LEAVING ENTITY #  -10.
TRANSACTION #    3 IS LEAVING ENTITY #  100.
TRANSACTION #    3 IS LEAVING ENTITY #  101.
TRANSACTION #    3 IS LEAVING ENTITY #  102.
TRANSACTION #    3 IS LEAVING ENTITY #  103.
TRANSACTION #    3 IS LEAVING ENTITY #  104.
TRANSACTION #    3 IS LEAVING ENTITY #  105.
TRANSACTION #    3 IS LEAVING ENTITY #  -18.
TRANSACTION #    2 IS LEAVING ENTITY #   15.
TRANSACTION #    2 IS LEAVING ENTITY #  115.
TRANSACTION #    2 IS LEAVING ENTITY #  116.
TRANSACTION #    2 IS LEAVING ENTITY #  117.
TRANSACTION #    2 IS LEAVING ENTITY #  -23.
TRANSACTION #    1 IS LEAVING ENTITY #   10.
TRANSACTION #    1 IS LEAVING ENTITY #  115.
TRANSACTION #    1 IS LEAVING ENTITY #  116.
TRANSACTION #    1 IS LEAVING ENTITY #  118.
TRANSACTION #    1 IS LEAVING ENTITY #  119.
TRANSACTION #    1 IS LEAVING ENTITY #  -16.
TRANSACTION #    4 IS LEAVING ENTITY #  100.
TRANSACTION #    4 IS LEAVING ENTITY #  101.
TRANSACTION #    4 IS LEAVING ENTITY #  102.
TRANSACTION #    4 IS LEAVING ENTITY #  103.
TRANSACTION #    4 IS LEAVING ENTITY #  104.
TRANSACTION #    4 IS LEAVING ENTITY #  105.
TRANSACTION #    4 IS LEAVING ENTITY #  -19.
TRANSACTION #    1 IS LEAVING ENTITY #   16.
TRANSACTION #    1 IS LEAVING ENTITY #  115.
TRANSACTION #    1 IS LEAVING ENTITY #  116.
TRANSACTION #    1 IS LEAVING ENTITY #  117.
TRANSACTION #    1 IS LEAVING ENTITY #  -23.
TRANSACTION #    5 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=         0; P3=         0; P4=         0; P5=         0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=        0; C 3=        0; C 4=        0; C 5=        0; C 6=        0; C 7=        0; C 8=        0; C 9=        0; C10=        0;
C11=       0; C12=        0; C13=        0; C14=        0; C15=        0; C16=        0; C17=        0; C18=        0; C19=        0; C20=        0;
C21=       0; C22=        0; C23=        0; C24=        0; C25=        2;
CLOCK TIME =    10 UNITS.
TRANSACTION #    5 IS LEAVING ENTITY #  201.
TRANSACTION #    5 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    2 IS LEAVING ENTITY #   23.
TRANSACTION #    2 IS REMOVED BY ENTITY #  120.
TRANSACTION #    6 IS LEAVING ENTITY #  100.
TRANSACTION #    6 IS LEAVING ENTITY #  101.
```

Figure 3.8

```
TRANSACTION #    6 IS LEAVING ENTITY #  102.
TRANSACTION #    6 IS LEAVING ENTITY #  103.
TRANSACTION #    6 IS LEAVING ENTITY #  104.
TRANSACTION #    6 IS LEAVING ENTITY #  105.
TRANSACTION #    6 IS LEAVING ENTITY #  -17.
TRANSACTION #    3 IS LEAVING ENTITY #   18.
TRANSACTION #    3 IS LEAVING ENTITY #  115.
TRANSACTION #    3 IS LEAVING ENTITY #  116.
TRANSACTION #    3 IS LEAVING ENTITY #  118.
TRANSACTION #    3 IS LEAVING ENTITY #  119.
TRANSACTION #    3 IS LEAVING ENTITY #  -18.
TRANSACTION #    7 IS LEAVING ENTITY #  100.
TRANSACTION #    7 IS LEAVING ENTITY #  101.
TRANSACTION #    7 IS LEAVING ENTITY #  102.
TRANSACTION #    7 IS LEAVING ENTITY #  103.
TRANSACTION #    7 IS LEAVING ENTITY #  104.
TRANSACTION #    7 IS LEAVING ENTITY #  105.
TRANSACTION #    7 IS LEAVING ENTITY #  -12.
TRANSACTION #    4 IS LEAVING ENTITY #   19.
TRANSACTION #    4 IS LEAVING ENTITY #  115.
TRANSACTION #    4 IS LEAVING ENTITY #  116.
TRANSACTION #    4 IS LEAVING ENTITY #  118.
TRANSACTION #    4 IS LEAVING ENTITY #  119.
TRANSACTION #    4 IS LEAVING ENTITY #  -19.
TRANSACTION #    6 IS LEAVING ENTITY #   17.
TRANSACTION #    6 IS LEAVING ENTITY #  115.
TRANSACTION #    6 IS LEAVING ENTITY #  116.
TRANSACTION #    6 IS LEAVING ENTITY #  118.
TRANSACTION #    6 IS LEAVING ENTITY #  119.
TRANSACTION #    6 IS LEAVING ENTITY #  -10.
TRANSACTION #    1 IS LEAVING ENTITY #   23.
TRANSACTION #    1 IS REMOVED BY ENTITY #  120.
TRANSACTION #    8 IS LEAVING ENTITY #  100.
TRANSACTION #    8 IS LEAVING ENTITY #  101.
TRANSACTION #    8 IS LEAVING ENTITY #  102.
TRANSACTION #    8 IS LEAVING ENTITY #  103.
TRANSACTION #    8 IS LEAVING ENTITY #  104.
TRANSACTION #    8 IS LEAVING ENTITY #  105.
TRANSACTION #    8 IS LEAVING ENTITY #  -17.
TRANSACTION #    7 IS LEAVING ENTITY #   12.
TRANSACTION #    7 IS LEAVING ENTITY #  115.
TRANSACTION #    7 IS LEAVING ENTITY #  116.
TRANSACTION #    7 IS LEAVING ENTITY #  118.
TRANSACTION #    7 IS LEAVING ENTITY #  119.
TRANSACTION #    7 IS LEAVING ENTITY #  -16.
TRANSACTION #    6 IS LEAVING ENTITY #   10.
TRANSACTION #    6 IS LEAVING ENTITY #  115.
TRANSACTION #    6 IS LEAVING ENTITY #  116.
TRANSACTION #    6 IS LEAVING ENTITY #  118.
TRANSACTION #    6 IS LEAVING ENTITY #  119.
TRANSACTION #    6 IS LEAVING ENTITY #  -17.
TRANSACTION #    3 IS LEAVING ENTITY #   18.
TRANSACTION #    3 IS LEAVING ENTITY #  115.
TRANSACTION #    3 IS LEAVING ENTITY #  116.
TRANSACTION #    3 IS LEAVING ENTITY #  118.
TRANSACTION #    3 IS LEAVING ENTITY #  119.
TRANSACTION #    3 IS LEAVING ENTITY #  -12.
TRANSACTION #    9 IS LEAVING ENTITY #  100.
TRANSACTION #    9 IS LEAVING ENTITY #  101.
TRANSACTION #    9 IS LEAVING ENTITY #  102.
TRANSACTION #    9 IS LEAVING ENTITY #  103.
TRANSACTION #    9 IS LEAVING ENTITY #  104.
TRANSACTION #    9 IS LEAVING ENTITY #  105.
TRANSACTION #    9 IS LEAVING ENTITY #  -10.
TRANSACTION #    7 IS LEAVING ENTITY #   16.
TRANSACTION #    7 IS LEAVING ENTITY #  115.
```

Figure 3.8

```
TRANSACTION #     7 IS LEAVING ENTITY #  116.
TRANSACTION #     7 IS LEAVING ENTITY #  118.
TRANSACTION #     7 IS LEAVING ENTITY #  119.
TRANSACTION #     7 IS LEAVING ENTITY #  -19.
TRANSACTION #     8 IS LEAVING ENTITY #   17.
TRANSACTION #     8 IS LEAVING ENTITY #  115.
TRANSACTION #     8 IS LEAVING ENTITY #  116.
TRANSACTION #     8 IS LEAVING ENTITY #  118.
TRANSACTION #     8 IS LEAVING ENTITY #  119.
TRANSACTION #     8 IS LEAVING ENTITY #  -12.
TRANSACTION #     4 IS LEAVING ENTITY #   19.
TRANSACTION #     4 IS LEAVING ENTITY #  115.
TRANSACTION #     4 IS LEAVING ENTITY #  116.
TRANSACTION #     4 IS LEAVING ENTITY #  118.
TRANSACTION #     4 IS LEAVING ENTITY #  119.
TRANSACTION #     4 IS LEAVING ENTITY #  -10.
TRANSACTION #     3 IS LEAVING ENTITY #   12.
TRANSACTION #     3 IS LEAVING ENTITY #  115.
TRANSACTION #     3 IS LEAVING ENTITY #  116.
TRANSACTION #     3 IS LEAVING ENTITY #  117.
TRANSACTION #     3 IS LEAVING ENTITY #  -23.
TRANSACTION #    10 IS LEAVING ENTITY #  100.
TRANSACTION #    10 IS LEAVING ENTITY #  101.
TRANSACTION #    10 IS LEAVING ENTITY #  102.
TRANSACTION #    10 IS LEAVING ENTITY #  103.
TRANSACTION #    10 IS LEAVING ENTITY #  104.
TRANSACTION #    10 IS LEAVING ENTITY #  105.
TRANSACTION #    10 IS LEAVING ENTITY #  -10.
TRANSACTION #     9 IS LEAVING ENTITY #   10.
TRANSACTION #     9 IS LEAVING ENTITY #  115.
TRANSACTION #     9 IS LEAVING ENTITY #  116.
TRANSACTION #     9 IS LEAVING ENTITY #  118.
TRANSACTION #     9 IS LEAVING ENTITY #  119.
TRANSACTION #     9 IS LEAVING ENTITY #  -12.
TRANSACTION #    11 IS LEAVING ENTITY #  200.
PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       6;
CLOCK TIME =   20 UNITS.
TRANSACTION #    11 IS LEAVING ENTITY #  201.
TRANSACTION #    11 IS TERMINATED AT ENTITY #  202.
TRANSACTION #     6 IS LEAVING ENTITY #   17.
TRANSACTION #     6 IS LEAVING ENTITY #  115.
TRANSACTION #     6 IS LEAVING ENTITY #  116.
TRANSACTION #     6 IS LEAVING ENTITY #  117.
TRANSACTION #     6 IS LEAVING ENTITY #  -23.
TRANSACTION #     8 IS LEAVING ENTITY #   12.
TRANSACTION #     8 IS LEAVING ENTITY #  115.
TRANSACTION #     8 IS LEAVING ENTITY #  116.
TRANSACTION #     8 IS LEAVING ENTITY #  118.
TRANSACTION #     8 IS LEAVING ENTITY #  119.
TRANSACTION #     8 IS LEAVING ENTITY #  -14.
TRANSACTION #    12 IS LEAVING ENTITY #  100.
TRANSACTION #    12 IS LEAVING ENTITY #  101.
TRANSACTION #    12 IS LEAVING ENTITY #  102.
TRANSACTION #    12 IS LEAVING ENTITY #  103.
TRANSACTION #    12 IS LEAVING ENTITY #  104.
TRANSACTION #    12 IS LEAVING ENTITY #  105.
TRANSACTION #    12 IS LEAVING ENTITY #  -16.
TRANSACTION #     4 IS LEAVING ENTITY #   10.
TRANSACTION #     4 IS LEAVING ENTITY #  115.
TRANSACTION #     4 IS LEAVING ENTITY #  116.
```

Figure 3.8

```
TRANSACTION #    4 IS LEAVING ENTITY #  117.
TRANSACTION #    4 IS LEAVING ENTITY #  -22.
TRANSACTION #    3 IS LEAVING ENTITY #   23.
TRANSACTION #    3 IS REMOVED BY ENTITY #  120.
TRANSACTION #    8 IS LEAVING ENTITY #   14.
TRANSACTION #    8 IS LEAVING ENTITY #  115.
TRANSACTION #    8 IS LEAVING ENTITY #  116.
TRANSACTION #    8 IS LEAVING ENTITY #  117.
TRANSACTION #    8 IS LEAVING ENTITY #  -23.
TRANSACTION #    9 IS LEAVING ENTITY #   12.
TRANSACTION #    9 IS LEAVING ENTITY #  115.
TRANSACTION #    9 IS LEAVING ENTITY #  116.
TRANSACTION #    9 IS LEAVING ENTITY #  118.
TRANSACTION #    9 IS LEAVING ENTITY #  119.
TRANSACTION #    9 IS LEAVING ENTITY #  -14.
TRANSACTION #   13 IS LEAVING ENTITY #  100.
TRANSACTION #   13 IS LEAVING ENTITY #  101.
TRANSACTION #   13 IS LEAVING ENTITY #  102.
TRANSACTION #   13 IS LEAVING ENTITY #  103.
TRANSACTION #   13 IS LEAVING ENTITY #  104.
TRANSACTION #   13 IS LEAVING ENTITY #  105.
TRANSACTION #   13 IS LEAVING ENTITY #  -15.
TRANSACTION #   12 IS LEAVING ENTITY #   16.
TRANSACTION #   12 IS LEAVING ENTITY #  115.
TRANSACTION #   12 IS LEAVING ENTITY #  116.
TRANSACTION #   12 IS LEAVING ENTITY #  117.
TRANSACTION #   12 IS LEAVING ENTITY #  -22.
TRANSACTION #   10 IS LEAVING ENTITY #   10.
TRANSACTION #   10 IS LEAVING ENTITY #  115.
TRANSACTION #   10 IS LEAVING ENTITY #  116.
TRANSACTION #   10 IS LEAVING ENTITY #  117.
TRANSACTION #   10 IS LEAVING ENTITY #  -21.
TRANSACTION #    7 IS LEAVING ENTITY #   19.
TRANSACTION #    7 IS LEAVING ENTITY #  115.
TRANSACTION #    7 IS LEAVING ENTITY #  116.
TRANSACTION #    7 IS LEAVING ENTITY #  117.
TRANSACTION #    7 IS LEAVING ENTITY #  -21.
TRANSACTION #    4 IS LEAVING ENTITY #   22.
TRANSACTION #    4 IS REMOVED BY ENTITY #  120.
TRANSACTION #    9 IS LEAVING ENTITY #   14.
TRANSACTION #    9 IS LEAVING ENTITY #  115.
TRANSACTION #    9 IS LEAVING ENTITY #  116.
TRANSACTION #    9 IS LEAVING ENTITY #  117.
TRANSACTION #    9 IS LEAVING ENTITY #  -22.
TRANSACTION #   14 IS LEAVING ENTITY #  100.
TRANSACTION #   14 IS LEAVING ENTITY #  101.
TRANSACTION #   14 IS LEAVING ENTITY #  102.
TRANSACTION #   14 IS LEAVING ENTITY #  103.
TRANSACTION #   14 IS LEAVING ENTITY #  104.
TRANSACTION #   14 IS LEAVING ENTITY #  105.
TRANSACTION #   14 IS LEAVING ENTITY #  -17.
TRANSACTION #   13 IS LEAVING ENTITY #   15.
TRANSACTION #   13 IS LEAVING ENTITY #  115.
TRANSACTION #   13 IS LEAVING ENTITY #  116.
TRANSACTION #   13 IS LEAVING ENTITY #  117.
TRANSACTION #   13 IS LEAVING ENTITY #  -20.
TRANSACTION #    6 IS LEAVING ENTITY #   23.
TRANSACTION #    6 IS REMOVED BY ENTITY #  120.
TRANSACTION #   10 IS LEAVING ENTITY #   21.
TRANSACTION #   10 IS REMOVED BY ENTITY #  120.
TRANSACTION #   15 IS LEAVING ENTITY #  100.
TRANSACTION #   15 IS LEAVING ENTITY #  101.
TRANSACTION #   15 IS LEAVING ENTITY #  102.
TRANSACTION #   15 IS LEAVING ENTITY #  103.
TRANSACTION #   15 IS LEAVING ENTITY #  104.
TRANSACTION #   15 IS LEAVING ENTITY #  105.
```

Figure 3.8

```
TRANSACTION #   15 IS LEAVING ENTITY #  -12.
TRANSACTION #   14 IS LEAVING ENTITY #   17.
TRANSACTION #   14 IS LEAVING ENTITY #  115.
TRANSACTION #   14 IS LEAVING ENTITY #  116.
TRANSACTION #   14 IS LEAVING ENTITY #  118.
TRANSACTION #   14 IS LEAVING ENTITY #  119.
TRANSACTION #   14 IS LEAVING ENTITY #  -15.
TRANSACTION #   12 IS LEAVING ENTITY #   22.
TRANSACTION #   12 IS REMOVED BY ENTITY #  120.
TRANSACTION #   13 IS LEAVING ENTITY #   20.
TRANSACTION #   13 IS REMOVED BY ENTITY #  120.
TRANSACTION #   16 IS LEAVING ENTITY #  100.
TRANSACTION #   16 IS LEAVING ENTITY #  101.
TRANSACTION #   16 IS LEAVING ENTITY #  102.
TRANSACTION #   16 IS LEAVING ENTITY #  103.
TRANSACTION #   16 IS LEAVING ENTITY #  104.
TRANSACTION #   16 IS LEAVING ENTITY #  105.
TRANSACTION #   16 IS LEAVING ENTITY #  -13.
TRANSACTION #   15 IS LEAVING ENTITY #   12.
TRANSACTION #   15 IS LEAVING ENTITY #  115.
TRANSACTION #   15 IS LEAVING ENTITY #  116.
TRANSACTION #   15 IS LEAVING ENTITY #  118.
TRANSACTION #   15 IS LEAVING ENTITY #  119.
TRANSACTION #   15 IS LEAVING ENTITY #  -10.
TRANSACTION #   14 IS LEAVING ENTITY #   15.
TRANSACTION #   14 IS LEAVING ENTITY #  115.
TRANSACTION #   14 IS LEAVING ENTITY #  116.
TRANSACTION #   14 IS LEAVING ENTITY #  118.
TRANSACTION #   14 IS LEAVING ENTITY #  119.
TRANSACTION #   14 IS LEAVING ENTITY #  -17.
TRANSACTION #   17 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       3;
CLOCK TIME =    30 UNITS.
TRANSACTION #   17 IS LEAVING ENTITY #  201.
TRANSACTION #   17 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    8 IS LEAVING ENTITY #   23.
TRANSACTION #    8 IS REMOVED BY ENTITY #  120.
TRANSACTION #    7 IS LEAVING ENTITY #   21.
TRANSACTION #    7 IS REMOVED BY ENTITY #  120.
TRANSACTION #   18 IS LEAVING ENTITY #  100.
TRANSACTION #   18 IS LEAVING ENTITY #  101.
TRANSACTION #   18 IS LEAVING ENTITY #  102.
TRANSACTION #   18 IS LEAVING ENTITY #  103.
TRANSACTION #   18 IS LEAVING ENTITY #  104.
TRANSACTION #   18 IS LEAVING ENTITY #  105.
TRANSACTION #   18 IS LEAVING ENTITY #  -12.
TRANSACTION #   16 IS LEAVING ENTITY #   13.
TRANSACTION #   16 IS LEAVING ENTITY #  115.
TRANSACTION #   16 IS LEAVING ENTITY #  116.
TRANSACTION #   16 IS LEAVING ENTITY #  118.
TRANSACTION #   16 IS LEAVING ENTITY #  119.
TRANSACTION #   16 IS LEAVING ENTITY #  -17.
TRANSACTION #   15 IS LEAVING ENTITY #   10.
TRANSACTION #   15 IS LEAVING ENTITY #  115.
TRANSACTION #   15 IS LEAVING ENTITY #  116.
TRANSACTION #   15 IS LEAVING ENTITY #  118.
TRANSACTION #   15 IS LEAVING ENTITY #  119.
TRANSACTION #   15 IS LEAVING ENTITY #  -14.
TRANSACTION #    9 IS LEAVING ENTITY #   22.
TRANSACTION #    9 IS REMOVED BY ENTITY #  120.
```

Figure 3.8

```
TRANSACTION #   14 IS LEAVING ENTITY #   17.
TRANSACTION #   14 IS LEAVING ENTITY #  115.
TRANSACTION #   14 IS LEAVING ENTITY #  116.
TRANSACTION #   14 IS LEAVING ENTITY #  117.
TRANSACTION #   14 IS LEAVING ENTITY #  -23.
TRANSACTION #   19 IS LEAVING ENTITY #  100.
TRANSACTION #   19 IS LEAVING ENTITY #  101.
TRANSACTION #   19 IS LEAVING ENTITY #  102.
TRANSACTION #   19 IS LEAVING ENTITY #  103.
TRANSACTION #   19 IS LEAVING ENTITY #  104.
TRANSACTION #   19 IS LEAVING ENTITY #  105.
TRANSACTION #   19 IS LEAVING ENTITY #  -14.
TRANSACTION #   18 IS LEAVING ENTITY #   12.
TRANSACTION #   18 IS LEAVING ENTITY #  115.
TRANSACTION #   18 IS LEAVING ENTITY #  116.
TRANSACTION #   18 IS LEAVING ENTITY #  118.
TRANSACTION #   18 IS LEAVING ENTITY #  119.
TRANSACTION #   18 IS LEAVING ENTITY #  -14.
TRANSACTION #   15 IS LEAVING ENTITY #   14.
TRANSACTION #   15 IS LEAVING ENTITY #  115.
TRANSACTION #   15 IS LEAVING ENTITY #  116.
TRANSACTION #   15 IS LEAVING ENTITY #  117.
TRANSACTION #   15 IS LEAVING ENTITY #  -23.
TRANSACTION #   20 IS LEAVING ENTITY #  100.
TRANSACTION #   20 IS LEAVING ENTITY #  101.
TRANSACTION #   20 IS LEAVING ENTITY #  102.
TRANSACTION #   20 IS LEAVING ENTITY #  103.
TRANSACTION #   20 IS LEAVING ENTITY #  104.
TRANSACTION #   20 IS LEAVING ENTITY #  105.
TRANSACTION #   20 IS LEAVING ENTITY #  -19.
TRANSACTION #   19 IS LEAVING ENTITY #   14.
TRANSACTION #   19 IS LEAVING ENTITY #  115.
TRANSACTION #   19 IS LEAVING ENTITY #  116.
TRANSACTION #   19 IS LEAVING ENTITY #  118.
TRANSACTION #   19 IS LEAVING ENTITY #  119.
TRANSACTION #   19 IS LEAVING ENTITY #  -12.
TRANSACTION #   16 IS LEAVING ENTITY #   17.
TRANSACTION #   16 IS LEAVING ENTITY #  115.
TRANSACTION #   16 IS LEAVING ENTITY #  116.
TRANSACTION #   16 IS LEAVING ENTITY #  118.
TRANSACTION #   16 IS LEAVING ENTITY #  119.
TRANSACTION #   16 IS LEAVING ENTITY #  -14.
TRANSACTION #   14 IS LEAVING ENTITY #   23.
TRANSACTION #   14 IS REMOVED BY ENTITY #  120.
TRANSACTION #   21 IS LEAVING ENTITY #  100.
TRANSACTION #   21 IS LEAVING ENTITY #  101.
TRANSACTION #   21 IS LEAVING ENTITY #  102.
TRANSACTION #   21 IS LEAVING ENTITY #  103.
TRANSACTION #   21 IS LEAVING ENTITY #  104.
TRANSACTION #   21 IS LEAVING ENTITY #  105.
TRANSACTION #   21 IS LEAVING ENTITY #  -18.
TRANSACTION #   19 IS LEAVING ENTITY #   12.
TRANSACTION #   19 IS LEAVING ENTITY #  115.
TRANSACTION #   19 IS LEAVING ENTITY #  116.
TRANSACTION #   19 IS LEAVING ENTITY #  118.
TRANSACTION #   19 IS LEAVING ENTITY #  119.
TRANSACTION #   19 IS LEAVING ENTITY #  -17.
TRANSACTION #   18 IS LEAVING ENTITY #   14.
TRANSACTION #   18 IS LEAVING ENTITY #  115.
TRANSACTION #   18 IS LEAVING ENTITY #  116.
TRANSACTION #   18 IS LEAVING ENTITY #  118.
TRANSACTION #   18 IS LEAVING ENTITY #  119.
TRANSACTION #   18 IS LEAVING ENTITY #  -18.
TRANSACTION #   22 IS LEAVING ENTITY #  100.
TRANSACTION #   22 IS LEAVING ENTITY #  101.
TRANSACTION #   22 IS LEAVING ENTITY #  102.
```

Figure 3.8

```
TRANSACTION #   22 IS LEAVING ENTITY #  103.
TRANSACTION #   22 IS LEAVING ENTITY #  104.
TRANSACTION #   22 IS LEAVING ENTITY #  105.
TRANSACTION #   22 IS LEAVING ENTITY #  -18.
TRANSACTION #   16 IS LEAVING ENTITY #   14.
TRANSACTION #   16 IS LEAVING ENTITY #  115.
TRANSACTION #   16 IS LEAVING ENTITY #  116.
TRANSACTION #   16 IS LEAVING ENTITY #  117.
TRANSACTION #   16 IS LEAVING ENTITY #  -23.
TRANSACTION #   20 IS LEAVING ENTITY #   19.
TRANSACTION #   20 IS LEAVING ENTITY #  115.
TRANSACTION #   20 IS LEAVING ENTITY #  116.
TRANSACTION #   20 IS LEAVING ENTITY #  118.
TRANSACTION #   20 IS LEAVING ENTITY #  119.
TRANSACTION #   20 IS LEAVING ENTITY #  -17.
TRANSACTION #   19 IS LEAVING ENTITY #   17.
TRANSACTION #   19 IS LEAVING ENTITY #  115.
TRANSACTION #   19 IS LEAVING ENTITY #  116.
TRANSACTION #   19 IS LEAVING ENTITY #  117.
TRANSACTION #   19 IS LEAVING ENTITY #  -22.
TRANSACTION #   23 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       4;
CLOCK TIME =    40 UNITS.
TRANSACTION #   23 IS LEAVING ENTITY #  201.
TRANSACTION #   23 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   15 IS LEAVING ENTITY #   23.
TRANSACTION #   15 IS REMOVED BY ENTITY #  120.
TRANSACTION #   24 IS LEAVING ENTITY #  100.
TRANSACTION #   24 IS LEAVING ENTITY #  101.
TRANSACTION #   24 IS LEAVING ENTITY #  102.
TRANSACTION #   24 IS LEAVING ENTITY #  103.
TRANSACTION #   24 IS LEAVING ENTITY #  104.
TRANSACTION #   24 IS LEAVING ENTITY #  105.
TRANSACTION #   24 IS LEAVING ENTITY #  -17.
TRANSACTION #   21 IS LEAVING ENTITY #   18.
TRANSACTION #   21 IS LEAVING ENTITY #  115.
TRANSACTION #   21 IS LEAVING ENTITY #  116.
TRANSACTION #   21 IS LEAVING ENTITY #  118.
TRANSACTION #   21 IS LEAVING ENTITY #  119.
TRANSACTION #   21 IS LEAVING ENTITY #  -10.
TRANSACTION #   20 IS LEAVING ENTITY #   17.
TRANSACTION #   20 IS LEAVING ENTITY #  115.
TRANSACTION #   20 IS LEAVING ENTITY #  116.
TRANSACTION #   20 IS LEAVING ENTITY #  117.
TRANSACTION #   20 IS LEAVING ENTITY #  -23.
TRANSACTION #   25 IS LEAVING ENTITY #  100.
TRANSACTION #   25 IS LEAVING ENTITY #  101.
TRANSACTION #   25 IS LEAVING ENTITY #  102.
TRANSACTION #   25 IS LEAVING ENTITY #  103.
TRANSACTION #   25 IS LEAVING ENTITY #  104.
TRANSACTION #   25 IS LEAVING ENTITY #  105.
TRANSACTION #   25 IS LEAVING ENTITY #  -18.
TRANSACTION #   19 IS LEAVING ENTITY #   22.
TRANSACTION #   19 IS REMOVED BY ENTITY #  120.
TRANSACTION #   21 IS LEAVING ENTITY #   10.
TRANSACTION #   21 IS LEAVING ENTITY #  115.
TRANSACTION #   21 IS LEAVING ENTITY #  116.
TRANSACTION #   21 IS LEAVING ENTITY #  118.
TRANSACTION #   21 IS LEAVING ENTITY #  119.
TRANSACTION #   21 IS LEAVING ENTITY #  -19.
```

Figure 3.8

```
TRANSACTION #   16 IS LEAVING ENTITY #   23.
TRANSACTION #   16 IS REMOVED BY ENTITY #  120.
TRANSACTION #   26 IS LEAVING ENTITY #  100.
TRANSACTION #   26 IS LEAVING ENTITY #  101.
TRANSACTION #   26 IS LEAVING ENTITY #  102.
TRANSACTION #   26 IS LEAVING ENTITY #  103.
TRANSACTION #   26 IS LEAVING ENTITY #  104.
TRANSACTION #   26 IS LEAVING ENTITY #  105.
TRANSACTION #   26 IS LEAVING ENTITY #  -11.
TRANSACTION #   24 IS LEAVING ENTITY #   17.
TRANSACTION #   24 IS LEAVING ENTITY #  115.
TRANSACTION #   24 IS LEAVING ENTITY #  116.
TRANSACTION #   24 IS LEAVING ENTITY #  118.
TRANSACTION #   24 IS LEAVING ENTITY #  119.
TRANSACTION #   24 IS LEAVING ENTITY #  -14.
TRANSACTION #   18 IS LEAVING ENTITY #   18.
TRANSACTION #   18 IS LEAVING ENTITY #  115.
TRANSACTION #   18 IS LEAVING ENTITY #  116.
TRANSACTION #   18 IS LEAVING ENTITY #  117.
TRANSACTION #   18 IS LEAVING ENTITY #  -23.
TRANSACTION #   27 IS LEAVING ENTITY #  100.
TRANSACTION #   27 IS LEAVING ENTITY #  101.
TRANSACTION #   27 IS LEAVING ENTITY #  102.
TRANSACTION #   27 IS LEAVING ENTITY #  103.
TRANSACTION #   27 IS LEAVING ENTITY #  104.
TRANSACTION #   27 IS LEAVING ENTITY #  105.
TRANSACTION #   27 IS LEAVING ENTITY #  -15.
TRANSACTION #   26 IS LEAVING ENTITY #   11.
TRANSACTION #   26 IS LEAVING ENTITY #  115.
TRANSACTION #   26 IS LEAVING ENTITY #  116.
TRANSACTION #   26 IS LEAVING ENTITY #  118.
TRANSACTION #   26 IS LEAVING ENTITY #  119.
TRANSACTION #   26 IS LEAVING ENTITY #  -13.
TRANSACTION #   24 IS LEAVING ENTITY #   14.
TRANSACTION #   24 IS LEAVING ENTITY #  115.
TRANSACTION #   24 IS LEAVING ENTITY #  116.
TRANSACTION #   24 IS LEAVING ENTITY #  118.
TRANSACTION #   24 IS LEAVING ENTITY #  119.
TRANSACTION #   24 IS LEAVING ENTITY #  -19.
TRANSACTION #   21 IS LEAVING ENTITY #   19.
TRANSACTION #   21 IS LEAVING ENTITY #  115.
TRANSACTION #   21 IS LEAVING ENTITY #  116.
TRANSACTION #   21 IS LEAVING ENTITY #  117.
TRANSACTION #   21 IS LEAVING ENTITY #  -22.
TRANSACTION #   20 IS LEAVING ENTITY #   23.
TRANSACTION #   20 IS REMOVED BY ENTITY #  120.
TRANSACTION #   28 IS LEAVING ENTITY #  100.
TRANSACTION #   28 IS LEAVING ENTITY #  101.
TRANSACTION #   28 IS LEAVING ENTITY #  102.
TRANSACTION #   28 IS LEAVING ENTITY #  103.
TRANSACTION #   28 IS LEAVING ENTITY #  104.
TRANSACTION #   28 IS LEAVING ENTITY #  105.
TRANSACTION #   28 IS LEAVING ENTITY #  -16.
TRANSACTION #   27 IS LEAVING ENTITY #   15.
TRANSACTION #   27 IS LEAVING ENTITY #  115.
TRANSACTION #   27 IS LEAVING ENTITY #  116.
TRANSACTION #   27 IS LEAVING ENTITY #  118.
TRANSACTION #   27 IS LEAVING ENTITY #  119.
TRANSACTION #   27 IS LEAVING ENTITY #  -15.
TRANSACTION #   26 IS LEAVING ENTITY #   13.
TRANSACTION #   26 IS LEAVING ENTITY #  115.
TRANSACTION #   26 IS LEAVING ENTITY #  116.
TRANSACTION #   26 IS LEAVING ENTITY #  118.
TRANSACTION #   26 IS LEAVING ENTITY #  119.
TRANSACTION #   26 IS LEAVING ENTITY #  -17.
TRANSACTION #   29 IS LEAVING ENTITY #  200.
```

Figure 3.8

```
PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       6;
CLOCK TIME =   50 UNITS.
TRANSACTION #   29 IS LEAVING ENTITY #  201.
TRANSACTION #   29 IS TERMINATED AT ENTITY #  202.
```

Figure 3.8

*** FACILITY STATISTICS ***

FACILITY NO.	TRANSACTION ENTRIES	AVERAGE UTILIZATION	AVERAGE TIME/TRANSACTION
10	8	0.319	2
11	1	0.039	2
12	7	0.279	2
13	2	0.079	2
14	7	0.279	2
15	5	0.199	2
16	4	0.159	2
17	10	0.579	2
18	5	0.479	4
19	6	0.539	4
20	1	0.079	4
21	2	0.159	4
22	5	0.359	3
23	10	0.759	3

Figure 3.8

*** QUEUE STATISTICS ***

QUEUE NO.	TRANSACTION ENTRIES	AVERAGE CONTENTS	MAXIMUM CONTENTS	NUMBER OF ZERO ENTRIES	AVERAGE TIME/ TRANSACTION	AVE.TIME/TRANS (NO ZERO ENTRIES)
-10	8	0	2	7	0	2
-11	1	0	1	1	0	0
-12	7	0	1	5	0	1
-13	2	0	1	2	0	0
-14	7	0	2	5	0	1
-15	5	0	1	5	0	0
-16	4	0	1	4	0	0
-17	10	0	1	7	0	2
-18	6	0	2	3	3	7
-19	6	0	1	4	0	1
-20	1	0	1	1	0	0
-21	2	0	1	1	1	3
-22	5	0	1	3	1	3
-23	10	0	1	3	1	2

Figure 3.8

```
PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       6;
CLCCK TIME =   50 UNITS.

THE SIMULATION HAS BEEN TERMINATED BY     5 TERMINATIONS FOR CONTROL BLOCK # 202.
```

Figure 3.8

3.3 Example 2

3.3.1 Statement of the Problem

This example, like the first, illustrates the simulation of a cafeteria. The physical configuration of this cafeteria is different from the first. Figure 3.9 illustrates the positions of the various food counters, the checkout counter and the queues leading to these facilities. The purpose of this simulation is to study the general performance of the system and compare the performance with example 1. Transactions enter this system with the same interarrival time as those of example 1. The facility processing times are also the same. This comparison is left to the reader for analysis.

3.3.2 Model Building Procedure

Stage 1

The problem was stated generally in section 3.2.1.

The system boundaries may be defined in the following manner: The system begins where customers enter the first waiting line, it ends where customers leave the check-out counter.

The outputs required for comparison of the two methods of cafeteria organization are counter utilizations, queue lengths, and the total number of customers in the system.

Figure 3.9 Example 2: Cafeteria

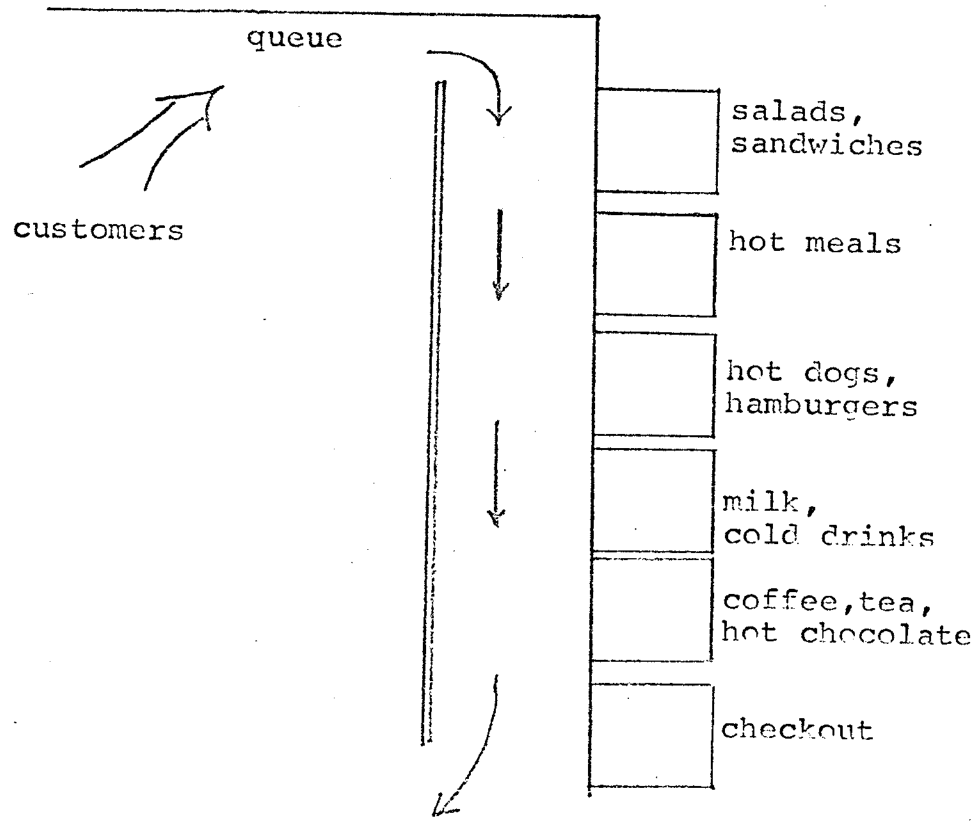
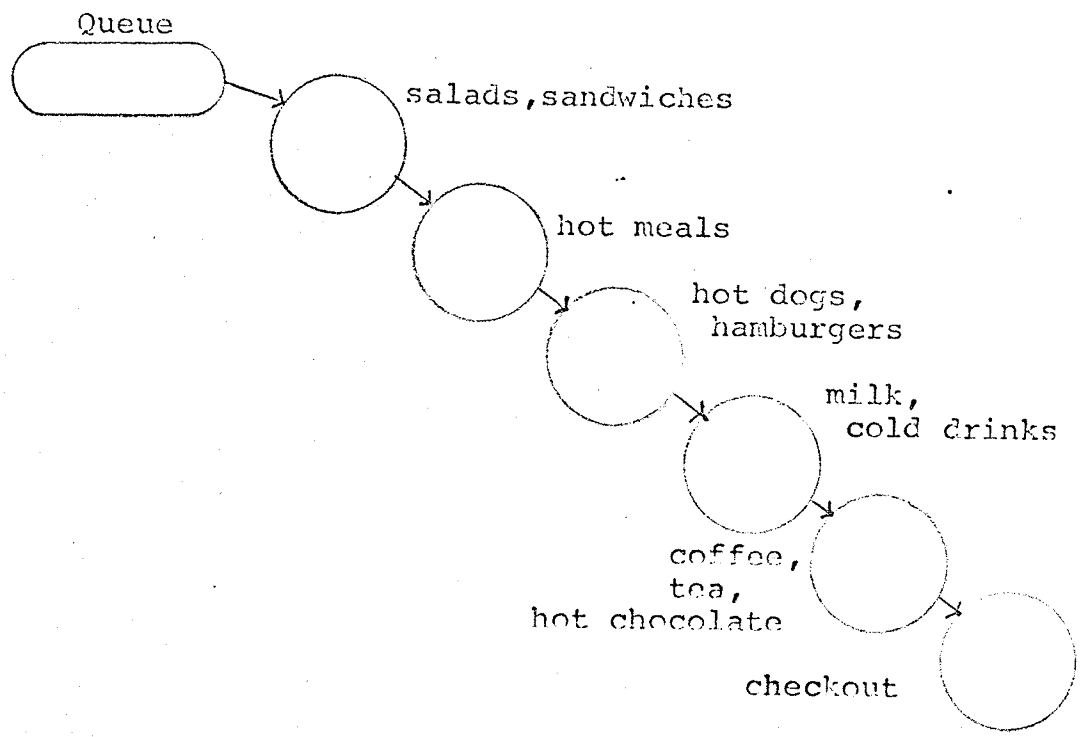


Figure 3.10 Example 2: Cafeteria Model I



Stage 2

The main assumption of this simulation is that the physical structure of figure 3.9 can be approximated by the configuration of queue and facilities defined in figure 3.10.

Another important assumption is the method of representing the flow of transactions through the facilities in such a manner as to subject this system to the same transaction flow as in example 1.

It is important to note that the physical configuration of the cafeteria in example 2 is different from that of example 1. In example 1 the two food types,"cold drinks" and "milk" had individual facilities. In this system these two have been grouped as one food type and are represented as one facility. Therefore there are now five food types instead of six. Each of these five can be chosen with equal probability.

Briefly stated the relevant rules governing the transaction flow in example 1 were;

1. Probability that a transaction will choose a food counter on its first choice is 1.
2. Probability that a transaction will choose a food counter on each of its second and third choice is 6/7.
3. On any one of a transaction's food counter choices each food type has an equal probability of being choosen.

4. The processing time for the counters representing the various food types are:

coffee, tea, hot chocolate	2 units
milk and cold drinks	2 units
hot dogs and hamburgers	3 units
hot meals	5 units
salads and sandwiches	2 units

Assuming one thousand transactions entered the system of example 1 : 1000 food choices would be made as first choice, 6/7 x 1000 as second choices and (6/7 x 1000) x 6/7 as third choices. Therefore the total number of food choices would be 2590.

Since each food type has an equal probability of being chosen, each food type would have received an equal number of these choices. Therefore each of the five food counters in example 2 would have received 2590/5 or 518 choices. To describe this relationship in another manner is to say that for any given counter 51.8 0/0 of the time the entering transactions will have the regular processing time. The remaining 42.8 0/0 entering transactions will have a zero processing time. This relationship is defined for the 2 unit processing time in function 1, for the 3 unit processing time in function 2 and for the 5 units processing time in function 3.

The <u>dynamic structures</u> or transactions in this system are the customers entering, moving through the food counters

and then leaving

and then leaving the system.

The static structures in this system are the various food counters and the checkout isle. These become facilities in figure.

The construction of the minimal system model is illustrated by the flow chart in figure 3.10.

Figure 3.11 Example 2
Generation and Facility Times

Facility Name	Mean Processing Time	Facility NO.
salads,etc.	2 units	10
hot meals	5 units	11
hot dogs,etc.	3 units	12
milk,etc.	2 units	13
coffee,etc.	2 units	14
checkout	4 units	20

Generation Interarrival Time = 2 units.

Note: One unit = 10 seconds.

Stage 3

The determination of the input data requirements led to the following requirements:

1. Mean inter-arrival time for customers entering the system.

2. Facility processing times for the food checkout isle.

These inputs were taken from example 1.

Stage 4

Determination of the output possibilities led to the following conclusions:

1. Facility utilization would be available;

2. Queue length would be available

3. The number of customers in the system would be available.

Stage 5

To prepare precise specification for the final model data was collected from example 1. Figure 3.11 specifies the generation and facility times. Consistent with the method of simulation using this simulation system figure 3.12 was prepared from which the model definition statements of figure 3.13 were prepared.

Figure 3.14 contains the output from the simulation.

3.3.3 Interpretation of Results

Figure 3.14 shows the output from the execution of this model. From this output the following relevant observations can be made:

1. Facilities 10,11,14 and 20 showed relatively high utilizations of .959, .779, .779 and .779, respectively.

2. The single queue which all transactions enter had maximum contents of 10, average contents of 4, and average transaction time of 90 seconds.

From these observations the following conclusions can be made:

1. The system operating under this transaction load is over congested (substantiated, mainly by the excessive queue waiting time).
2. The food counters' high utilization would be a desireable characteristic except for the queue performance. Re-arrangement of the facilities so that lower utilized ones are at the beginning of the system might be a slight improvement.

It should also be noted that in this configuration, since there are no queues between facilities, waiting transactions occupy previous facilities making them inaccessible to customers. This is a major drawback to this type of arrangement. It also serves to raise the facility utilizations falsely, without the corresponding revenue or use of these facilities. To accurately record the useage of these facilities a counter should be placed behind each facility to count the number of customers actually served.

Figure 3.12 Example 2:Model Diagram

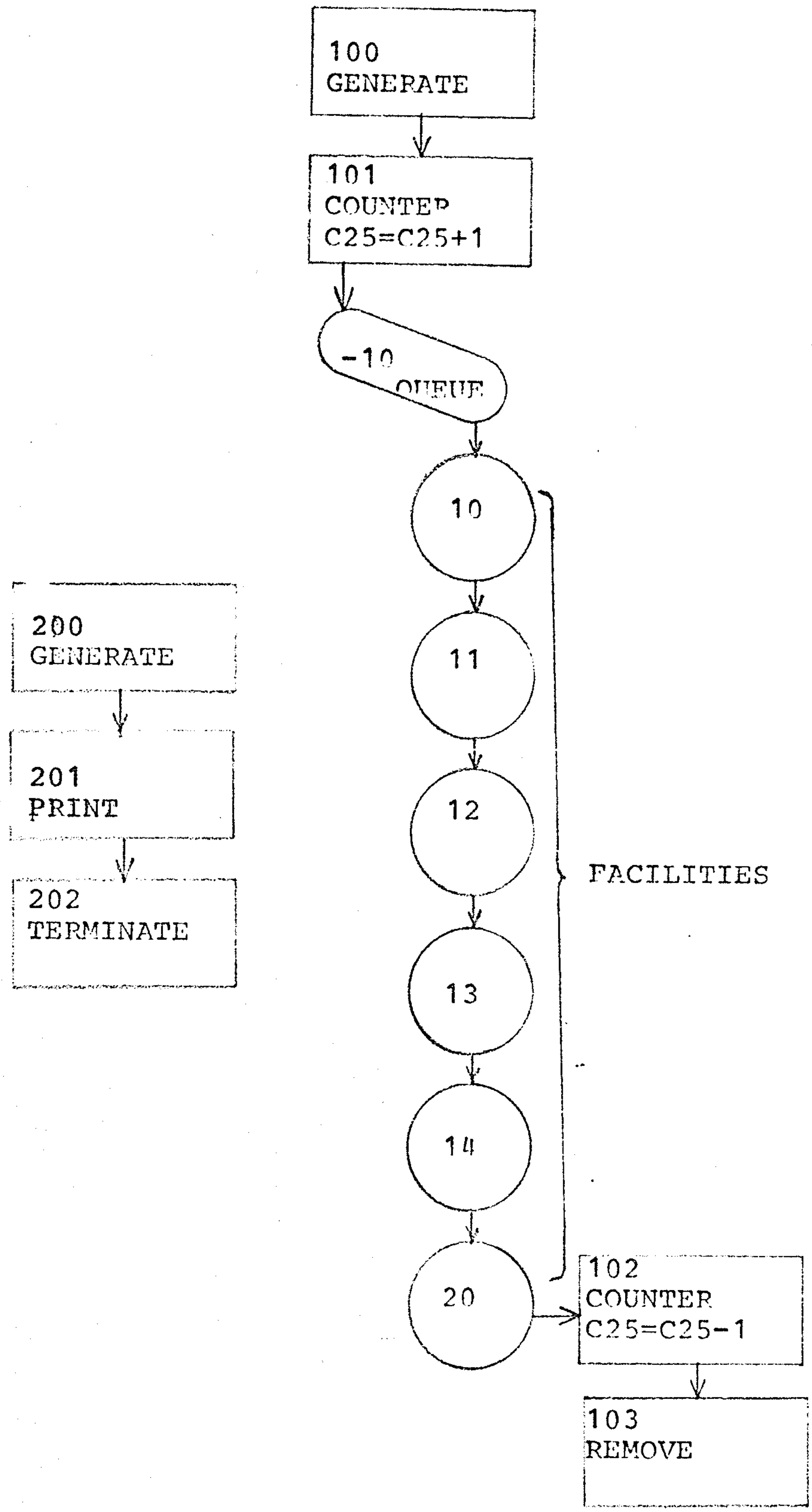


Figure 3.13 Example 2: Model Definition Statements

```
CGEN 100,1,(1,2),0,0,1000,(1,101)
COUNTR 101,8,(1,25),1,(1,1),(1,-10)
Q10 -10,(1,10)
F10 10,(2,1),(1,11)      F11 11,(2,3),(1,12)
F12 12,(2,2),(1,13)      F13 13,(2,1),(1,14)
F14 14,(2,1),(1,20)      F20 20,(1,4),(1,102)
COUNTR 102,8,(1,25),-1,(1,1),(1,103)
CRMVE 103,11
CGEN 200,1,(1,10),0,0,10,(1,201)
CPRT 201,(13,(1,202)
CTERM 202,12,(1,5)
FN1 1,2,4,2,3,(0,0/.518,2/1.0,0)
FN2 2,2,4,2,3,(0,0/.518,3/1.0,0)
FN3 3,2,4,2,3 (0,0/.518,5/1.0,0)
```

```
                    AN ELEMENTARY SIMULATION SYSTEM
THIS SIMULATION SYSTEM REQUIRES THAT THE PHYSICAL SYSTEM TO BE SIMULATED
BE DEFINE BY THE FOLLOWING MODEL DEFINITION STATEMENTS:-
      DECISION RULE (D).....NEXT ENTITY CANDIDATES & SPECIAL DEC. RULES FOR CHOICE
      FACILITY (F)         ...DESCRIPTION OF THE FACILITY
      QUEUE      (Q)       ...DESCRIPTION OF THE QUEUE
      STORAGE   (S)        ...DESCRIPTION OF THE STORAGE
      CONTROL BLOCK  (C) ...DESCRIPTION OF THE CONTROL BLOCK
      FUNCTION (FN)        ...DESCRIPTION OF THE FUNCTION
      VARIABLE EQUATION(V)..DESCRIPTION OF THE EQUATION
      MODEL DEFINITION INPUT SUMMARY AND EDIT
....5...10....5...20....5...30....5...40....5...50....5...60....5...70....5...80
CGEN 100,1,(1,2),0,0,1000,(1,101)
COUNTR 101,8,(1,25),1,(1,1),(1,-10)
Q10 -10,(1,10)
F10 10,(2,1),(1,11)     F11 11,(2,3),(1,12)
F12 12,(2,2),(1,13)      F13 13,(2,1),(1,14)
F14 14,(2,1),(1,20)    F20 20,(1,4),(1,102)
COUNTR 102,8,(1,25),-1,(1,1),(1,103)
CRMVE 103,11
CGEN 200,1,(1,10),0,0,10,(1,201)
CPPT 201,(13,(1,202)
CTERM 202,12,(1,5)
FN1 1,2,4,2,3(0,0/.518,2/1.0,0)
FN2 2,2,4,2,3(0,0/.518,3/1.0,0)
FN3 3,2,4,2,3(0,0/.518,5/1.0,0)




THERE ARE    0 ERRORS IN THE EDIT PHASE.
        SIMULATION IS CONTINUED.
```

Figure 3.14 Example 2: Output

```
TRANSACTION #    1 IS LEAVING ENTITY #  100.
TRANSACTION #    1 IS LEAVING ENTITY #  101.
TRANSACTION #    1 IS LEAVING ENTITY #  -10.
TRANSACTION #    2 IS LEAVING ENTITY #  100.
TRANSACTION #    2 IS LEAVING ENTITY #  101.
TRANSACTION #    2 IS LEAVING ENTITY #  -10.
TRANSACTION #    1 IS LEAVING ENTITY #   10.
TRANSACTION #    2 IS LEAVING ENTITY #   10.
TRANSACTION #    3 IS LEAVING ENTITY #  100.
TRANSACTION #    3 IS LEAVING ENTITY #  101.
TRANSACTION #    3 IS LEAVING ENTITY #  -10.
TRANSACTION #    4 IS LEAVING ENTITY #  100.
TRANSACTION #    4 IS LEAVING ENTITY #  101.
TRANSACTION #    4 IS LEAVING ENTITY #  -10.
TRANSACTION #    1 IS LEAVING ENTITY #   11.
TRANSACTION #    1 IS LEAVING ENTITY #   12.
TRANSACTION #    1 IS LEAVING ENTITY #   13.
TRANSACTION #    2 IS LEAVING ENTITY #   11.
TRANSACTION #    5 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=       0; P2=       0; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=      0; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      4;
CLOCK TIME =   10 UNITS.
TRANSACTION #    5 IS LEAVING ENTITY #  201.
TRANSACTION #    5 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    6 IS LEAVING ENTITY #  100.
TRANSACTION #    6 IS LEAVING ENTITY #  101.
TRANSACTION #    6 IS LEAVING ENTITY #  -10.
TRANSACTION #    1 IS LEAVING ENTITY #   14.
TRANSACTION #    3 IS LEAVING ENTITY #   10.
TRANSACTION #    3 IS LEAVING ENTITY #   11.
TRANSACTION #    2 IS LEAVING ENTITY #   12.
TRANSACTION #    2 IS LEAVING ENTITY #   13.
TRANSACTION #    7 IS LEAVING ENTITY #  100.
TRANSACTION #    7 IS LEAVING ENTITY #  101.
TRANSACTION #    7 IS LEAVING ENTITY #  -10.
TRANSACTION #    4 IS LEAVING ENTITY #   10.
TRANSACTION #    2 IS LEAVING ENTITY #   14.
TRANSACTION #    8 IS LEAVING ENTITY #  100.
TRANSACTION #    8 IS LEAVING ENTITY #  101.
TRANSACTION #    8 IS LEAVING ENTITY #  -10.
TRANSACTION #    1 IS LEAVING ENTITY #   20.
TRANSACTION #    1 IS LEAVING ENTITY #  102.
TRANSACTION #    1 IS REMOVED BY ENTITY #  103.
TRANSACTION #    3 IS LEAVING ENTITY #   12.
TRANSACTION #    3 IS LEAVING ENTITY #   13.
TRANSACTION #    3 IS LEAVING ENTITY #   14.
TRANSACTION #    6 IS LEAVING ENTITY #   10.
TRANSACTION #    9 IS LEAVING ENTITY #  100.
TRANSACTION #    9 IS LEAVING ENTITY #  101.
TRANSACTION #    9 IS LEAVING ENTITY #  -10.
TRANSACTION #    4 IS LEAVING ENTITY #   11.
TRANSACTION #    4 IS LEAVING ENTITY #   12.
TRANSACTION #    4 IS LEAVING ENTITY #   13.
TRANSACTION #    6 IS LEAVING ENTITY #   11.
TRANSACTION #   10 IS LEAVING ENTITY #  100.
TRANSACTION #   10 IS LEAVING ENTITY #  101.
TRANSACTION #   10 IS LEAVING ENTITY #  -10.
TRANSACTION #    2 IS LEAVING ENTITY #   20.
TRANSACTION #    2 IS LEAVING ENTITY #  102.
1250 TRANSACTION #    2 IS REMOVED BY ENTITY #  103.
```

Figure 3.14

```
TRANSACTION #    4 IS LEAVING ENTITY #   14.
TRANSACTION #   11 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=         0; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
C21=         0; C22=         0; C23=         0; C24=         0; C25=         7;
CLOCK TIME =   20 UNITS.
TRANSACTION #   11 IS LEAVING ENTITY #  201.
TRANSACTION #   11 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    7 IS LEAVING ENTITY #   10.
TRANSACTION #   12 IS LEAVING ENTITY #  100.
TRANSACTION #   12 IS LEAVING ENTITY #  101.
TRANSACTION #   12 IS LEAVING ENTITY #  -10.
TRANSACTION #    6 IS LEAVING ENTITY #   12.
TRANSACTION #    6 IS LEAVING ENTITY #   13.
TRANSACTION #    8 IS LEAVING ENTITY #   10.
TRANSACTION #   13 IS LEAVING ENTITY #  100.
TRANSACTION #   13 IS LEAVING ENTITY #  101.
TRANSACTION #   13 IS LEAVING ENTITY #  -10.
TRANSACTION #    3 IS LEAVING ENTITY #   20.
TRANSACTION #    3 IS LEAVING ENTITY #  102.
TRANSACTION #    3 IS REMOVED BY ENTITY #  103.
TRANSACTION #   14 IS LEAVING ENTITY #  100.
TRANSACTION #   14 IS LEAVING ENTITY #  101.
TRANSACTION #   14 IS LEAVING ENTITY #  -10.
TRANSACTION #    7 IS LEAVING ENTITY #   11.
TRANSACTION #    8 IS LEAVING ENTITY #   11.
TRANSACTION #    6 IS LEAVING ENTITY #   14.
TRANSACTION #   15 IS LEAVING ENTITY #  100.
TRANSACTION #   15 IS LEAVING ENTITY #  101.
TRANSACTION #   15 IS LEAVING ENTITY #  -10.
TRANSACTION #    4 IS LEAVING ENTITY #   20.
TRANSACTION #    4 IS LEAVING ENTITY #  102.
TRANSACTION #    4 IS REMOVED BY ENTITY #  103.
TRANSACTION #    9 IS LEAVING ENTITY #   10.
TRANSACTION #    7 IS LEAVING ENTITY #   12.
TRANSACTION #    7 IS LEAVING ENTITY #   13.
TRANSACTION #    7 IS LEAVING ENTITY #   14.
TRANSACTION #    8 IS LEAVING ENTITY #   12.
TRANSACTION #    9 IS LEAVING ENTITY #   11.
TRANSACTION #    9 IS LEAVING ENTITY #   12.
TRANSACTION #   16 IS LEAVING ENTITY #  100.
TRANSACTION #   16 IS LEAVING ENTITY #  101.
TRANSACTION #   16 IS LEAVING ENTITY #  -10.
TRANSACTION #   17 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=         0; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
C21=         0; C22=         0; C23=         0; C24=         0; C25=        10;
CLOCK TIME =   30 UNITS.
TRANSACTION #   17 IS LEAVING ENTITY #  201.
TRANSACTION #   17 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    8 IS LEAVING ENTITY #   13.
TRANSACTION #   10 IS LEAVING ENTITY #   10.
TRANSACTION #   12 IS LEAVING ENTITY #   10.
TRANSACTION #   18 IS LEAVING ENTITY #  100.
TRANSACTION #   18 IS LEAVING ENTITY #  101.
TRANSACTION #   18 IS LEAVING ENTITY #  -10.
TRANSACTION #    6 IS LEAVING ENTITY #   20.
TRANSACTION #    6 IS LEAVING ENTITY #  102.
```

Figure 3.14

```
TRANSACTION #     6 IS REMOVED BY ENTITY #  103.
TRANSACTION #     9 IS LEAVING ENTITY #   13.
TRANSACTION #    19 IS LEAVING ENTITY #  100.
TRANSACTION #    19 IS LEAVING ENTITY #  101.
TRANSACTION #    19 IS LEAVING ENTITY #  -10.
TRANSACTION #     8 IS LEAVING ENTITY #   14.
TRANSACTION #    20 IS LEAVING ENTITY #  100.
TRANSACTION #    20 IS LEAVING ENTITY #  101.
TRANSACTION #    20 IS LEAVING ENTITY #  -10.
TRANSACTION #    10 IS LEAVING ENTITY #   11.
TRANSACTION #    12 IS LEAVING ENTITY #   11.
TRANSACTION #     7 IS LEAVING ENTITY #   20.
TRANSACTION #     7 IS LEAVING ENTITY #  102.
TRANSACTION #     7 IS REMOVED BY ENTITY #  103.
TRANSACTION #    21 IS LEAVING ENTITY #  100.
TRANSACTION #    21 IS LEAVING ENTITY #  101.
TRANSACTION #    21 IS LEAVING ENTITY #  -10.
TRANSACTION #    13 IS LEAVING ENTITY #   10.
TRANSACTION #     9 IS LEAVING ENTITY #   14.
TRANSACTION #    10 IS LEAVING ENTITY #   12.
TRANSACTION #    10 IS LEAVING ENTITY #   13.
TRANSACTION #    13 IS LEAVING ENTITY #   11.
TRANSACTION #    14 IS LEAVING ENTITY #   10.
TRANSACTION #    22 IS LEAVING ENTITY #  100.
TRANSACTION #    22 IS LEAVING ENTITY #  101.
TRANSACTION #    22 IS LEAVING ENTITY #  -10.
TRANSACTION #     8 IS LEAVING ENTITY #   20.
TRANSACTION #     8 IS LEAVING ENTITY #  102.
TRANSACTION #     8 IS REMOVED BY ENTITY #  103.
TRANSACTION #    10 IS LEAVING ENTITY #   14.
TRANSACTION #    23 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=      12;
CLOCK TIME =    40 UNITS.
TRANSACTION #    23 IS LEAVING ENTITY #  201.
TRANSACTION #    23 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    24 IS LEAVING ENTITY #  100.
TRANSACTION #    24 IS LEAVING ENTITY #  101.
TRANSACTION #    24 IS LEAVING ENTITY #  -10.
TRANSACTION #    12 IS LEAVING ENTITY #   12.
TRANSACTION #    12 IS LEAVING ENTITY #   13.
TRANSACTION #    14 IS LEAVING ENTITY #   11.
TRANSACTION #    25 IS LEAVING ENTITY #  100.
TRANSACTION #    25 IS LEAVING ENTITY #  101.
TRANSACTION #    25 IS LEAVING ENTITY #  -10.
TRANSACTION #     9 IS LEAVING ENTITY #   20.
TRANSACTION #     9 IS LEAVING ENTITY #  102.
TRANSACTION #     9 IS REMOVED BY ENTITY #  103.
TRANSACTION #    15 IS LEAVING ENTITY #   10.
TRANSACTION #    13 IS LEAVING ENTITY #   12.
TRANSACTION #    26 IS LEAVING ENTITY #  100.
TRANSACTION #    26 IS LEAVING ENTITY #  101.
TRANSACTION #    26 IS LEAVING ENTITY #  -10.
TRANSACTION #    12 IS LEAVING ENTITY #   14.
TRANSACTION #    13 IS LEAVING ENTITY #   13.
TRANSACTION #    16 IS LEAVING ENTITY #   10.
TRANSACTION #    27 IS LEAVING ENTITY #  100.
TRANSACTION #    27 IS LEAVING ENTITY #  101.
TRANSACTION #    27 IS LEAVING ENTITY #  -10.
TRANSACTION #    10 IS LEAVING ENTITY #   20.
1351 TRANSACTION #    10 IS LEAVING ENTITY #  102.
```

Figure 3.14

```
TRANSACTION #   10 IS REMOVED BY ENTITY #  103.
TRANSACTION #   14 IS LEAVING ENTITY #   12.
TRANSACTION #   14 IS LEAVING ENTITY #   13.
TRANSACTION #   28 IS LEAVING ENTITY #  100.
TRANSACTION #   28 IS LEAVING ENTITY #  101.
TRANSACTION #   28 IS LEAVING ENTITY #  -10.
TRANSACTION #   15 IS LEAVING ENTITY #   11.
TRANSACTION #   15 IS LEAVING ENTITY #   12.
TRANSACTION #   18 IS LEAVING ENTITY #   10.
TRANSACTION #   13 IS LEAVING ENTITY #   14.
TRANSACTION #   29 IS LEAVING ENTITY #  200.
PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=      15;
CLOCK TIME =   50 UNITS.
TRANSACTION #   29 IS LEAVING ENTITY #  201.
TRANSACTION #   29 IS TERMINATED AT ENTITY #  202.
```

Figure 3.14

*** FACILITY STATISTICS ***

FACILITY NO.	TRANSACTION ENTRIES	AVERAGE UTILIZATION	AVERAGE TIME/TRANSACTION
10	15	0.959	3
11	14	0.779	2
12	13	0.559	2
13	12	0.379	1
14	11	0.779	3
20	10	0.779	3

Figure 3.14

*** QUEUE STATISTICS ***

QUEUE NO.	TRANSACTION ENTRIES	AVERAGE CONTENTS	MAXIMUM CONTENTS	NUMBER OF ZERO ENTRIES	AVERAGE TIME/ TRANSACTION	AVE.TIME/TRANS (NO ZERO ENTRIES)
10	24	4	10	2	9	9

Figure 3.14

```
PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=       0; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=      15;
CLOCK TIME =   50 UNITS.

THE SIMULATION HAS BEEN TERMINATED BY     5 TERMINATIONS FOR CONTROL BLOCK # 202.
```

Figure 3.14

3.4 Example 3

3.4.1 Statement of the Problem

This example analyses an inventory situation in which sales orders deplete the inventory until a pre-defined re-order level is reached at which time a purchase order is sent to a supplier who in turn replenishes the inventory. Figure 3.15 illustrates schematically the relevant factors of this situation. The purpose of this experiment is to evaluate the effeciency of:

1. the re-order inventory level,
2. the re-order quantity, and
3. the storage space allowed for this item.

The over-all objective of such a simulation is to arrive at an optimum combination of these three quantities to avoid unfilled sales orders, reduce the number of purchase orders and reduce the storage space allotted for this item.

The criteria for determining the effeciency of any one combination is a cost equation reflecting the cost of allocated storage, the cost of preparing a purchase order and receiving the purchased items and the cost of lost sales if orders are unfilled.

The three factors being examined are related in the following manner:

1. Decreasing the storage space necessitates more purchases and increases the possibility of unfilled sales orders.

2. Raising the re-order level or re-order quantity decreases the possibility of unfilled orders but increases the required storage space.

This problem has been studied in many Operational Research texts (5) (8) as it applies to constant sales demand and constant lead time (i.e. time required for a purchase order to be filled).However, in practice, the frequency of sales orders, their size, the lead times and the percentage of the initial purchase order received vary. By analysing the historical performance of a system a probability distribution can be defined for each of these factors.Fictitious but realistic distributions have been defined to describe the random variations of these factors in this example.

3.4.2 Model Building Procedure

Stage 1 The problem was stated generally in section 3.4.1

The system boundaries are, the entrance of sales orders into the inventory, and the departure of filled sales orders.

The outputs required to evaluate factor combinations are: total cost for the period analysed, the storage utilization, the number of unfilled sales units, and the number of purchases made.

Figure 3.15 Example 3:Inventory Model

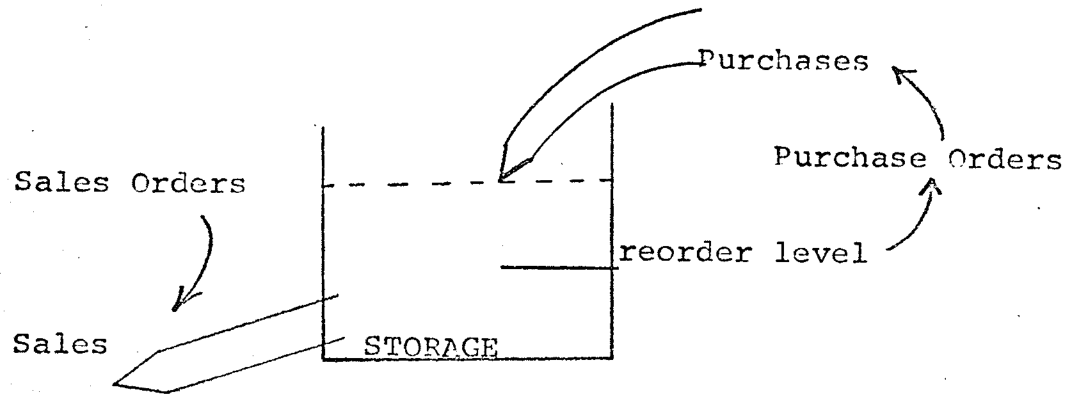
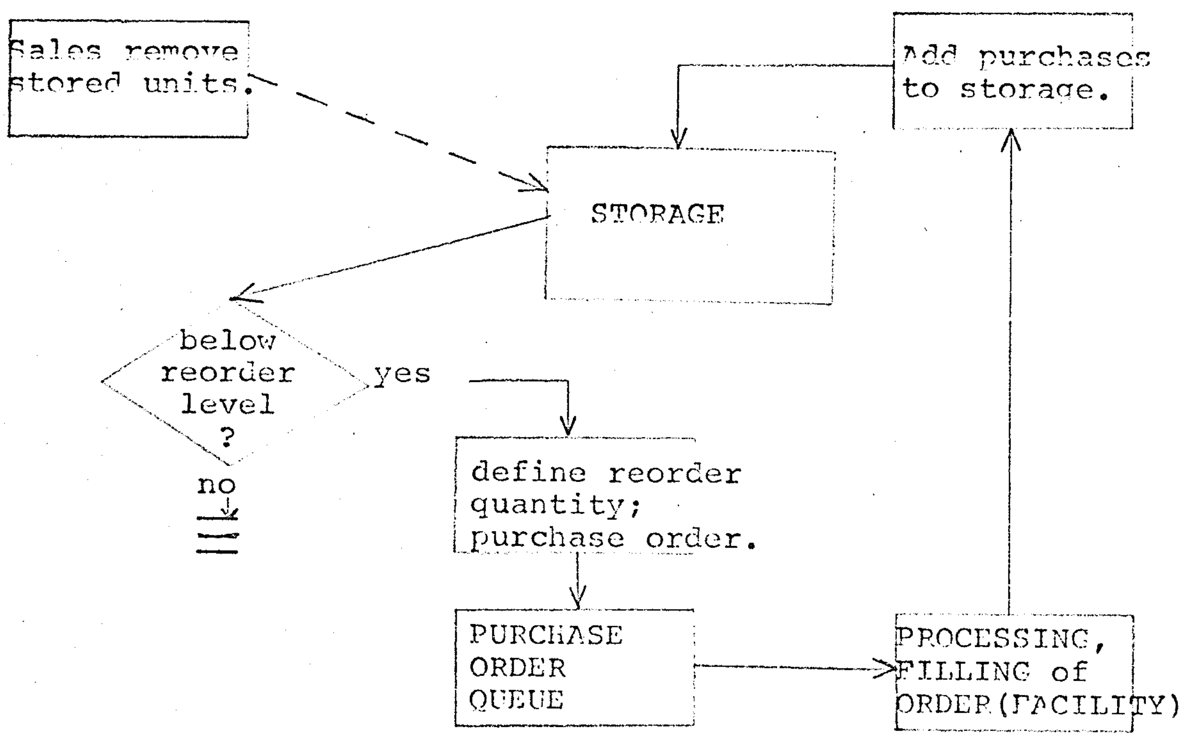


Figure 3.16 Example 3:Minimal System Model



Stage 2

The following assumptions have been made for this simulation:

1. The system defined in figure 3.15 can be represented accurately by figure 3.16.

2. The actual variations of sales order frequency, ordered quantities, lead times and quantities received vary according to the defined density functions.(Figure 3.17.)

The dynamic structures or transactions in this system are the sales orders and the inventory shipments.

The static structures are the storage, the purchase-supplier-receiving systems represented by a facility which accomplishes the simulation of varied lead times.

The construction of the minimal system model is represented by figure 3.16.

The distributions representing the random variations are defined in figure 3.17 .

The assumptions were assessed in the light of stage 1 goals and were found to be reasonable.

Stage 3

The determination of the input data requirements led to the following:

1. Sales order generation interarrival distribution

2. Sales order quantity distribution

3. Re-order level

4. Re-order quantity
5. Storage capacity
6. Lead time distribution.
7. Received-to-purchased order size factor distribution. i.e. a distribution defining the situation of received shipments being more or less than the quantity purchased.
8. Initial storage contents.

Stage 4

Determination of the output possibilities affirmed that the following output requirements would be available:

1. Storage utilization,
2. Average and maximum storage contents,
3. Number of storage units used per week,
4. Number of units of unfilled orders, and
5. Number of orders filled.

Stage 5

To prepare precise specifications for the final model data was collected as described in stage 3. The unit of time for the simulation was chosen as one day. Figure 3.17 illustrates the relevant order generation, lead time, and other distributions. The planned model was then diagrammed in the elementary system's notation (figure 3.18) The corresponding model definition statements are defined in figure 3.19 .

The plan for using this model specified a simulated time of twenty weeks. The various outputs would be analysed with changes in the variable factors (re-order level, re-order quantity and allotted storage) to achieve a least cost combination. The first model execution is described in the following section.

3.4.3 Interpretation of Results

Figure 3.20 shows the output from the first execution of this model. From this output the following relevant observations can be made:

1. All 19 sales orders were filled completely.
2. The number of purchase orders as indicated by a transaction entering block number 111 was 3 these being accomplished by transactions 5,18,and 34.
3. The number of units received into storage was 23 by transaction 5 and 35 by transaction 18. The order from transaction 34 was still being processed at the end of the 20 week interval.
4. An approximation of the number of units in storage (counter 1) at the end of each week was:

week	units	
1	51	
2	44	
3	37	
4	37	
5	36	
6	55	approximation
7	54	
8	53	
9	43	

4. cont'd

week	units
10	35
11	26
12	22
13	50
14	48
15	47
16	41
17	41
18	40
19	33
20	31
	824

5. The average contents of the storage was 40 units.

6. The maximum contents was 60.

7. The average utilization was 8 percent.

From these observations the following conclusions can be made:

1. The total cost of allotted storage at $ .25 per unit per week fixed cost was

   (20 weeks) x(500)x($ .25)= $2500.00

2. The total cost of stored units over the 20 weeks at $1.25 carrying cost per unit was

   (824) x ($1.25) = $1030.00

3. The total cost of purchase order preparation and administration at $30.00 per order was

   (3) x($30.00) =$90.00

4. The total cost through lost sales at $500 per unit was zero.

5. The total cost of operating under the policy of
   a) 500 units of allotted storage
   b) re-order level of 40 units, and
   c) re-order quantity = 70-contents at re-order time was $2500. + 1030. + 90. = $3620.00
6. From observing that the storage utilization is 8 percent, the average contents were 40 units and the maximum contents were 60 units it may be concluded that at the same re-order level and re-order quantity policies the allotted storage can be reduced to a reasonably safe level of 80 to 100 units.

This change and others are left to the reader to execute and continue the analysis of this situation.

Figure 3.17 Example 3 Probability Distributions

Sales Order Arrivals

Sales orders arrive an average of seven days apart. However, six and eight day interarrival durations also occur with equal frequency. Therefore a frequency table of the interarrivals of orders would appear as:

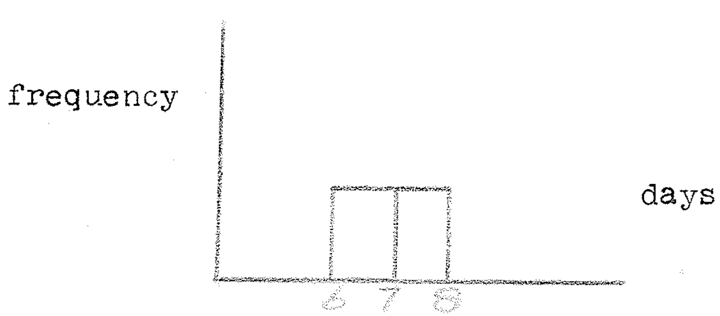


figure 3.17 cont'd

This is the specification of the generation of sales orders into the system.

Sales Order Quantity

The number of units ordered in a given order vary from 1 to 11. The continuous cumulative probability curve representing this relationship is defined by the points:

Order size	Cumulative probability
1	0
2	.2
3	.4
6	.6
8	.8
11	1.0

Function 1 defines this curve.

Re-order Quantities

The re-order level is at 40 units. The re-order quantity is deemed for this execution of the simulation to be 70 minus the present storage contents. The present storage contents will be placed in counter 1. Therefore variable 1, the re-order quantity will be

$$V1 = 60-C1$$

Lead Time

The lead time for an order to be filled when that order reaches the supplier is given by the following probabities...

figure 3.17 cont'd

Prob	Lead Time
.5	21 days
.2	24 "
.1	27 "
.1	30 "

Function 2 defines this relationship.

Size of Received Orders

The quantities received from an order do not always correspond to the quantities ordered. The relationship describing order sizes is illustrated below:

if "R"= ordered quantity

then

.7 R	is	received	with	probability	.1
.8 R	"	"	"	"	.2
.9 R	"	"	"	"	.2
1.0 R	"	"	"	"	.4
1.1 R	"	"	"	"	.1

Function 3 represents these factors multiplied by 10 with the corresponding probabities or occurence.

Variable 2 relates the received quantity with the ordered quantity (parameter 2) and the above factor (parameter 3).

$$V2 = P2*P3/10$$

Figure 3.18 Example 3:Model Diagram

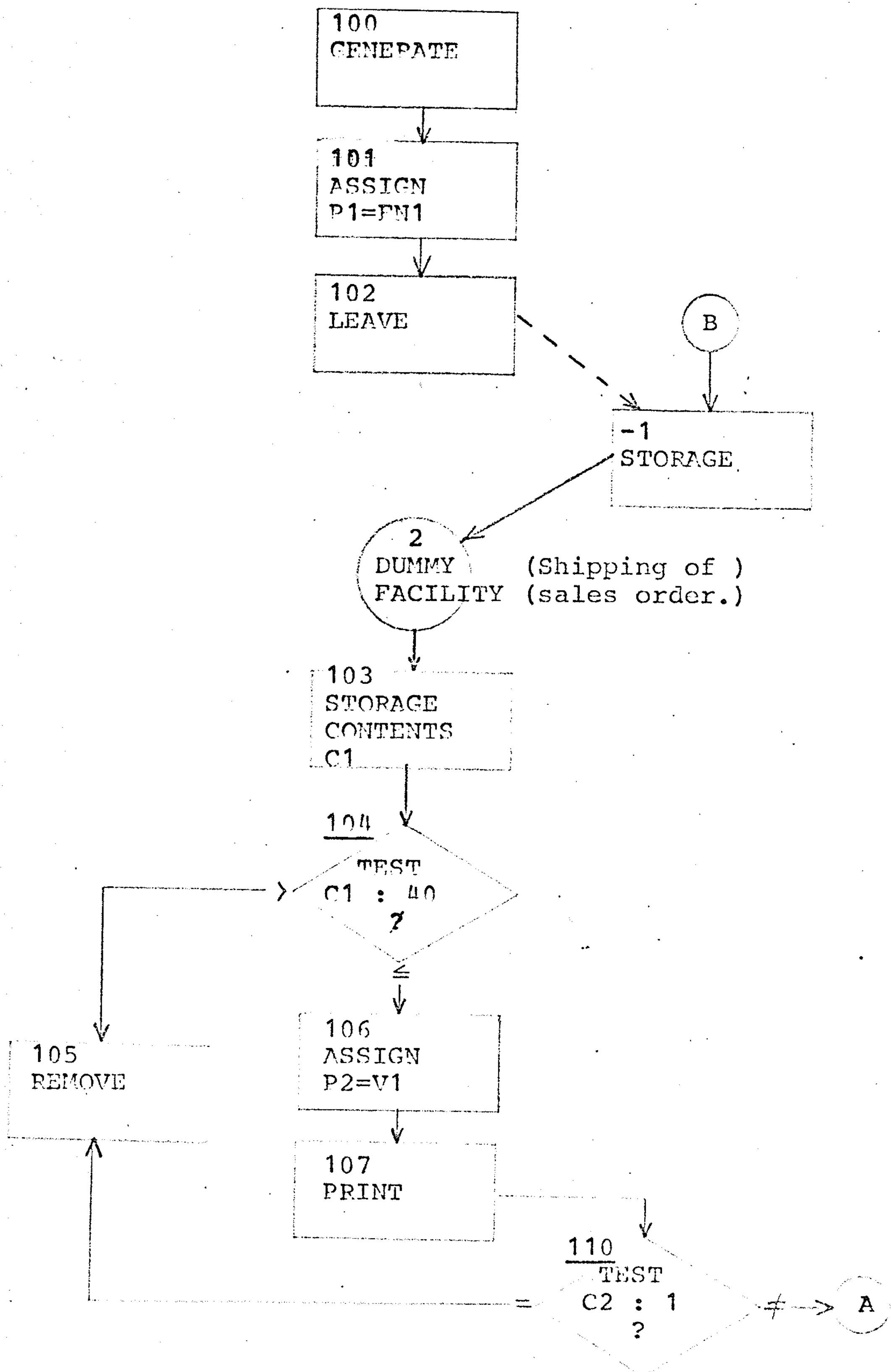


Figure 3.18 continued

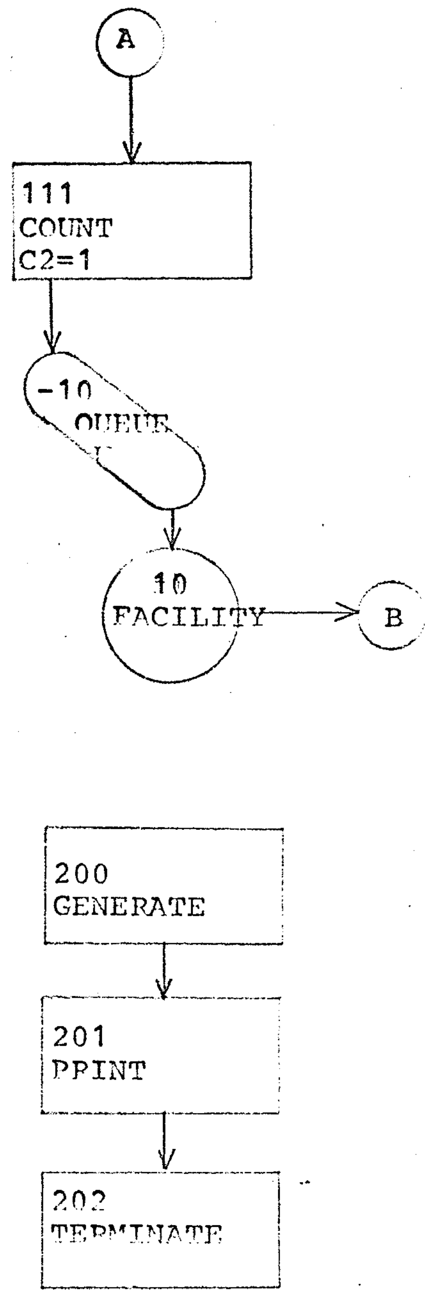


Figure 3.19 Example 3: Model Definition Statements

```
CGEN 100,1,(1,7,1)0,01000,(1,101)
CASGN 101,71(1,1),0,(2,1),(1,102)
CLEAVE 102,10,(1,-1)
CSTRCNT 103,14,-1,1,(1,104)
CTEST 104,9,(5,1),(1,40),106,106,105
CRMVE 105,11
CASGN 106,7,(1,2),0,(3,1),(1,107)
CPRT 107,13,(1,110)
CTEST 110,9,(5,2),(1,1),111,105,111
COUNT 111,8,(1,2),0,(1,1),(1,-10)
QUE -10,(1,10)
FAC 10,(2,2),(1,112)
COUNT 112,8,(1,2),(1,0),(1,108)
CASGN 108,7,(1,3),0,(2,3),(1,-1)
STOR -1,500,60,(3,2),(4,1),(1,2)
FACDUMMY 2,(1,0),(1,103)
  V1= 70-C1                  V2= P2*P3/10
FNORDSIZE 1,2,1,1,6(0,1/.2,2/.4,3/.6,6/.8,8/1.0,11)
FNLEADTIME 2,2,2,2,5(0.0/.5,21/.7,24/.9,27/1.0,30)
FNFACTOR 3,2,3,2,5(.1,7/.3,8/.5,9/.9,10/10,11)
CGENTIMECNTROL 200,1,(1,7),0,0,100,(1,201)
CPRNT 201,13,(1,202)
CTERMINATE 202,12,(1,8)
```

```
                    AN ELEMENTARY SIMULATION SYSTEM
THIS SIMULATION SYSTEM REQUIRES THAT THE PHYSICAL SYSTEM TO BE SIMULATED
BE DEFINE BY THE FOLLOWING MODEL DEFINITION STATEMENTS:-
      DECISION RULE (D).....NEXT ENTITY CANDIDATES & SPECIAL DEC. RULES FOR CHOICE
      FACILITY (F)        ...DESCRIPTION OF THE FACILITY
      QUEUE     (Q)       ...DESCRIPTION OF THE QUEUE
      STORAGE  (S)        ...DESCRIPTION OF THE STORAGE
      CONTROL BLOCK  (C) ...DESCRIPTION OF THE CONTROL BLOCK
      FUNCTION (FN)       ...DESCRIPTION OF THE FUNCTION
      VARIABLE EQUATION(V)..DESCRIPTION OF THE EQUATION
      MODEL DEFINITION INPUT SUMMARY AND EDIT
....5...10....5...20....5...30....5...40....5...50....5...60....5...70....5...80
CGEN 100,1,(1,7,1),0,1000,(1,101)
CASGN 101,7,(1,1),0,(2,1),(1,102)
CLEAVE 102,10,(1,-1)
CSTRCNT 103,14,-1,1,(1,104)
CTEST 104,9,(5,1),(1,40),106,106,105
CRMVE 105,11
CASGN 106,7,(1,2),0,(3,1),(1,107)
CPRT 107,13,(1,110)
CTEST 110,9,(5,2),(1,1),111,105,111
COUNT 111,8,(1,2),0,(1,1),(1,-10)
QUE -10,(1,10)
FAC 10,(2,2),(1,112)
COUNT 112,8,(,1,2),0,(1,0),(1,108)
CASGN 108,7,(1,3),0,(2,3),(1,-1)
STOR -1,500,60,(3,2),(4,1),(1,2)
FACDUMMY 2,(1,0),(1,103)
 V1=70-C1     V2=P2*P3/10
FNORDSIZE 1,2,1,1,6(0,1/.2,2/.4,3/.6,6/.8,8/1.0,11)
FNLEADTIME 2,2,2,2,5(0,0/.5,21/.7,24/.9,27/1.0,30)
FNFACTOR 3,2,3,2,5(.1,7/.3,8/.5,9/.9,10/1.0,11)
CGENTIMECNTROL 200,1,(1,7),0,0,100,(1,201)
CPRNT 201,13,(1,202)
CTERMINATE 202,12,(1,20)
```

Figure 3.20 Example 3:Output

```
TRANSACTION #    1 IS LEAVING ENTITY #  100.
TRANSACTION #    1 IS LEAVING ENTITY #  101.
TRANSACTION #    1 IS LEAVING ENTITY #  102.
TRANSACTION #    1 IS LEAVING ENTITY #   -1.
TRANSACTION #    1 IS LEAVING ENTITY #    2.
TRANSACTION #    1 IS LEAVING ENTITY #  103.
TRANSACTION #    1 IS LEAVING ENTITY #  104.
TRANSACTION #    1 IS REMOVED BY ENTITY #  105.
TRANSACTION #    2 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=      0; P2=      0; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=    51; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=     0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=     0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =     7 UNITS.
TRANSACTION #    2 IS LEAVING ENTITY #  201.
TRANSACTION #    2 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    3 IS LEAVING ENTITY #  100.
TRANSACTION #    3 IS LEAVING ENTITY #  101.
TRANSACTION #    3 IS LEAVING ENTITY #  102.
TRANSACTION #    3 IS LEAVING ENTITY #   -1.
TRANSACTION #    3 IS LEAVING ENTITY #    2.
TRANSACTION #    3 IS LEAVING ENTITY #  103.
TRANSACTION #    3 IS LEAVING ENTITY #  104.
TRANSACTION #    3 IS REMOVED BY ENTITY #  105.
TRANSACTION #    4 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=      0; P2=      0; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=    44; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=     0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=     0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =    14 UNITS.
TRANSACTION #    4 IS LEAVING ENTITY #  201.
TRANSACTION #    4 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    5 IS LEAVING ENTITY #  100.
TRANSACTION #    5 IS LEAVING ENTITY #  101.
TRANSACTION #    5 IS LEAVING ENTITY #  102.
TRANSACTION #    5 IS LEAVING ENTITY #   -1.
TRANSACTION #    5 IS LEAVING ENTITY #    2.
TRANSACTION #    5 IS LEAVING ENTITY #  103.
TRANSACTION #    5 IS LEAVING ENTITY #  104.
TRANSACTION #    5 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=      7; P2=     33; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=    37; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=     0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=     0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =    21 UNITS.
TRANSACTION #    5 IS LEAVING ENTITY #  107.
TRANSACTION #    5 IS LEAVING ENTITY #  110.
TRANSACTION #    5 IS LEAVING ENTITY #  111.
TRANSACTION #    5 IS LEAVING ENTITY #  -10.
TRANSACTION #    6 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=      0; P2=      0; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=    37; C 2=      1; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=     0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
```

Figure 3.20

```
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   21 UNITS.
TRANSACTION #    6 IS LEAVING ENTITY #  201.
TRANSACTION #    6 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    7 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      37; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   28 UNITS.
TRANSACTION #    7 IS LEAVING ENTITY #  201.
TRANSACTION #    7 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    8 IS LEAVING ENTITY #  100.
TRANSACTION #    8 IS LEAVING ENTITY #  101.
TRANSACTION #    8 IS LEAVING ENTITY #  102.
TRANSACTION #    8 IS LEAVING ENTITY #   -1.
TRANSACTION #    8 IS LEAVING ENTITY #    2.
TRANSACTION #    8 IS LEAVING ENTITY #  103.
TRANSACTION #    8 IS LEAVING ENTITY #  104.
TRANSACTION #    8 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=        1; P2=       34; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      36; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   29 UNITS.
TRANSACTION #    8 IS LEAVING ENTITY #  107.
TRANSACTION #    8 IS LEAVING ENTITY #  110.
TRANSACTION #    8 IS REMOVED BY ENTITY #  105.
TRANSACTION #    9 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      36; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   35 UNITS.
TRANSACTION #    9 IS LEAVING ENTITY #  201.
TRANSACTION #    9 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   10 IS LEAVING ENTITY #  100.
TRANSACTION #   10 IS LEAVING ENTITY #  101.
TRANSACTION #   10 IS LEAVING ENTITY #  102.
TRANSACTION #   10 IS LEAVING ENTITY #   -1.
TRANSACTION #   10 IS LEAVING ENTITY #    2.
TRANSACTION #   10 IS LEAVING ENTITY #  103.
TRANSACTION #   10 IS LEAVING ENTITY #  104.
TRANSACTION #   10 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=        4; P2=       38; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      32; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   37 UNITS.
TRANSACTION #   10 IS LEAVING ENTITY #  107.
TRANSACTION #   10 IS LEAVING ENTITY #  110.
TRANSACTION #   10 IS REMOVED BY ENTITY #  105.
TRANSACTION #    5 IS LEAVING ENTITY #   10.
TRANSACTION #    5 IS LEAVING ENTITY #  112.
```

1796

Figure 3.20

```
TRANSACTION #     5 IS LEAVING ENTITY #  108.
TRANSACTION #     5 ENTERING STORAGE #     -1 WITH      23 UNITS.
TRANSACTION #    11 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=        32; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
C21=         0; C22=         0; C23=         0; C24=         0; C25=         0;
CLOCK TIME =    42 UNITS.
TRANSACTION #    11 IS LEAVING ENTITY #  201.
TRANSACTION #    11 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    12 IS LEAVING ENTITY #  100.
TRANSACTION #    12 IS LEAVING ENTITY #  101.
TRANSACTION #    12 IS LEAVING ENTITY #  102.
TRANSACTION #    12 IS LEAVING ENTITY #   -1.
TRANSACTION #    12 IS LEAVING ENTITY #    2.
TRANSACTION #    12 IS LEAVING ENTITY #  103.
TRANSACTION #    12 IS LEAVING ENTITY #  104.
TRANSACTION #    12 IS REMOVED BY ENTITY #  105.
TRANSACTION #    13 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=        54; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
C21=         0; C22=         0; C23=         0; C24=         0; C25=         0;
CLOCK TIME =    49 UNITS.
TRANSACTION #    13 IS LEAVING ENTITY #  201.
TRANSACTION #    13 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    14 IS LEAVING ENTITY #  100.
TRANSACTION #    14 IS LEAVING ENTITY #  101.
TRANSACTION #    14 IS LEAVING ENTITY #  102.
TRANSACTION #    14 IS LEAVING ENTITY #   -1.
TRANSACTION #    14 IS LEAVING ENTITY #    2.
TRANSACTION #    14 IS LEAVING ENTITY #  103.
TRANSACTION #    14 IS LEAVING ENTITY #  104.
TRANSACTION #    14 IS REMOVED BY ENTITY #  105.
TRANSACTION #    15 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=        53; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
C21=         0; C22=         0; C23=         0; C24=         0; C25=         0;
CLOCK TIME =    56 UNITS.
TRANSACTION #    15 IS LEAVING ENTITY #  201.
TRANSACTION #    15 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    16 IS LEAVING ENTITY #  100.
TRANSACTION #    16 IS LEAVING ENTITY #  101.
TRANSACTION #    16 IS LEAVING ENTITY #  102.
TRANSACTION #    16 IS LEAVING ENTITY #   -1.
TRANSACTION #    16 IS LEAVING ENTITY #    2.
TRANSACTION #    16 IS LEAVING ENTITY #  103.
TRANSACTION #    16 IS LEAVING ENTITY #  104.
TRANSACTION #    16 IS REMOVED BY ENTITY #  105.
TRANSACTION #    17 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=          0; P2=          0; P3=          0; P4=          0; P5=          0;
COUNTER VALUES:1 TO 25
C 1=        43; C 2=         0; C 3=         0; C 4=         0; C 5=         0; C 6=         0; C 7=         0; C 8=         0; C 9=         0; C10=         0;
C11=         0; C12=         0; C13=         0; C14=         0; C15=         0; C16=         0; C17=         0; C18=         0; C19=         0; C20=         0;
```

Figure 3.20

```
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   63 UNITS.
TRANSACTION #   17 IS LEAVING ENTITY #  201.
TRANSACTION #   17 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   18 IS LEAVING ENTITY #  100.
TRANSACTION #   18 IS LEAVING ENTITY #  101.
TRANSACTION #   18 IS LEAVING ENTITY #  102.
TRANSACTION #   18 IS LEAVING ENTITY #   -1.
TRANSACTION #   18 IS LEAVING ENTITY #    2.
TRANSACTION #   18 IS LEAVING ENTITY #  103.
TRANSACTION #   18 IS LEAVING ENTITY #  104.
TRANSACTION #   18 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=       8; P2=      35; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=      35; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   64 UNITS.
TRANSACTION #   18 IS LEAVING ENTITY #  107.
TRANSACTION #   18 IS LEAVING ENTITY #  110.
TRANSACTION #   18 IS LEAVING ENTITY #  111.
TRANSACTION #   18 IS LEAVING ENTITY #  -10.
TRANSACTION #   19 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=       0; P2=       0; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=      35; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   70 UNITS.
TRANSACTION #   19 IS LEAVING ENTITY #  201.
TRANSACTION #   19 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   20 IS LEAVING ENTITY #  100.
TRANSACTION #   20 IS LEAVING ENTITY #  101.
TRANSACTION #   20 IS LEAVING ENTITY #  102.
TRANSACTION #   20 IS LEAVING ENTITY #   -1.
TRANSACTION #   20 IS LEAVING ENTITY #    2.
TRANSACTION #   20 IS LEAVING ENTITY #  103.
TRANSACTION #   20 IS LEAVING ENTITY #  104.
TRANSACTION #   20 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=       9; P2=      44; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=      26; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   71 UNITS.
TRANSACTION #   20 IS LEAVING ENTITY #  107.
TRANSACTION #   20 IS LEAVING ENTITY #  110.
TRANSACTION #   20 IS REMOVED BY ENTITY #  105.
TRANSACTION #   21 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=       0; P2=       0; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=      26; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   77 UNITS.
TRANSACTION #   21 IS LEAVING ENTITY #  201.
TRANSACTION #   21 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   22 IS LEAVING ENTITY #  100.
```

1797

Figure 3.20

```
TRANSACTION #   22 IS LEAVING ENTITY #  101.
TRANSACTION #   22 IS LEAVING ENTITY #  102.
TRANSACTION #   22 IS LEAVING ENTITY #   -1.
TRANSACTION #   22 IS LEAVING ENTITY #    2.
TRANSACTION #   22 IS LEAVING ENTITY #  103.
TRANSACTION #   22 IS LEAVING ENTITY #  104.
TRANSACTION #   22 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=        4; P2=       48; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      22; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =    77 UNITS.
TRANSACTION #   22 IS LEAVING ENTITY #  107.
TRANSACTION #   22 IS LEAVING ENTITY #  110.
TRANSACTION #   22 IS REMOVED BY ENTITY #  105.
TRANSACTION #   23 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      22; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =    84 UNITS.
TRANSACTION #   23 IS LEAVING ENTITY #  201.
TRANSACTION #   23 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   24 IS LEAVING ENTITY #  100.
TRANSACTION #   24 IS LEAVING ENTITY #  101.
TRANSACTION #   24 IS LEAVING ENTITY #  102.
TRANSACTION #   24 IS LEAVING ENTITY #   -1.
TRANSACTION #   24 IS LEAVING ENTITY #    2.
TRANSACTION #   24 IS LEAVING ENTITY #  103.
TRANSACTION #   24 IS LEAVING ENTITY #  104.
TRANSACTION #   24 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=        1; P2=       49; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      21; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =    84 UNITS.
TRANSACTION #   24 IS LEAVING ENTITY #  107.
TRANSACTION #   24 IS LEAVING ENTITY #  110.
TRANSACTION #   24 IS REMOVED BY ENTITY #  105.
TRANSACTION #   18 IS LEAVING ENTITY #   10.
TRANSACTION #   18 IS LEAVING ENTITY #  112.
TRANSACTION #   18 IS LEAVING ENTITY #  108.
TRANSACTION #   18 ENTERING STORAGE #    -1 WITH       35 UNITS.
TRANSACTION #   25 IS LEAVING ENTITY #  100.
TRANSACTION #   25 IS LEAVING ENTITY #  101.
TRANSACTION #   25 IS LEAVING ENTITY #  102.
TRANSACTION #   25 IS LEAVING ENTITY #   -1.
TRANSACTION #   25 IS LEAVING ENTITY #    2.
TRANSACTION #   25 IS LEAVING ENTITY #  103.
TRANSACTION #   25 IS LEAVING ENTITY #  104.
TRANSACTION #   25 IS REMOVED BY ENTITY #  105.
TRANSACTION #   26 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      50; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
```

Figure 3.20

```
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   91 UNITS.
TRANSACTION #   26 IS LEAVING ENTITY #  201.
TRANSACTION #   26 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   27 IS LEAVING ENTITY #  100.
TRANSACTION #   27 IS LEAVING ENTITY #  101.
TRANSACTION #   27 IS LEAVING ENTITY #  102.
TRANSACTION #   27 IS LEAVING ENTITY #   -1.
TRANSACTION #   27 IS LEAVING ENTITY #    2.
TRANSACTION #   27 IS LEAVING ENTITY #  103.
TRANSACTION #   27 IS LEAVING ENTITY #  104.
TRANSACTION #   27 IS REMOVED BY ENTITY #  105.
TRANSACTION #   28 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      48; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =   98 UNITS.
TRANSACTION #   28 IS LEAVING ENTITY #  201.
TRANSACTION #   28 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   29 IS LEAVING ENTITY #  100.
TRANSACTION #   29 IS LEAVING ENTITY #  101.
TRANSACTION #   29 IS LEAVING ENTITY #  102.
TRANSACTION #   29 IS LEAVING ENTITY #   -1.
TRANSACTION #   29 IS LEAVING ENTITY #    2.
TRANSACTION #   29 IS LEAVING ENTITY #  103.
TRANSACTION #   29 IS LEAVING ENTITY #  104.
TRANSACTION #   29 IS REMOVED BY ENTITY #  105.
TRANSACTION #   30 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      47; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =  105 UNITS.
TRANSACTION #   30 IS LEAVING ENTITY #  201.
TRANSACTION #   30 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   31 IS LEAVING ENTITY #  100.
TRANSACTION #   31 IS LEAVING ENTITY #  101.
TRANSACTION #   31 IS LEAVING ENTITY #  102.
TRANSACTION #   31 IS LEAVING ENTITY #   -1.
TRANSACTION #   31 IS LEAVING ENTITY #    2.
TRANSACTION #   31 IS LEAVING ENTITY #  103.
TRANSACTION #   31 IS LEAVING ENTITY #  104.
TRANSACTION #   31 IS REMOVED BY ENTITY #  105.
TRANSACTION #   32 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      41; C 2=       0; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =  112 UNITS.
TRANSACTION #   32 IS LEAVING ENTITY #  201.
TRANSACTION #   32 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   33 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
```

1798

Figure 3.20

```
COUNTER VALUES:1 TO 25
C 1=     41; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  119 UNITS.
TRANSACTION #    33 IS LEAVING ENTITY #  201.
TRANSACTION #    33 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    34 IS LEAVING ENTITY #  100.
TRANSACTION #    34 IS LEAVING ENTITY #  101.
TRANSACTION #    34 IS LEAVING ENTITY #  102.
TRANSACTION #    34 IS LEAVING ENTITY #   -1.
TRANSACTION #    34 IS LEAVING ENTITY #    2.
TRANSACTION #    34 IS LEAVING ENTITY #  103.
TRANSACTION #    34 IS LEAVING ENTITY #  104.
TRANSACTION #    34 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=      1; P2=     30; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=     40; C 2=      0; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  120 UNITS.
TRANSACTION #    34 IS LEAVING ENTITY #  107.
TRANSACTION #    34 IS LEAVING ENTITY #  110.
TRANSACTION #    34 IS LEAVING ENTITY #  111.
TRANSACTION #    34 IS LEAVING ENTITY #  -10.
TRANSACTION #    35 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=      0; P2=      0; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=     40; C 2=      1; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  126 UNITS.
TRANSACTION #    35 IS LEAVING ENTITY #  201.
TRANSACTION #    35 IS TERMINATED AT ENTITY #  202.
TRANSACTION #    36 IS LEAVING ENTITY #  100.
TRANSACTION #    36 IS LEAVING ENTITY #  101.
TRANSACTION #    36 IS LEAVING ENTITY #  102.
TRANSACTION #    36 IS LEAVING ENTITY #   -1.
TRANSACTION #    36 IS LEAVING ENTITY #    2.
TRANSACTION #    36 IS LEAVING ENTITY #  103.
TRANSACTION #    36 IS LEAVING ENTITY #  104.
TRANSACTION #    36 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=      7; P2=     37; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=     33; C 2=      1; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  126 UNITS.
TRANSACTION #    36 IS LEAVING ENTITY #  107.
TRANSACTION #    36 IS LEAVING ENTITY #  110.
TRANSACTION #    36 IS REMOVED BY ENTITY #  105.
TRANSACTION #    37 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=      0; P2=      0; P3=      0; P4=      0; P5=      0;
COUNTER VALUES:1 TO 25
C 1=     33; C 2=      1; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  133 UNITS.
```

Figure 3.20

```
TRANSACTION #   37 IS LEAVING ENTITY #  201.
TRANSACTION #   37 IS TERMINATED AT ENTITY #  202.
TRANSACTION #   38 IS LEAVING ENTITY #  100.
TRANSACTION #   38 IS LEAVING ENTITY #  101.
TRANSACTION #   38 IS LEAVING ENTITY #  102.
TRANSACTION #   38 IS LEAVING ENTITY #   -1.
TRANSACTION #   38 IS LEAVING ENTITY #    2.
TRANSACTION #   38 IS LEAVING ENTITY #  103.
TRANSACTION #   38 IS LEAVING ENTITY #  104.
TRANSACTION #   38 IS LEAVING ENTITY #  106.

PARAMETER VALUES:1 TO 5
P1=        2; P2=       39; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      31; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =  134 UNITS.
TRANSACTION #   38 IS LEAVING ENTITY #  107.
TRANSACTION #   38 IS LEAVING ENTITY #  110.
TRANSACTION #   38 IS REMOVED BY ENTITY #  105.
TRANSACTION #   39 IS LEAVING ENTITY #  200.

PARAMETER VALUES:1 TO 5
P1=        0; P2=        0; P3=        0; P4=        0; P5=        0;
COUNTER VALUES:1 TO 25
C 1=      31; C 2=       1; C 3=       0; C 4=       0; C 5=       0; C 6=       0; C 7=       0; C 8=       0; C 9=       0; C10=       0;
C11=       0; C12=       0; C13=       0; C14=       0; C15=       0; C16=       0; C17=       0; C18=       0; C19=       0; C20=       0;
C21=       0; C22=       0; C23=       0; C24=       0; C25=       0;
CLOCK TIME =  140 UNITS.
TRANSACTION #   39 IS LEAVING ENTITY #  201.
TRANSACTION #   39 IS TERMINATED AT ENTITY #  202.
```

Figure 3.20

*** FACILITY STATISTICS ***

FACILITY NO.	TRANSACTION ENTRIES	AVERAGE UTILIZATION	AVERAGE TIME/TRANSACTION
10	3	0.464	21
2	19	0.0	0

*** STORAGE STATISTICS ***

STORAGE NO.	TOT. UNITS ENTERED	AVERAGE UTILIZATION	AVERAGE CONTENTS	AVERAGE TIME/ TRANSACTION	MAXIMUM CONTENTS	CAPACITY	CURRENT CONTENTS
-1	118	0.081	40	48	60	500	31

*** QUEUE STATISTICS ***

QUEUE NO.	TRANSACTION ENTRIES	AVERAGE CONTENTS	MAXIMUM CONTENTS	NUMBER OF ZERO ENTRIES	AVERAGE TIME/ TRANSACTION	AVE.TIME/TRANS (NO ZERO ENTRIES)
-10	3	0	1	3	0	0

```
PARAMETER VALUES:1 TO 5
P1=       0; P2=       0; P3=       0; P4=       0; P5=       0;
COUNTER VALUES:1 TO 25
C 1=     31; C 2=      1; C 3=      0; C 4=      0; C 5=      0; C 6=      0; C 7=      0; C 8=      0; C 9=      0; C10=      0;
C11=      0; C12=      0; C13=      0; C14=      0; C15=      0; C16=      0; C17=      0; C18=      0; C19=      0; C20=      0;
C21=      0; C22=      0; C23=      0; C24=      0; C25=      0;
CLOCK TIME =  140 UNITS.
THE SIMULATION HAS BEEN TERMINATED BY    20 TERMINATIONS FOR CONTROL BLOCK # 202.
```

Figure 3.20

3.5 Review of the Examples

The purpose of this section is similiar to that of section 2.16, that is, to caution the student that although this technique of simulation may appear relatively simple, it is not always so. With reference to the three examples in this chapter a general consideration is mentioned.

The execution of a simulation with this system begins with no transaction in the model. In actual fact except at opening times does a system contain no entries. Therefore when simulating a system which does have entries in it at the start of the interval to be simulated;the simulation must be run long enough with enough transaction entries to allow the system to reach its regular pattern of operation. This regular pattern of operation is called the steady state. It is not uncommon for a several thousand transactions to be run through a model, thus reaching a steady state and subsequently accurately simulating the actual system's performance.

CHAPTER IV

System Description

4.1 Basic System Concepts

The basic system concepts are most easily described when related to the general purpose of the system. This purpose was to facilitate the translation of a model of a real system into a computer executable form. More specifically, this was interpreted to mean, firstly, the virtual elemination of programming language with its implied grammar, systax and application intuition all which require education and experience. And, secondly, to reduce the time and effort to define the model specifications in the executable form.

In addition, to make the system useful and practical a certain set of simulation facilities was deemed necessary. Thus the concepts of the physical entities, the facility, storage and queue were essential. Secondly the concept of a transaction oriented simulation system was chosen. A transaction oriented system was chosen over a descriptive system for the following reasons:

1. The wide popularity of GPSS, a transaction oriented system,
2. Its ease of comprehension for the beginning student, and,

3. Its suitability to an introductory model building procedure which relies on a diagram of the real world system through which the real transactions flow. In essence, its similarity to the structure of the real world system.

Thirdly, another essential concept of a simulation system is a means to define the logic rules governing the flow of transactions through the physical entities.

The forth concept which requires representation in a useable simulation system is that of the mathematical relationships both stochastic and deterministic which exist between the factors of a real world system.

The implementation of these basic concepts was accomplished by the model definition statements described in Chapter 2, and briefly mentioned here. The physical entities' characteristics are defined by the model definition statements Facility, Queue and Storage. The logic rules are defined by Control block model definition statements. FUnction (or FuNction) and Variable definition statements accomplish the specification of stochastic and deterministic relationships. Inherent in the use of all these model definition statements is the transaction concept.

Given that the model builder has the concepts of the physical entities, control blocks and mathematical relationships, all within a general transaction framework, the problem remains of translating the configuration and characteristics of the model into an executable form. Added to this basic problem was the objective of avoiding the necessity for the model builder to be an experienced programmer. Thus the notation must be natural.

Further analysis of models in general led to the conclusion that a model can be described numerically.

1. The physical entity characteristics are numeric values.
2. The flow of transactions through the model can be described numerically, that is, control block-attributes can be given by numbers.

Such a numeric representation of the model reduces the problem of creating the computer's interval model from the external representation to a very simple one. All the problems of translating a programming language are avoided.

The objective of designing a simple numeric coding scheme was achieved by a simple numbering scheme which attaches to each box in the model diagram a unique number. These numbers combined with the observation that the entire model configuration can be isolated to be a series of simple next box decisions, one occurring at each box in the model's diagram. Therefore at each physical entity and control block, as well as having the numerical characteristics of that entity or block, one additional specification defining the next entity or block was required to satisfy the objective of a single set of real numbers which alone define the entire model configuration and its characteristics.

The final implementation of the function of inputting a set of model definition numbers is that model definition statement. This statement in essence,,is simply the formatted set of numerical attributes necessary to specify each physical entity or control block. The identification of the statement is accomplished by one of the alphabetic letters F for each facility, S for each storage, Q for each queue, and C for each control block. FU or FN for each function and V for each variable equation. This letter is the first character of the statement.The numerical attributes corresponding to each statement type follow the identifying character. Each statement type has a different number of attributes. The interpretation of each number is related to its position in the sequence of numbers for each statement. Apart from the above specifications for the statements they are free-formatted. The detailed specifications are defined in Chapter 2.

Therefore, in summary, the basic concepts of this elementary simulation system are;

1. the physical entities; facility, storage, and queue,
2. control blocks to define logic rules,
3. the representation of stochastic and deterministic relationships, and
4. the general framework of a transaction oriented simulation system.

Subsequent sections of this chapter describe the significant aspects of the implementation of these concepts.

4.2 General System Flow

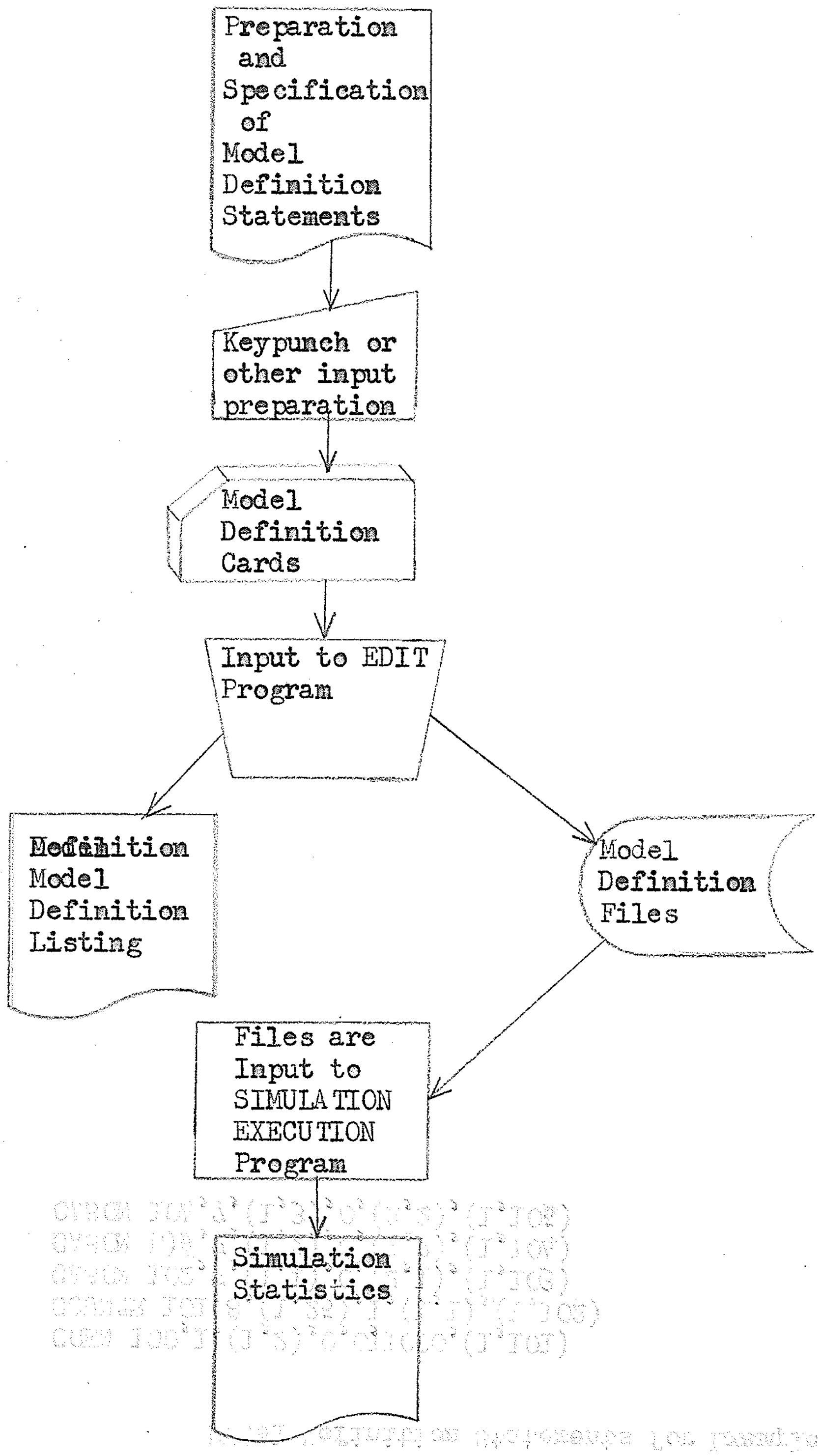
There are two main sections to the general system. The first section is an edit stage. The model definition statements are input to this section. Each statement is analysed and the numerical attributes are isolated. Error checking for statement format; the number of numerical arguments for each statement type and the analysis of valid argument ranges is performed. Appendix A lists the errors detected in the edit stage. If no errors are detected in the edit stage five files are created by the edit program. These files contain the complete definition of the model in a form acceptable to the second stage of the simulation system.

The second stage of the system performs the simulation of the model. Four of the five files from the edit stage are GPSS or Fortran routines which are combined with the permentant GPSS and Fortran programs. The resultant program is compiled and executed. The fifth file is read in during the execution stage of the simulation program. The execution of the simulation produces facility, storage and queue performance statistics, transaction paths and execution error analysis.

Figure 4.1 illustrates the general flow of the simulation system.

Figure 4.1

General System Flow



4.3 The Edit Program

The Edit program is the first section of the system. This program analyses the model definition statements and produces as output, input to the second section. The program is written in PL/1 (F-level) IBM 360/65 under the Operating System. (OS-Release 18)

The general flow of the program is described in figure 4.2. Each model definition statement is analysed as a unit. A separate analysing routine exists for each statement type. When the statement has been identified program control is given to the corresponding routine.

Each analysing routine, although unique incorporates a similiar method of analysis. This is described in figure 4.3. As a statement is analysed its numeric values are placed in the corresponding matrix for later output to the second stage.

The basic functions of each analysing routine are presented in tabular form in figure 4.4.

The use of each of the five files is described in figure 4.5.

When the last model definition statement has been analysed, and the error checking has been completed with no resulting errors detected, control is given to an output routine. The function of this routine is described in figure 4.6.

Appendix G contains a listing with sample output of the Edit program.

Figure 4.2

General Flow of EDIT Program

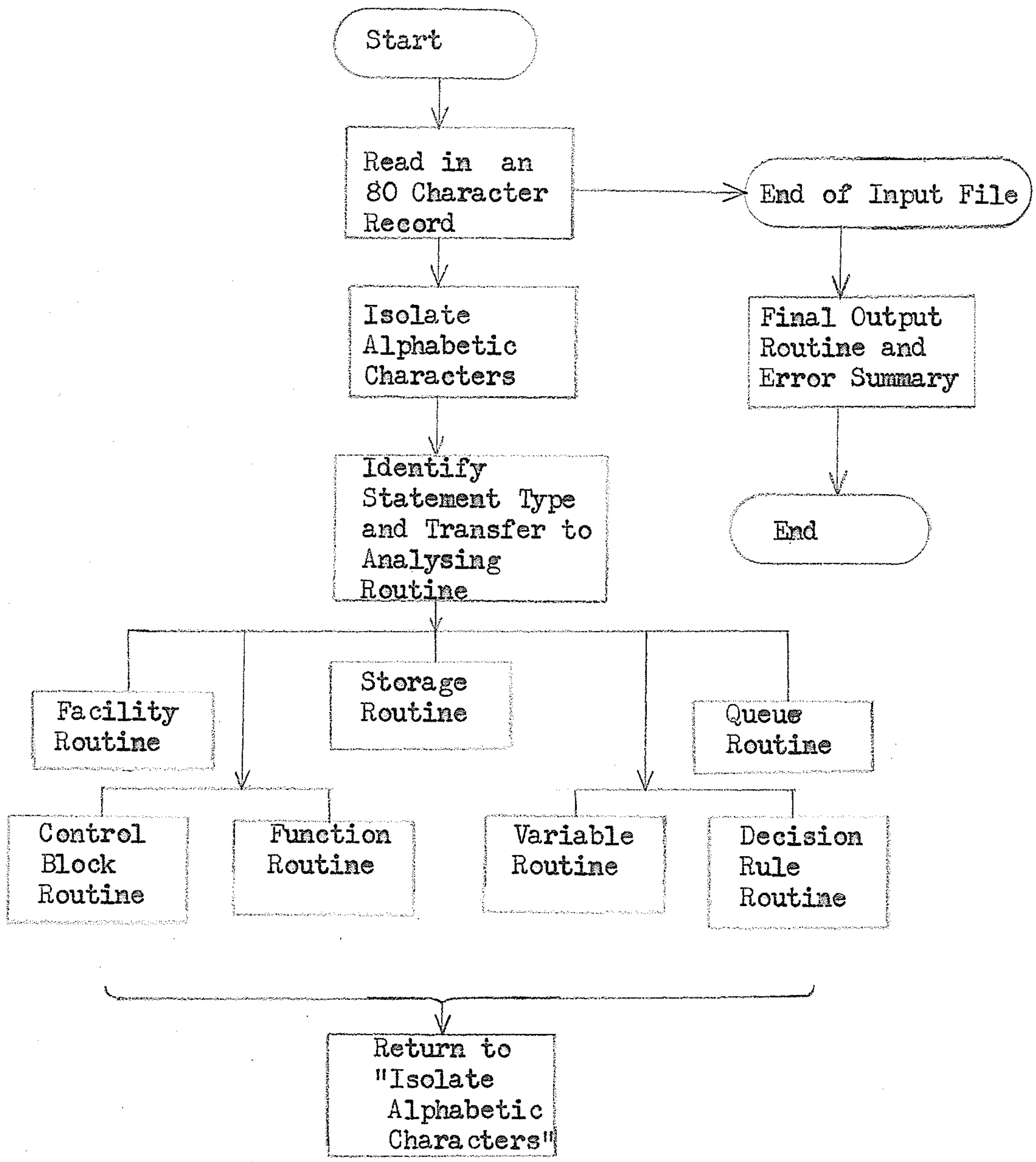


Figure 4.3

Basic Flow of the Model Definition Statement Analysing Routines

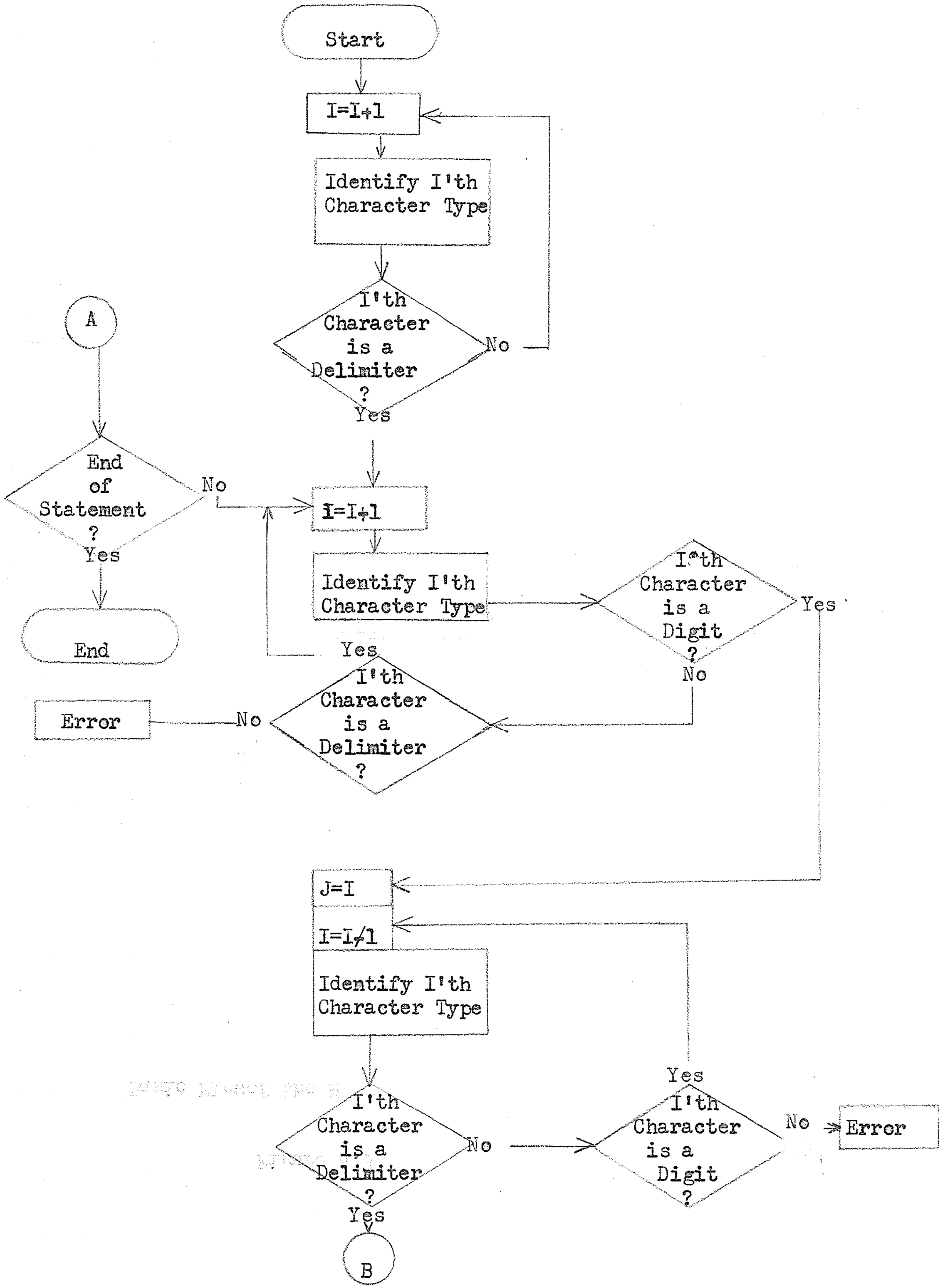


Figure 4.3 continued

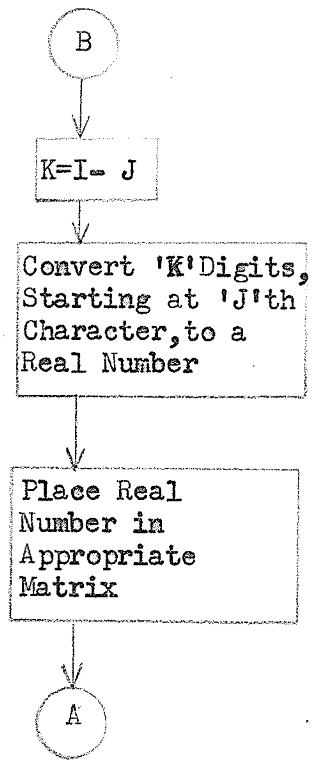


Figure 4.4

Purposes of the Model Definition Statement Analysing Routines

Routine: Facility

Purposes:

1. Isolate each real number.
2. Place number into facility matrix.
3. Maintain highest facility number.
4. Count the number of facilities.

Routine: Storage

Purposes:

1. Isolate each real number.
2. Place number into storage matrix.
3. Maintain lowest storage number.
4. Count the number of storages.
5. Create a GPSS "STORAGE" statement for each storage.
6. Write to DISK 1 each GPSS "STORAGE" statement.

Routine: Queue

Purposes:

1. Isolate each real number.
2. Place number into queue matrix.
3. Maintain highest queue number.
4. Count the number of queues.

figure 4.4 cont'd

Routine: Function

Purposes:

1. Isolate each real number.
2. Translate numbers into GPSS "FUNCTION" statement.
3. Write GPSS "FUNCTION" statement to DISK3.
4. Maintain the defined function numbers.

Routine: Variable

Purposes:

1. Convert keywords into Fortran programming code.
2. Analysis variable expression for valid Fortran syntax.
3. Create a Fortran assignment statement.
4. Write Fortran statements to DISK5.

Routine: Decision Rule

Purposes:

1. Isolate each real number.
2. Place numbers into decision rule matrix.
3. Count the number of decision rules.

Routine: Control Blocks

Purposes:

1. Isolate each real number.
2. Place numbers into control block matrix.

figure 4.4 cont'd

3. Maintain lowest control block number.
4. Count the number of control blocks.
5. For "generate" control blocks:
    a) create GPSS "GENERATE","ASSIGN", and "TRANSFER" statements,
    b) write these statements to DISK4.

Figure 4.5

Disk File Useage

Name: DISK1

Uses:

1.Contains the GPSS"STORAGE" definition statements.

2.Contains a GPSS save value "INITIAL" statement which defines the number of storages.

Name: DISK2

Uses:

1.none

Name: DISK3

Uses:

1.Contains the GPSS "FUNCTION" definition statements.

Name: DISK4

Uses:

1. For each "generate" control block the GPSS statements "GENERATE", "ASSIGN," AND"TRANSFER:" are written into this file.

Name: DISK5

Uses:

1. Contains in the general form a FORTRAN Subroutine Fortran mathematical assignment statements which reflect the definition of each variable.

figure 4.5 cont'd

Name: DISK6

Uses:

1. Contains the contents of each of the facility, storage, queue, control block and decision rule matrices as well as a vector which contains the number of each type of entity, block, or rule and the maximum and minimum identity numbers of the facilities, storages, queues and control blocks.

Figure 4.6

Purposes of the Output Routine

Routine: Final Output

Purposes:

1. To check that facility and control block identity numbers do not overlap.
2. To check that storage and queue identity numbers do not overlap.
3. To check that those functions referenced by "generate" control blocks have been defined.
4. To check that all defined variables are properly numbered.
5. To output to DISK6 the contents of each matrix.
6. To output to DISK6 the number of entities, blocks and rules defined.
7. To output to DISK6 the maximum facility number, the minimum queue number, the maximum control block number and the maximum queue and storage numbers.

4.4 The Simulation Program

4.4.1 General Description

The simulation program consists of three individual job steps. The first two of these are necessary to incorporate the Fortran and GPSS program segments into the main Fortran and GPSS programs. The third step is the actual execution of the simulation. These three steps are illustrated in figure 4.7.

In the first step the following operations are performed:

1. File DISK5, the file containing the Fortran subroutine for the execution of the variable equations is compiled.
2. The resulting object module is added to the main Fortran simulation program. (see Appendix D)

In the second step the files DISK1, DISK3, and DISK4 are concatenated with three main GPSS source code files to produce the complete GPSS simulation program. Figure 4.7 illustrates the combination of files. Appendix E contains the three GPSS main files.

The third step is the execution of the completed GPSS program by the standard GPSS system. This execution is described in the following section, 4.4.2 The Simulation Execution.

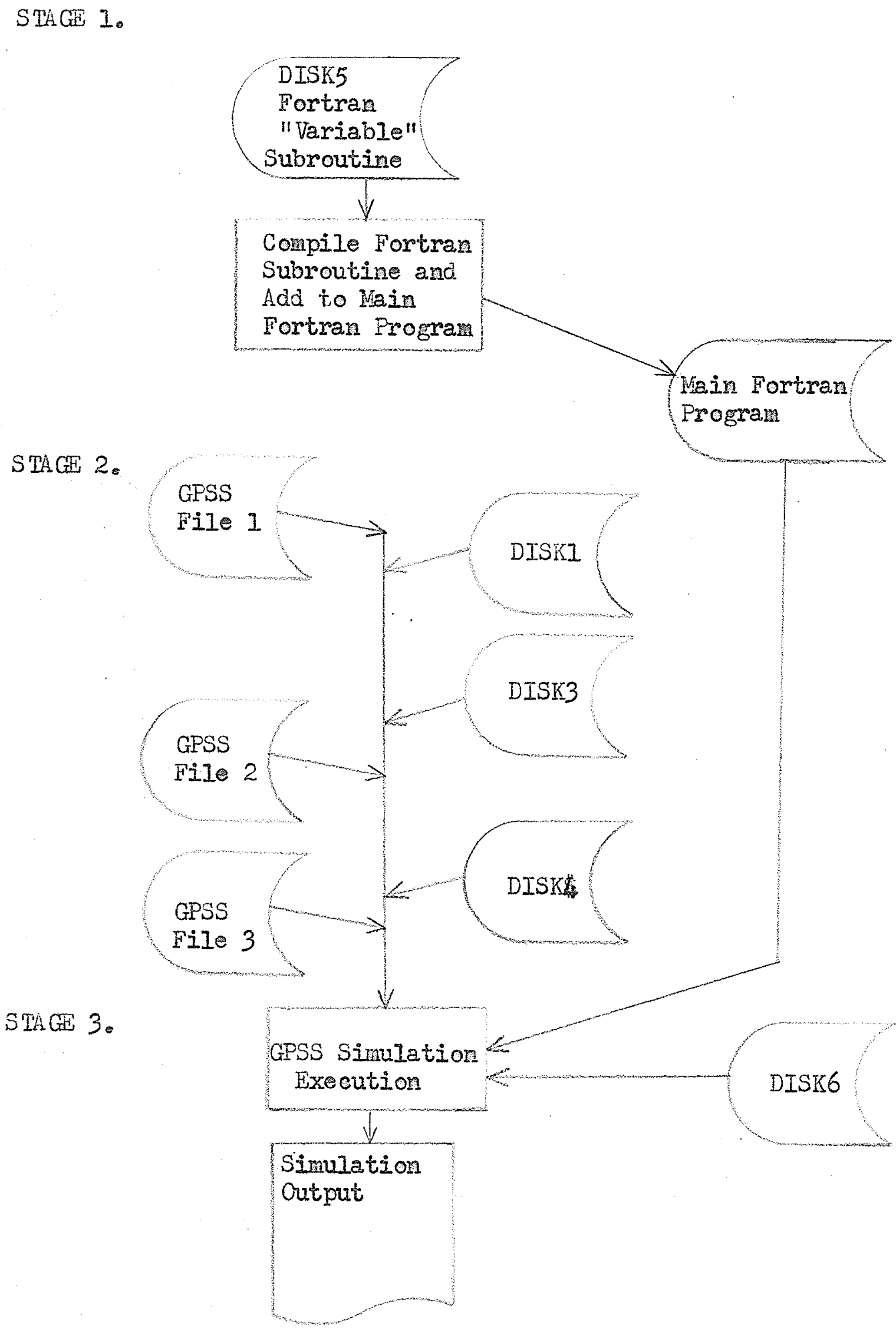
4.4.2 The Simulation Execution

The resultant GPSS program from step two in Section 4.4.1 becomes the main program of the simulation execution.

The first step of this program, before the simulation begins is the initialization of the storages, matrices and

Figure 4.7

Three Stages of the SIMULATION EXECUTION Program



vectors of the simulation. The matrices and vectors are set to zero automatically by GPSS. The disk file "DISK6" is read and the system's matrices and major vector initialized. The storages are then filled to the initial contents specified by the storage model definition statements.

The matrices and the major vector which were defined from the file "DISK6" contain the major definition of the model configuration and characteristics . The other files containing the Fortran and GPSS source code are essential but constitute a minor part of the main model.The matrices and the major vector are common to both the main Fortran and main GPSS programs. Figure 4.9 illustrates their use and characteristics. They are passed as arguments of the Fortran subroutine. The capabilities of the GPSS program to call the Fortran subroutine is achieved through the GPSS "HELP" block. Control is transferred to an Assembler routine. (Appendix F) which evaluates the addresses of the matrices and major vector and presents these to the main Fortran program in the form of a subroutine parameter list.

The main Fortran program consists of three subroutines, SIMT01, SIMT02,and SIMT03. The main Fortran processing is accomplished in SIMT01 which is the only subroutine which is called by the GPSS program. The subroutine SIMT02 performs the output of final simulation statistics and is called by SIMT01. The third subroutine, SIMT03, is created by the Edit program and is output as DISK5 (figure 4.7). This subroutine contains the Fortran mathematical assignment statements representing the variable model definition statements.

Transactions are generated in the standard GPSS manner by the GPSS source code contained in file DISK4 and concatenated into the main program. Control of the processing of these transactions is taken over by the main Fortran program which is treated as a subroutine called by the main GPSS program. Figure 4.8 illustrates the general organization and processing of the simulation execution.

4.5 Transaction Processing

Since transactions always originate at generate control blocks the corresponding control block is the transaction's first entity. Next, the next entity specification for that control block is evaluated. The transaction is assigned to this next entity and its type is determined. According to its type the corresponding matrix is searched to find its attributes. Once located the transaction is processed according to the nature of this entity, as defined by its attributes. The final stage of processing for a transaction in a given block or entity is the evaluation of the next entity specification. Once evaluated the cycle is repeated. Figure 4.10 illustrates this general transaction processing procedure. The processing for each type of entity and block is described in subsequent sections.

Figure 4.8
General Organization and Processing of Simulation Execution

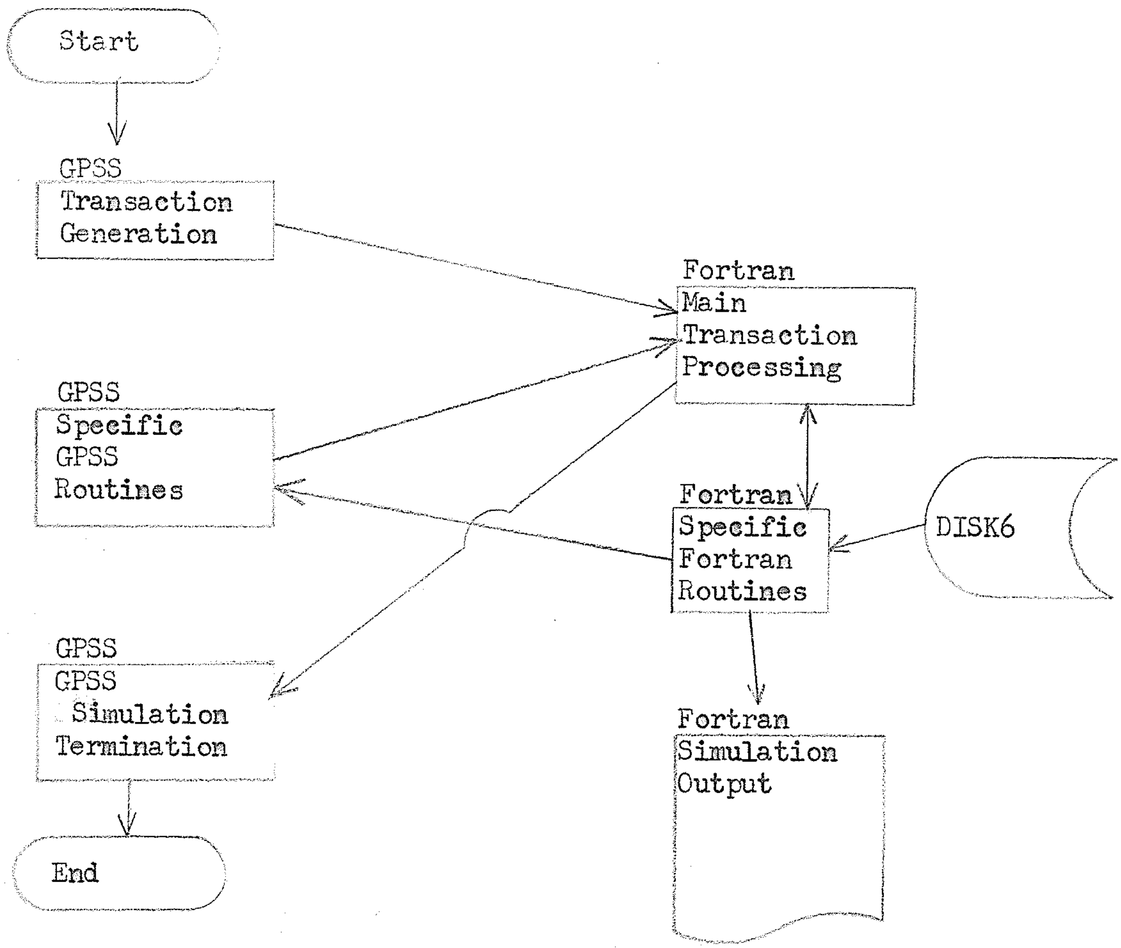


Figure 4.9

System Matrices and Vector.

GPSS name	Fortran name	Dimension	Use	Fortran Format
Matrix Savevalue 1	MX1	MX1(1,1)	not used	
Matrix Savevalue 2 "MSAVEVALUE 2" "MATRIX 2" "MX2"	MX2	MX2(50,15)	Facility definition	Row #= System Facility number col.1- facility model number cols 2 & 3- transaction processing time specification cols 4 & 5- next entity specification col.7- transaction entity count col.8- facility utilization col.9- average time per transaction col.10- indicator=1 after first transation has entered. cols 6,11 -15- unused.
Matrix savevalue 3 "MSAVEVALUE 3" MATRIX 3 "MX3"	MX3	MX3(50,20)	Storage definition	Row #= System Storage number col.1- storage model number col.2- storage capacity col.3- initial contents cols 4&5- storage addition specification cols 6&7- storage subtraction specification cols 8&9- next entity specification col.14- transaction entity count col.15- storage utilization col.16- average contents col.17- max. contents col.18- average time per transaction col.19- current storage contents. cols 10-13,20- unused
Matrix Savevalue 4 "MSAVEVALUE 4" "MATRIX 4" "MX4"	MX4	MX4(50,20)	Control Block definition	col.1- control block model number col.2- control block type cols 3-10 are dependent on the control block type See Chapter 2, sections2.12.1 to 2.12.14 attributes are stored in same order as they are defined in the model definition statements Note for Control block 12"Terminate" the total terminations at any time for this block is stored in col.7.

Figure 4.9 cont'd

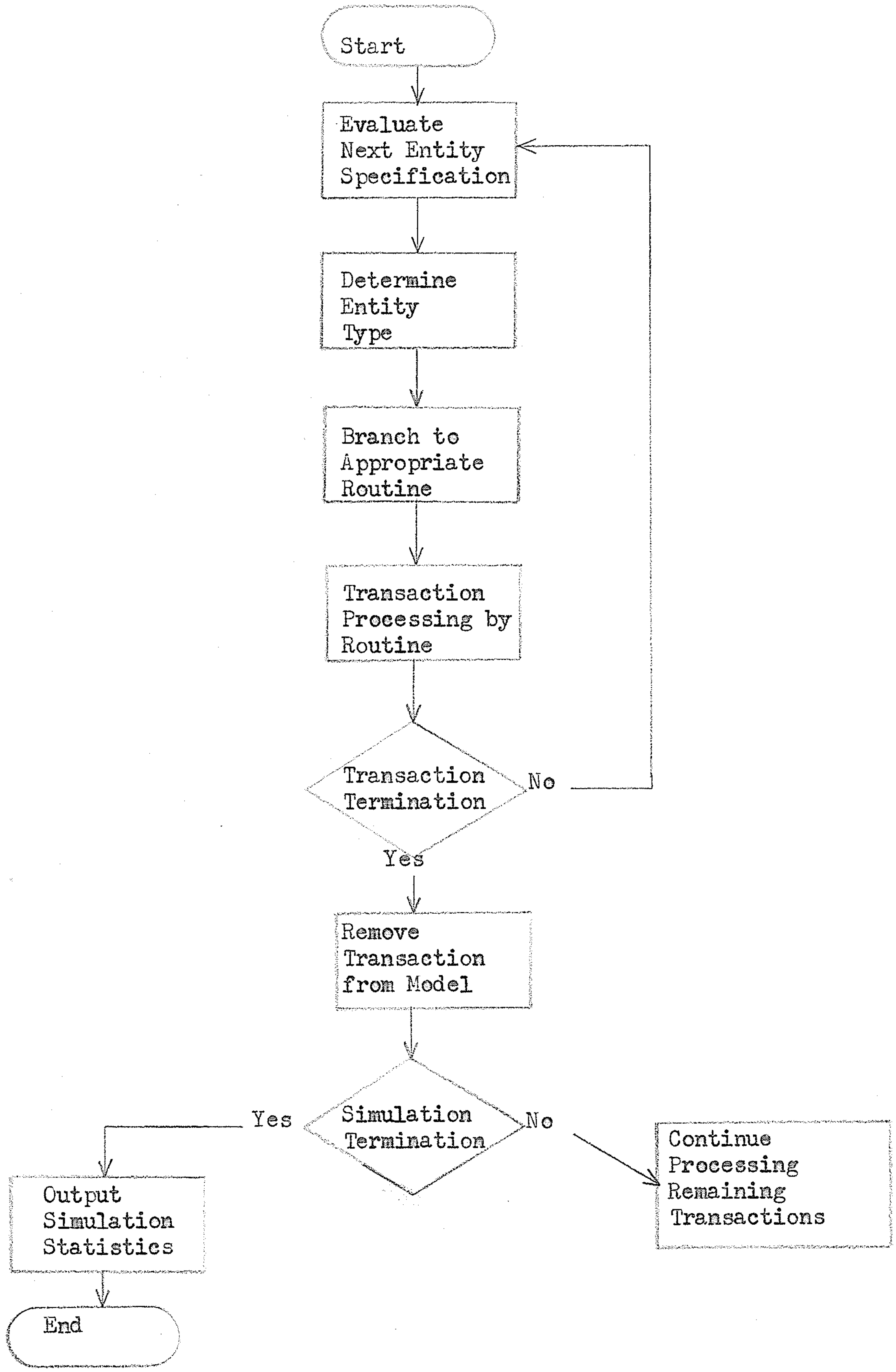
GPSS name	Fortran name	Dimension	Use	Fortran Format
Matrix Savevalue 5 "MSAVEVALUE 5" "MATRIX 5" "MX5"	MX5	MX5(50,10)	Queue definition	Row #= System Queue number col.1- model queue number cols 2&3- next entity specification col.4- unused col.5- transaction entity count col.6- average contents col.7- max. contents col.8- number of zero entries col.9- average time per transaction col.10- as for col.9 excluding zero entries
"Matrix Savevalue 6" "MSAVEVALUE 6" "MATRIX 6" "MX6"	MX6	MX6(20,20)	Decision rule Definition	col.1- entity number where decision rule is to be used. col.2- rule type col.3- rule number cols 4-20- next entity candidates
Matrix Savevalue 7 "MSAVEVALUE 7" "MATRIX 7" "MX7"	MX7	MX7(1,1)	Not used	
Matrix Savevalue 8 "MSAVEVALUE 8" "MATRIX 8" "MX8"	MX8	MX8(5,20)	Work matrix used in the analysis of decision rules	
Matrix Savevalue 9 "MSAVEVALUE 9" "MATRIX 9" "MX9"	MX9	MX9(1,1)	Not used	
Savevalues 1 to 100 Savevalues 1 to 20 are equated to the current transaction's 20 parameters.	A	A(100)	1. Represents transaction parameters 2. Represents user counters 1 to 25 3. Control of processing in Fortran and GPSS 4. Contains block and entity limits and totals	A(1) to A(5) (user parameters 1 to 5) are defined and referenced by the user. A(21) to A(45) are the user counters 1 to 25 A(6)- contains entity to be entered or just entered. A(7)- row of entity A(6) in corresponding matrix.

Figure 4.9 cont'd

A(8)-Fortran control number to determine to which Fortran routine control will be transferred when transaction enters Fortran from GPSS.
A(9)1.-When an error occurs in GPSS the error number is transferred to Fortran in this element.
2.-Also used when a transaction attempts to remove more storage units from a storage than exist-with the "LEAVE" control block.
A(10)-Used in ASSIGN,COUNTER and TEST blocks.
A(11)-contains the transaction number
A(12)-units to be removed from storage
A(13)-in the evaluation of an attribute specification A(13)contains the attribute type,A(16)contains the attribute number. When evaluated A(13) contains the result.
A(14)-Used in conjunction with the freeing of physical entities.
A(15)-Contains the last physical entity ie. the physical entity to be freed when a transaction arrives at another physical entity.
A(16)-See A(13)
A(17)-number of calling routine when last physical entity is freed.
A(18)-not used
A(19)-clock time when the second situation for A(9) occurs.
A(20)-not used
A(46-49)-not used
A(50-59)-parameters for the GPSS macro "FORT1"
A(60)-not used
A(61)-the number of the next GPSS routine when a transaction returns form Fortran
(62)-most negative queue number
(63)-most negative storage number
(64)-highest facility number
(65)-highest control block number
(66)-not used
(67)-number of facilities
(68)-number of storages
(69)-number of control blocks
(70)-number queues
(71)-number of decision rules
A(72-90)- not used
A(91)-work area to affect truncation of Function
A(93-95)-not used
A(96)= clock time for output with Print control block and at end of run.
A(97)- -1
A(98)-not used
A(99)-a work counter in GPSS
A(100)-not used
A(92)-number of transactions created

Figure 4.10

General Transaction Processing



4.6 Facility Processing

The facility attributes are stored in matrix "MX2". When a transaction's next entity is a facility, control is transferred to line number 395 of the main Fortran program (Appendix D). The processing of a transaction through a facility is executed in the following manner:

1. Locate the facility row in matrix MX2.
2. Place the row number in element seven of the major vector "A".
3. Initialize this facility by placing a "1" in the tenth column.
4. Transfer to GPSS routine GPR13.
5. Using the GPSS facility processing, "SEIZE" the facility of the same number as the row (element seven in vector "A").
6. If the facility is not available the transaction will remain at this point until it becomes available.
7. When the facility is available the transaction proceeds to free the last physical entity that the transaction entered.
8. The facility processing time for this transaction is evaluated by the GPSS macro "EVAL" and placed in element thirteen of vector "A".

9. The transaction enters the GPSS "ADVANCE" block which advances the simulation clock by the evaluated time thus simulating the passing of time.

10. When the facility processing has been simulated the facility number, element seven of vector "A" is placed in element fifteen of vector "A" thus defining the last physical entity number.

11. A zero is placed in parameter 14 (element fourteen of vector "A") indicating that the last physical entity was a facility.

12. Control is returned to line number 379 of the Fortran program.

13. A partial system check is made that the facility processing has proceeded correctly.

14. The next entity specification from columns four and five of the matrix MX2 is evaluated and this next entity is analysed for processing.

15. This transaction remains in the facility, preventing any other transactions from seizing it until this transaction enters a storage, or queue or seizes another facility. At that time the facility is freed for use by other transactions. The GPSS term freeing a facility is "RELEASE".

4.7 Storage Processing

The storage attributes are stored in matrix "MX3". When a transaction's next entity is a storage, control is transferred to line number 407 of the main Fortran program (Appendix D.)

The processing of a transaction through a storage is executed in the following manner:

1. Locate the storage row in matrix MX3.
2. Place the row number in element seven of the major vector "A".
3. Evaluate the number of units to add to the storage contents from the specification in columns four and five.
4. Printout the transaction number, the storage number and the number of units to be added.
5. Transfer to GPSS routine GPR12.
6. Using the GPSS storage processing "ENTER" the storage of the same number as the row in MX3 (element seven in vector "A") with the number of units evaluated in 5.
7. If this addition to the storage contents is greater than the available space the transaction will wait at this point until sufficient space becomes available.
8. When the storage has been successfully entered and the contents incremented the last physical entity is freed.

9. The entering transaction is now removed from the system.

The removal of units from a storage is accomplished by the "LEAVE" control block.

4.8 Queue Processing

The queue attributes are stored in matrix MX5. When a transaction's next entity is a queue, control is transferred to line number 425 in the main Fortran program (Appendix D).

The processing of a transaction through a queue is executed in the following manner:

1. Locate the queue row number in matrix MX5.
2. Place the row number in element seven of vector "A".
3. Transfer to GPSS routine GPR11.
4. Enter the queue of the same number as the row in MX5 (element seven of vector "A"),using the GPSS queue processing command "QUEUE".
5. Free the last physical entity which this transaction had entered.
6. A two is placed in parameter 14 (element fourteen of vector "A"). indicating that the last physical entity was a queue.
7. The system queue number contained in element seven of vector"A" is assigned to element fifteen to be used in the freeing process when this transaction enters another physical entity.

8. Transfer to the main Fortran program, line number 371 of Appendix D.
9. A partial system check is made that the queue processing has proceeded correctly.
10. The next entity specification from columns two and three of MX5 are evaluated and this next entity is analysed for processing.
11. This transaction remains in the queue, its presence indicated by a unit contribution to the present queue contents until this transaction enters another physical entity. At that time, through the freeing process the transaction will "DEPART: from this queue, the queue contents be decreased by one.

4.9 Control Block Processing

The control block attributes are stored in matrix MX4. When a transaction's next entity is a control block, control is transferred to line number 329 in the main Fortran program. (Appendix D.) The general processing of the control blocks is described below, followed by consideration of the specific operations of the individual control block types. The general processing of control blocks is accomplished in the following manner:

1. Locate the control block row in matrix MX4.
2. Place the row number in element seven of vector "A".

3. Look up the block type in column two of the corresponding row.
4. According to the block type transfer to the corresponding Fortran routine for the actual processing.

These four steps exist for all control blocks except the "generate" control block.

4.9.1 Generate Control Block

The generate control block originates transactions into a model. The operations of the generate block are executed in the following manner:

1. The GPSS statement"GENERATE" which was created in the Edit program, written into file DISK4, and concatenated with other files into a complete GPSS program, creates each transaction.
2. The user control block number is assigned to parameter 6 (element six of vector "A").
3. Control is transferred to Fortran line number 66.
4. The transaction is given a number (element eleven of vector "A").
5. The control block is located in matrix MX4.
6. The next entity specification as defined in columns eight and nine is evaluated and this next entity is analysed for processing.

4.9.2 Split Control Block

The operations of the split control block are executed as follows:

1. Transfer to GPSS routine GPR27.
2. The number of copies specified in column five of the control block matrix for this row are created in the standard GPSS manner.
3. The original transaction returns control to the Fortran main program.
4. The parent next entity specification in columns three and four are evaluated and the next entity analysed for processing.
5. When the parent transaction has either been removed from the model or has reached an entity or block where its continued procedding is delayed, the first copy is processed.
6. The copy transaction returns control to the main Fortran program.
7. The copy next entity specification in columns six and seven are evaluated and the next entity analysed for processing.
8. When each copy has either been removed or has reached an entity or block where its continued processing is delayed the next copy is processed by the steps 6, 7 and 8.

4.9.3 Assemble Control Block

The operation of the assemble control block are executed as follows:

1. Test if this is the <u>first</u>, <u>intermediate</u> or <u>last</u> transaction to enter the assemble block in a given group of transactions to be assembled.

First 2. The number of transactions be assembled as specified in columns three and four is evaluated, the result being placed in columns seven.

3. The total for this group in column seven is decreased by one.

4. Control is transferred to GPSS routine GPR19.

5. The last physical entity entered by this transaction is freed and the transaction removed from the model.

Intermediate 6. Steps 3, 4 and 5 are repeated.

Last 7. The total for this group of transactions is set to zero.

8. The next entity specification defined in columns five and six are evaluated and the next entity analysed for processing.

4.9.4 Facility Availability Control Block

The operations of the facility availability control block are executed as follows:

1. The facility number defined in column three of this control block is located in the facility matrix MX2.

2. If this facility has not been initialized element thirteen of vector "A" is set to zero. Next step is number 6.

3. If the facility has been initialized control is transferred to GPSS routine GPR28.
4. The facility availability is placed in element thirteen of vector "A". A zero indicates availability, a one indicates "in use".
5. Control is transferred to the main Fortran program.
6. The contents of element thirteen are placed in the counter whose number is defined in column four of the corresponding row of matrix MX4.
7. The next entity specification as defined in columns five and six is evaluated and the next entity analysed for processing.

4.9.5 Gather Control Blocks

The operations of the gather control block are executed as follows:

1. A test is made to determine if the present transaction is the first, an intermediate or the last transaction to enter relative to that group in the "gather" state.

First 2. The number of transactions to be gathered as specified in columns three and four are evaluated and placed in columns seven and eight.
3. The value in column seven is decreased by one.
4. Control is transferred to GPSS routine GPR17.
5. The transaction is held at a GPSS "GATE" block until the logic switch, of the same number as

the row in MX4, is put into a "set" or "on" position.

Intermediate 6. Steps 3, 4, and 5 are repeated.

Last 7. The value of column seven is assigned to minus one.

8. Control is transferred to GPSS routine GPR17.

9. The logic switch of the same number as the row in MX4 is "set or placed "on".

10. The first transaction moves through the logic switch. The next transaction will not move until the processing of the prior transaction is halted or delayed.

11. When the last transaction has left the logic switch, the switch is set "off".

4.9.6 Queue Contents

The operation of the queue contents control block is executed as follows:

1. The queue number defined in column three of this control block is located in matrix MX5. This row number becomes the system queue number.
2. If this queue has not been initialized element thirteen of vector "A" is set to zero. Next step in number 6.
3. If the queue has been initialized control is transferred to GPSS routine GPR29.
4. The contents of the queue are placed in element thirteen of vector "A".

5. Control is transferred to the main Fortran program.

6. The contents of element thirteen are placed in the counter whose number is defined in column four of the corresponding row of the matrix MX4.

7. The next entity specification as defined in columns five and six is evaluated and the next entity analysed for processing.

4.9.7 Assign Control Block

The operations of the assign control block are executed as follows:

1. The specification indicating the parameter number to be modified as defined in columns three and four is evaluated the parameter number being placed in element ten of vector "A".

2. The value of the modification as specified in columns six and seven is evaluated.

3. The type of modification, addition, subtraction or assignment, as defined in column five is analysed and the appropriate modification made to the parameter number in element ten of vector "A".

4. The next entity specification as defined in columns eight and nine is evaluated and the next entity analysed for processing.

4.9.8 Counter Control Block

The operations of the counter control block are executed as follows:

1. The specification indicating the counter number to be altered, as defined in columns three and four is evaluated and placed in element ten of vector "A".
2. The value of the modification as specified in columns six and seven is evaluated.
3. The type of modification, addition, subtraction or assignment, as defined in column five, is analysed and the appropriate modification made to the counter number specified in element ten.
4. The next entity specification as defined in columns eight and nine is evaluated and the next entity analysed for processing.

4.9.9 Test Control Block

The operations of the test control block are executed as follows:

1. The specification for the "A" argument, as defined in columns three and four is evaluated and placed in element ten of the vector "A".
2. The specification for the "B" argument, as defined in columns five and six is evaluated.
3. A comparison is made between arguments "A" and "B". If $A < B$ the next entity becomes the value of column

seven. If $A = B$ the value of column eight becomes the next entity and if $A > B$ column nine becomes the next entity. This next entity is then analysed for processing.

4.9.10 Leave Control Block

The operations of the leave control block are executed as follows:

1. The storage number specification as defined in columns three and four of MX4 in the row of the leave control block are evaluated.
2. This storage number is located in MX3. The row number being placed in element seven, the storage number being placed in elements thirteen and six of the vector "A".
3. Control is transferred to GPSS routine GPR20.
4. The last physical entity is freed.
5. Element fourteen is assigned the value one indicating that the last physical entity after this block will be a storage. Element fifteen is assigned the system storage number so that this storage will be freed when the next physical entity is entered.
6. Control is transferred to the Fortran main program.
7. Since element six of vector "A" now contains the storage number the transaction is processed as leaving the storage entity.

8. The next entity specification as defined in columns eight and nine of the storage matrix MX3 is evaluated and the next entity is analysed for processing.

4.9.11 Remove Control Block

The operations of the remove control block are executed as follows:

1. Control is transferred to GPSS routine GPR19.
2. The last physical entity is freed.
3. The transaction is removed from the system.

4.9.12 Terminate

The operations of the terminate control block are executed as follows:

1. The number of termination specification to end the simulation, as specified in columns three and four is evaluated.
2. The number of terminations as maintained in column seven is incremented by one.
3. A comparison is made to determine if the simulation should be terminated.
4. If not, control is transferred to GPSS routine GPR19 where the last physical entity is freed then the transaction removed from the system.
5. If the simulation is to be terminated control is transferred to GPSS routine GPR32.
6. The last physical entity is freed.

7. The relevant simulation statistics for each of the physical entities are collected and placed in the appropriate columns of the corresponding matrices.
8. Control is transferred to the main Fortran program.
9. Fortran subroutine SIMT03 is called to output the simulation statistics.
10. Control is transferred to GPSS routine GPR14 where the simulation is terminated.

4.9.13 Print Control Block

The operations of the print control block are executed as follows:

1. Control is transferred to GPSS routine GPR31.
2. The present value of the simulation clock is placed in element ninety-six of the vector "A".
3. Control is transferred to the main Fortran program.
4. The printout resulting lists the following quantities
   a) The five user parameters
   b) The twenty-five user counters
   c) The present clock time
5. The next entity specification as defined in columns three and four is evaluated and the next entity analysed for processing.

4.9.14 Storage Contents

The operation of the storage contents control block is executed as follows:

1. The storage number defined in column three of this control block is located in matrix MX3. The row number becomes the system storage number. This system storage number is placed in element thirteen of vector "A".
2. Control is transferred to GPSS routine GPR30.
3. The contents of the storage are placed in element thirteen of the vector "A".
4. Control is transferred to the main Fortran program.
5. The contents of element thirteen are placed in the counter whose number is defined in column four of the corresponding row of the matrix MX4.
6. The next entity specification as defined in columns five and six is evaluated and the next entity analysed for processing.

4.10 Model Attribute Evaluation

The evaluation of the model attribute specifications is accomplished by a separate Fortran routine within SIMT01. This routine may be referenced from any other routine within SIMT01. Figure 4.11 illustrates the context and general use of this routine.

Figure 4.11 Model Attribute Evaluation

The execution procedures for each of the model attribute,scalar, function, variable, parameter and counter are described below:

Attribute: Scalar

Specification: 1, all integers

Execution Procedure:

1. Place the value of the interger into element thirteen of the vector "A".
2. Return control to the calling routine.

Attribute Function

Specification: 2,1 to 20

Execution Procedure:

1. Place the number two in element thirteen of vector "A".
2. Place the number of the function to be referenced in element sixteen of vector "A".
3. Transfer control to GPSS routine GPR10.
4. Evaluate the function number contained in element sixteen of vector "A". Place the result in element thirteen.
5. Transfer control to the main Fortran program.
6. Return control to the calling routine.

Attribute: Variable

Specification: 3,1 to 20

Execution Procedure:

1. Call Fortran subroutine "SIMT02".

2. This subroutine affects the evaluation of all variable equations.

3. Place the value of the referenced variable in element thirteen of vector "A."

4. Return control to Fortran subroutine "SIMT01".

5. Return control to the calling Fortran routine.

Attribute: Parameter

Specification: 4,1 to 5

Execution Procedure:

1. Place the value of the parameter element of vector "A",whose number is in element sixteen of vector "A", in element thirteen of vextor"A".

2. Return control to the calling routine.

Attribute: Counter

Specification: 5,1 to 25

Execution Procedure:

1. Add twenty to the contents of element sixteen in vector"A".

2. Place the contents of the counter,(element in vector "A"), whose number is in element sixteen of vector "A", into element thirteen of vector"A".

3. Return, control to the calling routine.

4.11 Freeing of Physical Entities

Sections 4.6, 4.7 and 4.8 described the detailed aspects of the particular physical entities. In each case reference was made to the concept of freeing the last physical entity. When a transaction enters a physical entity it is restricted, by the definition of this language, from, simultaneously occupying any other physical entity.Therefore when a physical entity is entered the transaction must leave any other physical entity. A special GPSS routine performs the freeing process. If the last entity was a facility it is released to enable awaiting transactions to occupy it. If the last entity was a storage the specified number of units are subtracted from the storage contents. If the last entity was a queue the queue contents are decreased by one.

4.12 Next Entity Decision Rule Processing

The next entity specification, as well as being an evaluation of a model attribute, can be a decision rule. The decision rules are applied when the next entity candidates are all queues, storages or facilities. When this situation exists the next entity decision can be made according to a comparison of the present status or past performance of these candidates. If the candidates are facilities the decision may be based on present availability or the relative utilizations of the candidates. For storages the decisions may be based on utilization, present contents, or average contents. The queue attributes of present and average contents are available for analysis. The general operations of the decision rule processing are as follow:

1. Evaluate decision rule type i.e.facility, storage or queue.
2. Place the user entity numbers in row one of the work matrix MX8.
3. Place the system entity numbers into row three as defined from an analysis of nx2 for facility candidates, nx3 for storage candidates or MX5 for queue candidates.
4. Transfer control to the appropriate GPSS routine for obtaining the current values of the quantities on which the comparison will be base.
5. Place the current value of these quantities in the respective columns of row two.
6. Return to Fortran where the values of row two are evaluated for the minimum or maximum quantity.
7. Isolate the user entity number corresponding to the desired optimum value.
8. Place this value in element six of the vector "A" and process this entity as the next entity for this transaction.

4.13 Error Processing

The analysis, detection and display of errors exists in both the Edit program and the Simulation program. Appendix A contains the error messages for the Edit program and the form of the error message, while Appendix B illustrates the simulation execution errors and a sample error message. When an error is detected in either program a characteristic error number is printed with relevant error parameters. The purpose of the error parameters is to assist the user in locating the exact cause of the error.

In the Edit program the errors usually reflect either an invalid specification of a model definition statement or an error related to the numbering of the physical entities, control blocks, functions or variables. In the former case the error parameter is the column number in the user's model definition statement and, in the latter case, the error parameter is the invalid entity or block number or a number relating to the error situation.

In the Simulation program errors are detected for the following general causes:

1. An entity or block number referenced as a next entity has not been defined by a model definition statement.
2. An invalid model attribute i.e. parameter,function, variable, or counter has been referenced.

The error parameter displayed with the error number are composed of two lines each containing fifteen values. The corresponding error message in Appendix C identifies the relevant parameter values.

The detection of an error by the Edit program terminates the processing of that line even though other statements may be defined on the same line. Subsequent lines containing one or more statements will be processed normally. The detection of an error in the Simulation program terminates the simulation execution. The occurance of an error in the Edit program prohibits the execution of the Simulation program.

4.14 Output Processing

The standard output of a completed simulation is presented in Figure 3.20.The first page(s) are output by the Edit program. Subsequent pages contain the Simulation program output. Apart from the output initiated by the print control block or by the occurance of a simulation error the standard simulation execution output consists of:

1. Trace statements indicating the flow of each transaction through the model,
2. Facility simulation statistics,
3. Storage simulation statistics,
4. Queue simulation statistics,
5. A printout of the standard print control block printout.
6. The control block at which termination occurred.
7. The clock time at simulation termination.

4.15 Language Expansion

Several provisions exist in the Edit and Simulation programs which can be used directly to incorporate additional language developemnt. Section 4.3 described the general form of the Edit program. The general flow of this program is reviewed below:

1. Input one eighty column line.
2. Isolate model definition statement identifying character(s).
3. Transfer to appropriate processing routine.
4. Process statement creating
   i) GPSS or Fortran source code
   ii) additional elements of a matrix
   iii) both i and ii.
5. Output files containing source code and matrices.

To add addition language concepts to the Edit program the following general procedure should be followed:

1. Corresponding to step 2 above add PL-1 statements to test for the new statements' identifying character(s) and transfer control to a unique statement label at the beginning of the new processing routine.
2. Write PL-1 source code for new language statement.

   Note: Additional GPSS or Fortran source code can be written into unused file DISK2.

   Note: Additional matrix elements can be written into unused matrix MX1.

3. The output of source code into file DISK2 and the output of new matrix elements into DISK6 can be achieved in a manner similiar to the present method.

Once the additional language concepts have been added into the Edit program they must be provided for in the Simulation program.

The concatenation of two files using the Job Control Language is a simple matter. The required statements are illustrated in Appendix H.

Note:

The GPSS representation of a two-dimensional matrix is the opposite tothe representation in Fortran. Thus the number of Fortran rows equals the number of GPSS columns and the number of Fortran columns equals the number of GPSS rows. Similarily in referencing as element in GPSS the row and column indices are reversed from these in Fortran.

The Fortran and/or GPSS system coding to process the new addition may be inserted within the general flow of the Simulation program as illustrated in Figure 4.10.

In particular the flowchart blocks "Determine Entity Type", "Branch to Appropriate Routine" and "Transaction is Processed by Routine" will be changed to affect the processing of the new addition(s). In addition to the above expansion considerations the use of the vector"A" should be accomplished with little difficulty in a manner similiar to the present. Several elements in this vector are unused.

In the event that the language was to achieve a reasonable acceptance as a teaching or elementary application language the addition of more user parameters and counters are suggested. In addition the GPSS concepts of the logic switch and match blocks might also be intelligent additions to expand the language.

4.16 Operational Considerations

The system was developed and tested on an I B M system 360 model 65J operating under MUT with releases 18 and 19 of the full Operating System. The system uses Pl-1, Fortran G, Assemble F and GPSS 360. Implementation of the system for maximum effeciency should incorporate the following concepts:

1. Cataloguing of the Edit program.
2. Cataloguing of the Assemble routine and Fortran subroutine SIMT01 and SIMT02 .
3. Compilation of the Edit created subroutine SIMT03 and subsequent addition to the Assembler and Fortran module.

4. Cataloguing of the system's Job Control Language statements.

Representative core requirements are one hundred and thirty "K" for the Edit program and between two and three hundred "K" for the Simulation execution program.

The Edit program execution normally requires approximately five seconds for 100 model definition statements. The simulation execution time is dependent on the number of transactions to be generated. The execution time required for the examples in Chapter 3 averaged 45 seconds.

4.17 Importance of GPSS

The importance of GPSS in the developement and operation of this system is obvious from the contents of Chapter 4 and Appendix E. Several of the model definition statement concepts are based on similiar concepts in GPSS. In some cases the execution of a control block is actually accomplished by an equivalent GPSS command. A partial list of these functions which are performed by the GPSS system included: creation of transactions, transaction control, facility service time simulation, facility output statistics, queue output statistics, storage output statistics, simulation clock processing, and execution termination.

The extensive use of the concepts and methods of the GPSS system simplify the transition from this elementary system to the GPSS language. Students familiar with this elementary simulation system will find several similarities in the GPSS language. In addition the comprehension of those features unique to GPSS will be facilitated by a knowledge of the elementary simulation system.

CHAPTER V

Conclusions

5.1 Review of Objectives

The elementary simulation system described in previous chapters was designed to accomplish the following objectives:

1. to illustrate the fundemental concepts of transaction oriented simulation thus assisting in the instruction of these concepts,
2. to require a minimum prior computer programming knowledge for the comprehension and use of this system, and
3. to facilitate the translation of non-complex models into computer executable form.

The degree to which this system assists in the instruction of simulation concepts cannot be conclusively evaluated until such time as it is used in an education environment. The second objective's fulfilment might also be measured more effectively when the system achieves a reasonable degree of use. However, when compared with other simulation languages such as GPSS, SIMSCRIPT or FORTRAN it must be conceded that its syntax and grammer are less complicated if only for the reason that it is a smaller language. (reference Section 1.2)

The effectiveness of this system in the expression of non-complex models in a computer executable form, the third objective, can be partially established by the examples in chapter III and further by a relative comparison of the other simulation languages.

In summary, although it would appear that this elementary simulation system is a usefull tool for the simulation of non-complex models, future use and acceptance will be necessary for its final exaluation.

5.2 Future Use and Expansion

The main uses of this system would appear to be in the instruction of simulation concepts, and the simulation of non-complex business and industrial models. Courses in operations research business management and engineering design, by using this system could illustrate the application of simulation as a decision making tool in the evaluation of alternate systems, policies or physical configurations. The system has been designed to allow expansion as it becomes necessary. (Section 4.15).

From an operational viewpoint , the computer cost of running small models, compares favourably with other simulation languages. Section 4.16 presents recommendations to allow effecient operation of the system.

5.3 Limitations

Inspite of the fact that this language has not had widespread use certain limitations are apparent. The first of these is the use of numeric codes to represent model attributes. It is suggested that mnemonics beused as an alternative. In this manner the specification of a parameter, for example, might be indicated by a "P" followed by the parameter number. The letters "V" for the variable, "F" for the function, "C" for the counter and "K" for a scalar constant are the extension of this suggestion.

A second feature which is suggested for this system is the ability to surpress the large amount of tracing output. This could be accomplished by having two versions of this system; one with the trace feature; one without it. Alternately an additional model definition statement called "TRACE" might be used to initiate the trace output.

A third limitation is found in the output statistics. In particular, the addition of the statistics, the variance for the frequency distribution of the contents for each queue is an important value in the evaluation of a model's performance.

APPENDIX A

ERRORS DETECTED

in

EDIT PROGRAM

Model Definition Statement Errors

Errors Detected in EDIT Program

Error Number	Explanation
	General Errors:
1.	invalid placement of numeric, period or sign character.
2.	invalid statement definition character.
3.	invalid character, character not included in character set.
	Function Statement Errors:
4.	alphabetic character, indicating a new model definition statement, occurred before specified number of function points were found.
5.	invalid function number; cannot be greater than 20.
6.	invalid function argument type, statement parameter two must be one of 1, 2, 3, or 4.
7.	for function argument (second statement parameter) equal to 1 (parameter) range of argument number (third statement parameter) is 1, 2, 3, 4 or 5.

8. for function argument (second statement parameter) equal to 2 (randon number), range of argument number (third statement parameter) is from 1 to 8 inclusive.

9. for function argument (second statement parameter) equal to 3 (counter), range of argument number (third statement parameter) is from 1 to 25 inclusive.

10. invalid function type (forth statement parameter),must be 1, continuous, or 2, discrete.

11. more function points have been found than the number specified in the fifth statement parameter.

Storage Statement Errors:

12. insufficient number of storage arguments specified, there must be nine.

13. more storage arguments specified than the required nine.

14. invalid placement of a negative storage argument, these are allowed for arguments one and nine.

15. correct number of storage arguments (nine) not specified.

16. maximum of fifty storages exceeded.

17. storage numbers overlap queue numbers

Control Block Statement Errors:

20. insufficient number of control block arguments for this control block type.

21. more arguments specified than required for this control block type.

22. invalid control block type.

23. invalid generation control block argument; argument three, "mean transaction inter-arrival time " type must be 1 (constant), 2 (function) or 3 (variable).

24. maximum of fifty control blocks exceeded.

25. generation control block statement contains an undefined variable.

26. generation control block statement contains an undefined function.

27. invalid negative argument.

28. generation control block interarrival modifier argument five cannot be larger than the mean interarrival time as specified by arguments three and four.

Queue Statement Errors:

30. maximum of fifty queues exceeded.

31. incorrect number of arguments specified, must be three.

32. invalid negative argument.

Facility Statement Errors:

40. maximum of fifty facilities exceeded.

41. insufficient number of arguments specified must be five.

42. more arguments specified than the required five.

43. invalid negative argument.

44. facility entity numbers overlap control block numbers.

Variable Statement Errors:

60. invalid alphabetic character.

61. redefinition of variable is invalid,variable number has previously been defined.

62. maximum of twenty variables exceeded.

63. variable expression must be preceeded by an equal sign (" = ").

64. variable expression is too long,segment by defining two or more variables.

65. unmatched brackets detected in variable expression.

66. variable expression ending with an operator is invalid.

67. invalid one alpha character operand.

68. invalid two alpha character special function.

69. invalid three alpha character special function.

70. invalid four alpha character special function.

71. invalid five alpha character special function.

72. invalid alphabetic operand or special function.

73. invalid placement of special function.

74. special function argument must be enclosed in brackets.

75. invalid placement of operand.

76. alphabetic operand must be followed immediately by operand number.

77. invalid counter number.

78. invalid parameter number.

79. invalid placement of numeric constant.

80. invalid placement of decimal point.

81. invalid placement of operator.

82. invalid operator.

83. invalid placement of open bracket.

84. invalid placement of closed bracket.

85. not all referenced variables have been defined.

86. variable numbers must be numbered sequentially and inclusively from one.

Decision Statement Errors:

90. maximum of fifty decision rules exceeded.

91. maximum number of arguments (fifty) has been exceeded.

92. invalid decision rule must be 6, 7 or 8.

93. invalid decision rule type for decision rule 6. Decision rule 6 has types 1, 2 or 3.

94. invalid decision rule type for decision rule 7. Decision rule 7 has types 1, 2, 3 or 4.

95. invalid decision rule type for decision rule 8. Decision rule 8 has types 1, or 2.

APPENDIX B

ERRORS DETECTED in SIMULATION EXECUTION

SIMULATION EXECUTION ERRORS

Error Number	Explanation
1.	- a transaction attempted to enter a GENERATE control block. Transactions are generated by this control block but cannot enter it. The first number found in the first line and the "ERROR PARAMETER" is the number of the missused GENERATE block.
2.	- the facility to be examined by the facility availability control block, given as the first number in the first line of the "ERROR PARAMETER", has not been defined.
3.	- the counter referenced in the queue contents, storage contents or facility availability control block, given by the first number in the first line of the "ERROR PARAMETER", is outside the range 1 to 25.
4.	- the queue to be examined by the queue contents control block, given as the first number in the first line of the "ERROR PARAMETERS", has not been defined.
5.	- the storage to be examined by the storage contents control block, given as the first number in the first line of the"ERROR PARAMETERS", has not been defined.

6. - the storage referenced by the leave control block given as the first number in the first line of " ERROR PARAMETERS" has not been defined.

7. - the entity number found in the first position in the first line of the " ERROR PARAMETERS" is not a defined control block.

8. - the facility referenced as the first number in the first line of the " ERROR PARAMETERS" is not defined.

9. - the storage referenced as the first number in the first line of the " ERROR PARAMETERS" is not defined.

10. - the queue referenced as the first number in the first line of the" ERROR PARAMETERS" is not defined.

11. - in the evaluation of a model attribute for the entity number found in the first position of the first line of the " ERROR PARAMETERS", the attribute type is invalid.

12. - the entity number found in the first position in the first line of the" ERROR PARAMETERS" references an undefined variable.

13. - the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a parameter outside the range 1 to 5.

14. - the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a counter outside the range 1 to 25.

15. - the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a next entity decision rule which has not been defined.

16. - the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a next entity decision rule which contains no next entity candidates.

17. - the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a next entity decision rule which contains more than one type of next entity candidate. Next entity candidates must be all storages or queues or facilities or control blocks, not a combination.

18. -the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a next entity decision rule. One of the next entity candidates for this rule, given by the eighteenth number in the first line of the "ERROR PARAMETERS", is an undefined facility.

19. -as for error number 18. The undefined next entity candidate is a storage.

20. -as for error number 18. The undefined next entity candidate is a queue.

21. -the entity number found in the first position in the first line of the "ERROR PARAMETERS" references a next entity decision rule. The specification of this decision rule contains a correct rule type but an invalid rule number for this type.

SIMULATION EXECUTION SYSTEM ERRORS

The elementary simulation system contains thirteen system checks. The purpose of these system checks is to insure the correct use and implementation of the simulation system.

If a user finds a system error it should be brought to the attention of those responsible for the implementation and maintenance of this system.

The following error explanation will assist the system's programmer in identifying the error.

ERROR NUMBER	EXPLANATION
70. (Fortran statement 72)	A generate control block has been defined in GPSS but does not exist in the control block matrix MX4. Possible sources of error are: the EDIT program, output of files, handling of the generate control block model definition statement, the EXECUTION program , the input of the matrix definition file.(DISK1)
71. (Fortran statement 349)	The next entity evaluation for a generate control block, which should have been accomplished in statements 64 to 82 of the Execution program (Fortran) has been bypassed or the element A(7) which identifies the row in MX4 of the present control block has been altered

during control processing in either the GPSS or the Fortran sections of the Execution program. The correct control block number is that given by the first number in the first line of the "ERROR PARAMETERS".

72. (Fortran statement 354) The next entity evaluation for a split control block, which should have been accomplished in statements 85 to 101 of the Execution program (Fortran) has been bypassed. The error has the same cause as error number 71.

73. (Fortran statement 356 or 370) The next entity evaluation for the test control block, which should have been accomplished in statements 180 to 201 of the Execution program(Fortran) has been bypassed. The error has the same causes as error 71.

74. (Fortran statement 375) In the processing of the queue given by the first number in the first row of the "ERROR PARAMETERS" the queue row number in matrix MX5 given by A(7) has been altered. See error 71 for further explanation.

75. (Fortran statement 383)	Similiar error for storage processing as error 74.
76. (Fortran statement 391)	Similiar error for facility processing as error 74.
77. (Fortran statement 479)	After the evaluation of a model attribute type and number an attempt has been made to transfer control to a part of the program unknown to this routine. Check the use of A (20), which identifies the calling routine.
78. Fortran statement 568)	In the evaluation of the optinum next entity candidate in a decision rule the element A (13) which identifies the rule type, has been incorrectly altered.
79. (Fortran statement 589)	The error has occurred in the evaluation and decision of the next entity candidate of a decision rule.
80. (GPSS statement 334)	In the transfering of control from the Fortran section of the Execution program to the GPSS section, the identity of the GPSS routine given by A (61), has either been altered or the called routine does not exist.

81. (GPSS statement 376) The error has occurred in the freeing of a physical entity. The value of element 14, which identifies the last physical entity processed, has been altered.

81. (Fortran statement 38) An attempt has been made to create matrix MX1. In the current version MX1 is unused. The matrix definition file (DISK1) which was created in the EDIT program contains invalid data. If the system has been changed to incorporate new language features it is likely that the new EDIT program and this version of the Execution program are incompatible.

APPENDIX C

MODEL DEFINITION REFERENCE CARD

MODEL ATTRIBUTE SUMMARY

Name	Type	Range of Numbers
Scalar	1	all integers
Function	2	1 to 20
Parameter	3	1 to 5
Variable	4	1 to 20
Counter	5	1 to 25

FACILITY: F

no.	processing time attrib. type	processing time attrib. no.	next entity attrib. type	next entity attrib. no.
n_1	n_2	n_3	n_4	n_5

STOPAGE: S

no.	capacity	initial contents	units entering attrib. type	units entering attrib. no.	units leaving attrib. type	units leaving attrib. no.	next entity attrib. type	next entity attrib. no.
n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9

QUEUE : Q

no.	next entity attrib. type	next entity attrib. no.
n_1	n_2	n_3

CONTROL BLOCK SUMMARY

name	1.	2	3	4	5	6	7	8	9
GENERATE	no.	1	interarrival mean type	no.	modifier	offset	transaction count	next type	entity no.
SPLIT	no.	2	parent next entity type	no.	copies	copy next type	entity no.		
ASSEMBLE	no.	3	assemble type	count no.	next type	entity no.			
FACILITY UTILIZATION	no.	4	fac. no.	counter no.	next type	entity no.			
GATHER	no.	5	gather type	count no.	next type	entity no.			
QUEUE CONTENTS	no.	6	que no.	counter no.	next type	entity no.			
ASSIGN	no.	7	parameter identity type	no.	+ 1 = 0 - -1	modifier type	identity no.	next type	entity no.
COUNTER	no.	8	counter identity type	no.	+ 1 = 0 - -1	modifier type	identity no.	next type	entity no.
TEST	no.	9	"A"argument type	no.	"B"argument type	no.	A < B entity	A = B entity	A > B entity
LEAVE	no.	10	storage identity type	no.					
REMOVE	no.	11							
TERMINATE	no.	12	termination count type	no.					
PRINT	no.	13							
STORAGE CONTENTS	no.	14	stor no.	counter no.	next type	entity no.			

VARIABLE: V_i=expression

NOTES:

1."i" is the variable no. i≤20.

2."expression" is a mathematical expression containing no embedded blanks.

3."expression"= operand operator operand
where:
"operand"=expression,
=special function,
=mnemonic,
=constant.
"operator" =* multiplication,
=** power,
=/ division,
=+ addition,
=- subtraction.

4."special function"= name(operand)
where:
"name"=LN,EXP,LOG,SIN,TAN,COS,SQRT,
=SINH,COSH,ATAN,ARSIN,ARCOS,
=COTAN.

5

5."mnemonic"=V_i , i=any defined variable,
=C_i , i=1,2,...25 counter no.,
=P_i , i=1,2,...5 parameter no..

6."(" and ")" may surround any operand.

FUNCTION: FN FU	no.	x-axis argument attrib. type	x-axis argument attrib. no.	function type	no. of x,y points to be defined	$x_1,y_1/x_2,y_2/\ldots x_k,y_k$
	n_1	n_2	n_3	n_4	k	← k points →

NOTES:

1. $n_1 \leq 20$.
2. $n_2=1$ implies parameter; n_3=parameter no. ($1\leq n_3\leq 5$).
3. $n_2=2$ implies random no.; n_3=generator seed ($1\leq n_3\leq 8$).
4. $n_2=3$ implies counter; n_3=counter no. ($1\leq n_3\leq 25$).
5. $n_2=4$ implies clock time; n_3=dummy argument ($0\leq n_3\leq 1000$)

DECISION RULE: D	entity no. using this rule	rule type	rule no.	first candidate	. . .	last candidate (k'th)
	n_1	n_2	n_3	n_4		n_{k+3}

NOTES:

1.

n_2	n_3	explanation
6 (facilities)	1	first available
	2	lowest utilized
	3	highest utilized
7 (storages)	1	lowest utilized
	2	highest utilized
	3	least full
	4	most full
8 (queues)	1	shortest
	2	longest

2. k<47.

APPENDIX D

FORTRAN PROGRAM

```
                  SUBROUTINE SIMT01(A,MX1,MX2,MX3,MX4,MX5,MX6,MX7,MX8,MX9)
                  IMPLICIT INTEGER (A-Z)
            C  DISK FILE PESIM20.A3540,UM1404
            C     FORTRAN SECTION  1 INITIALIZE MATRICES AND TABLES
                  DIMENSION A(100),MX1(1,1),MX2(50,15),MX3(50,20),MX4(50,20),
                 1MX5(50,10),MX6(20,20),MX7(1,1),MX8(5,20),MT(50),ERROR(100)
                 2,MX9(1,1)
            C
                  DO 8887 I=1,100
             8887 ERROR(I)=0
            C     READ IN MODEL DEFINITION CARDS
             9988 CONTINUE
            C
                  IPR=6
                  GO TO 9000
            C
            C A(8) MUST BE SET TO 17 FOR FORTRAN INITIALIZATION STEP TO BE EXECUTED
             9001 M1=0
                  M2=0
                  M3=0
                  M4=0
                  M5=0
                  M6=0
                  M7=0
                  M8=0
                  K=0
                  IDK=9
             8888 FORMAT('  IDK IS EQUAL TO ', I5)
               98 READ(IDK,99,END=9999) I,J,(MT(N),N=1,J)
               99 FORMAT(52I4)
            C   NEW MATRIX IS ASSUMMED WHEN ROW = I IS EQUAL TO 1
                  GO TO (9202,102,402,502,602,702,802,900),I
            C     VECTOR  "A(100)"
             9202 DO 101 M=1,J
              101 A(M)=MT(M)
            C CONTROL VECTOR
                  A(61)=2
                  A(62)=MT(1)
                  A(63)=MT(2)
                  A(64)=MT(3)
                  A(65)=MT(4)
                  A(66)=MT(5)
                  A(67)=MT(6)
                  A(68)=MT(7)
                  A(69)=MT(8)
                  A(70)=MT(9)
                  A(71)=MT(10)
                  A(72)=MT(11)
                  GO TO 98
              102 ERROR(81)=1
                  GO TO 1000
            C                 MATRIX  2
              402 M2=M2+1
                  DO 9203 M=1,J
             9203 MX2(M2,M)=MT(M)
                  GO TO 98
            C                 MATRIX  3
              502 M3=M3+1
```

```
0045              DO 401 M=1,J
0046          401 MX3(M3,M)=MT(M)
0047              GO TO 98
            C              MATRIX  4
0048          602 M4=M4+1
0049              DO 504 M=1,J
0050          504 MX4(M4,M)=MT(M)
0051              GO TO 98
            C              MATRIX  5
0052          702 M5=M5+1
0053              DO 601 M=1,J
0054          601 MX5(M5,M)=MT(M)
0055              GO TO 98
            C              MATRIX  6
0056          802 M6=M6+1
0057              DO 701 M=1,J
0058          701 MX6(M6,M)=MT(M)
0059              GO TO 98
            C              MATRIX  7
0060          900 M7=M7+1
0061              DO 801 M=1,J
0062          801 MX7(M7,M)=MT(M)
0063              GO TO 98
            C         ENTER GPSS AND PROCESS ROUTINE GPR19
            C
            C     INSERT ALL FORTRAN PROCESSING
            C
            C
            C     SECTION 10  GENERATE PROCESSING
            C
            C ALLTRANS COMING FROM GENERATE BLOCKS WILL GO TO THIS SECTION
            C
            C  FIND ROW OF GEN BLK IN MX4
0064         9199 ENT=A(6)
0065              A(92)=A(92)+1
0066              A(11)=A(92)
0067              WRITE(IPR,4299) A(11),A(6)
0068              IRX=A(69)
0069              DO 9300 IR=1,IRX
0070              IF(ENT.EQ.MX4(IR,1))GO TO 9301
0071         9300 CONTINUE
0072              ERROR(70)=1
0073              GO TO 1000
0074        9301  A(20)=15
0075              IA=MX4(IR,8)
0076              IB=MX4(IR,9)
            C DEFINE PARAMETERS
0077              A(12)=-1
0078              A(14)=-1
0079              A(15)=-1
0080              GO TO 316
0081        9304  A(6)=A(13)
0082              GO TO 200
            C END OF SECTION 10..............................
            C234567890123456789012345678901234567890123456789012345678901234567890
            C     FORTRAN SECTION.8
            C
            C         CONTROL BLOCK PROCESSING
```

```
                 C
                 C      CONTROL BLOCKS ARE:-
                 C
                 C    CONTROL BLOCK PROCESSING
                 C
                 C   CB(1)"GENERATE"
                 C             THIS PROCESSING HAS BEEN ACCOMPLISHED IN SIMTAB EDIT 01
                 C   10   GENERATE BLOCK CANNOT BE ENTERED BY TRANSACTION
 0083              10 ERROR(1)=1
 0084                 GO TO 1000
                 C   CB(2) "SPLIT"
                 C   METHOD:- MX4(I,5)=# TO BE SPLIT;SPLIT TRANS(GPSS)
 0085            11      A(61)=27
 0086                    A(8)=0
 0087                    I=A(7)
 0088                    GO TO 9999
                 C PARENT RETURN
 0089            1025  I=A(7)
 0090                  IA=MX4(I,3)
 0091                  IB=MX4(I,4)
 0092                  A(20)=2
 0093                  GO TO 316
                 C  OFFSPRING  RETURN
 0094            1026  I=A(7)
 0095                  IA=MX4(I,6)
 0096                  IB=MX4(I,7)
 0097                  A(20)=2
 0098                  GO TO 316
                 C  A(20)=2 RETURNS TO 1027......
 0099             1027 WRITE(IPR,4299) A(11),A(6)
 0100                  A(6)=A(13)
 0101                  GO TO 200
                 C  CB(3) "ASSEMBLE-A"
 0102               12 IF(MX4(I,7)-1)1100,1125,1150
                 C     EVALUATE ATTRIBUTES
 0103             1100 A(20)=4
 0104                  IA=MX4(I,3)
 0105                  IB=MX4(I,4)
 0106                  GO TO 316
 0107             1101 MX4(I,7)=A(13)-1
                 C
                 C     GO TO 'X' TERMINATE CODE "19"
                 C
 0108                  A(61)=19
 0109                  GO TO 9999
                 C
                 C     LAST 'X' TO BE ASSEMBLEC ; GO TO SECTION 3
                 C
 0110             1125 MX4(I,7)=-1
 0111                  GO TO 300
                 C
                 C     INTERMEDIATE 'X'
                 C
 0112             1150 MX4(I,7)=MX4(I,7)-1
 0113                  A(61)=19
 0114                  GO TO 9999
                 C  CB(5) "GATHER"
                 C #990
```

```
0115          14 IF(MX4(I,7)-1)1400,1425,1450
            C    EVALUATE ATTRIBUTE
0116        1400 IA=MX4(I,3)
0117             IB=MX4(I,4)
0118             A(20)=3
0119             GO TO 316
0120        1401 MX4(I,7)=A(13)-1
0121             MX4(I,8)=A(13)
0122             A(61)=17
0123             A(8)=3
0124             A(6)=MX4(I,1)
0125             A(7)=I
0126             GO TO 9999
            C    LAST 'X'
0127        1425 I=A(7)
0128             MX4(I,7)=-1
0129             A(61)=17
0130             A(8)=3
0131             GO TO 9999
            C    INTERMEDIATE 'X'
0132        1450 I=A(7)
0133             MX4(I,7)=MX4(I,7)-1
0134             A(8)=3
0135             A(61)=17
0136             GO TO 9999
            C
            C  CB(7) "ASSIGN"
0137          16 A(20)=5
0138             IA=MX4(I,3)
0139             IB=MX4(I,4)
0140             GO TO 316
0141        1600 N=A(13)
0142             A(10)=N
0143             IF(N-5)1603,1603,1699
0144        1603 A(20)=6
0145             IA=MX4(I,6)
0146             IB=MX4(I,7)
0147             GO TO 316
0148        1601 N=A(10)
0149             IF(MX4(I,5))1602,1610,1620
0150        1602 A(N)=A(N)-A(13)
0151             GO TO 300
            C #1500
0152        1610 A(N)=A(13)
0153             GO TO 300
0154        1620 A(N)=A(N)+A(13)
0155             GO TO 300
0156        1699 ERROR(13)=1
0157             GO TO 1000
            C
            C  CB(8) "COUNTER"
0158          17 A(20)=7
0159             IA=MX4(I,3)
0160             IB=MX4(I,4)
0161             GO TO 316
0162        1700 N=A(13)
0163             N=N+20
0164             A(10)=N
```

```
0165              IF(N-45)1701,1701,1799
0166         1701 A(20)=8
0167              IA=MX4(I,6)
0168              IB=MX4(I,7)
0169              GO TO 316
0170         1702 N=A(10)
0171              IF(MX4(I,5))1703,1710,1720
0172         1703 A(N)=A(N)-A(13)
0173              GO TO 300
0174         1710 A(N)=A(13)
0175              GO TO 300
0176         1720 A(N)=A(N)+A(13)
0177              GO TO 300
0178         1799 ERROR(14)=1
0179              GO TO 1000
            C
            C  CB(9) "TEST"
            C
0180           18 A(20)=9
0181              IA=MX4(I,3)
0182               IB=MX4(I,4)
0183              GO TO 316
0184         1800 A(10)=A(13)
0185              IA=MX4(I,5)
0186              IB=MX4(I,6)
0187              A(20)=10
0188              GO TO 316
0189         1801 N2=A(13)
0190              N=A(10)
0191              IF(N-N2)1802,1803,1804
            C A<B 1802
0192        1802  NENT=MX4(I,7)
0193              GO TO 1805
            C A=B 1803
0194        1803  NENT=MX4(I,8)
0195              GO TO 1805
            C A>B 1804
0196        1804  NENT=MX4(I,9)
0197              GO TO 1805
0198        1805  LENT=A(6)
0199              WRITE(IPR,4299) A(11),A(6)
0200          308 A(6)=NENT
0201              GO TO 200
            C
            C  CB(12) "TERMINATE"
            C
0202           20 A(20)=11
0203              WRITE(IPR,4298) A(11),A(6)
0204         4298 FORMAT(' TRANSACTION #',I5,' IS TERMINATED AT ENTITY #',I5,'.')
0205              IA=MX4(I,3)
0206              IB=MX4(I,4)
0207              GO TO 316
0208         2100 N1=A(13)
0209              A(20)=12
0210         2101 MX4(I,7)=MX4(I,7)+1
0211              IF(N1.LE.MX4(I,7))GO TC 2102
0212              A(61)=19
0213              A(8)=0
```

```
            GO TO 9999
      2102  A(8)=0
            A(95)=MX4(I,7)
            A(94)=MX4(I,1)
            A(61)=14
            GO TO 2200
C     GET FACILITY AVAILABILITY  CB(4)
        13  A(61)=28
            A(8)=15
            A(7)=I
            FAC=MX4(I,3)
            IH=A(67)
            DO 8254 I =1,IH
            IF(MX2(I,1).EQ.FAC)GO TO 8255
      8254  CONTINUE
            ERROR(2)=1
            GO TO 1000
      8255  A(13)=I
            IF(MX2(I,10).NE.0)GO TO 9999
            A(13)=0
      3050  I=A(7)
            IX=MX4(I,4)+20
            IF(IX.GT.19.AND.IX.LT.46)GO TO 3057
      3058  ERROR(3)=1
            GO TO 1000
      3057  A(IX)=A(13)
            GO TO 300
C     GET QUEUE CONTENTS  CB(6)
        15  A(61)=29
            A(8)=11
            A(7)=I
            QUE=MX4(I,3)
            IQ=A(70)
            DO 8252 I=1,IQ
            IF(MX5(I,1).EQ.QUE)GO TO 8253
      8252  CONTINUE
            ERROR(4)=1
            GO TO 1000
      8253  A(13)=I
            IF(MX5(I,10).NE.0)GO TO 9999
            A(13)=0
      2000  I=A(7)
            IX=MX4(I,4)+20
            IF(IX.LT.20.OR.IX.GT.45)GO TO 3058
            A(IX)=A(13)
            GO TO 300
C     GET STORAGE CONTENTS  CB(10)
        23  A(61)=30
            A(8)=12
            A(7)=I
            STR=MX4(I,3)
            IS=A(68)
            DO 8250 I=1,IS
            IF(MX3(I,1).EQ.STR)GO TO 8251
      8250  CONTINUE
            ERROR(5)=1
            GO TO 1000
      8251  A(13)=I
```

```
0269               GO TO 9999
0270         2001  I=A(7)
0271               IX=MX4(I,4)+20
0272               IF(IX.LT.20.OR.IX.GT.45)GO TO 1000
0273               A(IX)=A(13)
0274               GO TO 300
             C  REMOVE TRANSACTION  CB(11)
0275            21 A(61)=19
0276               WRITE(IPR,4296) A(11),A(6)
0277          4296 FORMAT(' TRANSACTION #',I5,' IS REMOVED BY ENTITY #',I5,'.')
0278               GO TO 9999
             C  PRINT BLOCK  CB(13)
0279           22  A(61)=31
0280               A(8)=16
0281               GO TO 9999
             C NOTE A(13) CONTAINS CLOCK TIME FOR OUTPUT
0282         3070  WRITE(IPR,3069) (I,A(I),I=1,5),(J,A(J+20),J=1,25),A(96)
0283         3069  FORMAT('0PARAMETER VALUES:1 TO 5 '/5(1X,'P',I1,'=',I8,';')/
                  1' COUNTER VALUES:1 TO 25 '/2(10(1X,'C',I2,'=',I7,';')/),
                  2(5(1X,'C',I2,'=',I7,';')/),' CLOCK TIME =',I5,' UNITS.')
0284               GO TO 300
             C  LEAVE   CB(10)    METHOD...
             C1. EVAL STR #; 2. FREE PREV PHYS ENT; 3.SET FREE PARMS; 4.GO TO NENT;
0285            19 A(20)=13
0286               IA=MX4(I,3)
0287               IB=MX4(I,4)
0288               A(7)=I
0289               GO TO 316
0290          2770 A(61)=20
0291               ENT=A(13)
0292               IS=A(68)
0293               DO 2771 I=1,IS
0294               IF(ENT.EQ.MX3(I,1))GO TO 2772
0295         2771  CONTINUE
0296               ERROR(6)=1
0297               GO TO 1000
0298         2772  A(13)=ENT
0299               WRITE(IPR,4299) A(11),A(6)
0300               A(6)=ENT
0301               A(7)=I
0302               A(8)=3
0303               GO TO 9999
             C     ATTEMPT TO 'LEAVE' STR BY MORE THAN CONTENTS
0304         2783  A(61)=21
0305               IS=A(15)
0306               JS=MX3(IS,1)
0307               IPR=6
0308               WRITE(IPR,2784) A(12),JS,A(13),A(19)
0309         2784  FORMAT('0 **** A TRANSACTION ATTEMPTED TO REMOVE ',I6,
                  1' UNITS FROM STORAGE #',I5,'.'/'       THE STORAGE CONTAINED '
                  2,I6,' UNITS. THE CLOCK TIME IS ',I5,' UNITS.'/
                  3'       THE STORAGE IS NOW EMPTY.')
0310               A(8)=A(9)
0311               GO TO 9999
             C
             C   FORTRAN SECTION 9      ERROR PROCESSING
             C
0312         1000  DO 1001 I=1,100
```

```
                IF(ERROR(I))1002,1001,1002
          1002  WRITE(IPR,1003) I,(A(J),J=6,20),(A(K),K=61,75)
          1003  FORMAT(' SIMULATION ERROR NUMBER ',I4,20X,'ERROR PARAMETERS'/
               1' LINE 1: ',15I6/' LINE 2: ',15I6)
                ERROR(I)=0
          1001  CONTINUE
                GO TO 2200
          1004  I=A(9)
                ERROR(I)=1
                GO TO 1000
          C     FORTRAN SECTION  2
          C              TRANSFER CONTROL TO CORRECT FORTRAN ROUTINE
          C  A(8)=17 CORRESPONDS TO THE INITIALIZATION STEP
          9000  IBR=A(8)+1
                GO TO (1009,1101,1009,300,1401,1600,1601,1700,1702,1800,1801,
               12000,2001,2100,2101,3050,3070,9001,9199,4065,499
               2,2200,2201,1025,1026,1004,4422,2783), IBR
          1009  ERROR(22)=1
                GO TO 1000
          C     FORTRAN SECTION  4
          C              IDENTIFY NEXT BLOCK TYPE AND DIRECT PROCESSING
          C     A(62)--- QUEUES---- A(63)----STORAGES--- 0--- FACILITIES-A(64)---
          C----- CONTROL BLOCKS
           200  ENT=A(6)
                IF( ENT.LT.A(63))GO TO 201
                IF(ENT.LT.0)GO TO 225
                IF(ENT.LE.A(64))GO TO 250
          C     ENT IS A CONTROL BLOCK
                IRX=A(69)
                DO 4 I=1,IRX
                IF(MX4(I,1).EQ.ENT)GO TO 5
             4  CONTINUE
                ERROR(7)=1
                GO TO 1000
          C     FIND BLOCK TYPE AND GO TO PROPER ROUTINE
             5  J=MX4(I,2)
                A(7)=I
                GO TO (10,11,12,13,14,15,16,17,18,19,21,20,22,23),J
          C     FORTRAN SECTION 3
          C              CHOOSE NEXT ENTITY
          C     IDENTIFY LAST ENTITY
           300  LENT=A(6)
                WRITE(IPR,4299) A(11),A(6)
          4299  FORMAT(' TRANSACTION #',I5,' IS LEAVING ENTITY #',I5,'.')
                IF(LENT.LT.A(63))GO TO 301
                IF(LENT.LT.0)GO TO 321
                IF(LENT.LE.A(64))GO TO 341
          C
          C     ENTITY WAS A CONTROL BLOCK / CHOOSE NEXT ENTITY
          C
          C   DEFINE NEXT ENTITY DECISION PARAMETERS FOR EVALUATION
                A(20)=1
                I=A(7)
                J=MX4(I,2)
                GO TO(3100,3101,3102,3102,3102,3102,3103,3103,3104,3102,
               13106,3106,3108,3102),J
          3100  ERROR(71)=1
                GO TO 1000
```

```
      3102 IA=MX4(I,5)
           IB=MX4(I,6)
           GO TO 316
C 3101 SPLIT BLOCK WAS DONE IN SPLIT CONTROL BLOCK PROCESSING
      3101 ERROR(72)=1
           GO TO 1000
C 3104 TEST DONE IN CBLK PROCESSING
      3104 ERROR(73)=1
           GO TO 1000
      3103 IA=MX4(I,8)
           IB=MX4(I,9)
           GO TO 316
      3105 IA=MX4(I,7)
           IB=MX4(I,8)
           GO TO 316
      3107 IA=MX4(I,6)
           IB=MX4(I,7)
           GO TO 316
      3108 IA=MX4(I,3)
           IB=MX4(I,4)
           GO TO 316
      3106 ERROR(73)=1
           GO TO 1000
C
C          ENTITY WAS A QUEUE -FIND ROW IN MX5
C
       301 IQ= A(7)
           A(20)=1
           IF(LENT.EQ.MX5(IQ,1))GC TO 303
           ERROR(74)=1
           GO TO 1000
       303 IA=MX5(IQ,2)
           IB=MX5(IQ,3)
           GO TO 316
C
C          ENTITY WAS A STORAGE-FIND ROW IN MX3
C
       321 IS=A(7)
           A(20)=1
           IF(MX3(IS,1).EQ.LENT)GO TO 323
           ERROR(75)=1
           GO TO 1000
       323 IA=MX3(IS,8)
           IB=MX3(IS,9)
           GO TO 316
C
C          ENTITY WAS A FACILITY-FIND ROW IN MX2
C
       341 IF=A(7)
           A(20)=1
           IF(MX2(IF,1).EQ.LENT)GO TO 343
           ERROR(76)=1
           GO TO 1000
       343 IA=MX2(IF,4)
           IB=MX2(IF,5)
           GO TO 316
C
C          FORTRAN SECTICN 5   I.E.NENT IS A FACILITY....
```

```
      C
      C         FACILITY PROCESSING
      C
      C         PROCESSING PRIOR TO ENTRY TO SECTION 5 HAS BEEN
      C              1.SECTION  3.-CHOOSE NEXT ENTITY
      C              2.SECTION  4.-DIRECT FLOW TO ENTITY PROCESSING
      C
      C               ENTRY TO GPSS ROUTINE GPR13, FACILITY PROCESSING IN GPSS
      C               ,REQUIRES  P6= NEXT FACILITY #,
      C                          P7= ROW # OF FACILITY IN MX2,
      C                          P11= ROW + COL. IN MX1
      C
        250 A(61)=13
            A(6)=ENT
            A(8)=3
            IF=A(67)
            DO 251 I=1,IF
            IF(ENT.EQ.MX2(I,1))GO TO 252
      251   CONTINUE
            ERROR(8)=1
            GO TO 1000
      252   A(7)=I
            MX2(I,10)=1
            GO TO 9999
      C
      C     FORTRAN SECTION 6
      C
      C         STORAGE PROCESSING
        225 A(61)=12
            A(6)=ENT
            A(8)=3
            DO 253 I=1,50
            IF(ENT.EQ.MX3(I,1))GO TO 254
      253   CONTINUE
            ERROR(9)=1
            GO TO 1000
      254   A(7)=I
      C EVAL UNIT ENTERING STORAGE
            A(20)=14
            IA=MX3(I,4)
            IB=MX3(I,5)
            GO TO 316
      2781  A(61)=12
            WRITE(IPR,4297) A(11),A(6),A(13)
       4297 FORMAT(' TRANSACTION #',I5,' ENTERING STORAGE #',I6,' WITH',
           1I8,' UNITS.')
            A(8)=3
            GO TO 9999
      C     FORTRAN SECTION 7
      C
      C         QUEUE PROCESSING
      C
      C
        201 A(61)=11
            A(6)=ENT
            A(8)=3
            DO 256 I=1,50
            IF(ENT.EQ.MX5(I,1))GO TO 257
```

```
 256  CONTINUE
      ERROR(10)=1
      GO TO 1000
 257  A(7)=I
      MX5(I,10)=1
      GO TO 9999
 9950 FORMAT(' MATRIX',I4/5(15I4/))
 9999 RETURN
C     THIS CODING PERFORMS THE FOLLOWING FUNCTIONS:
C   IN GENERAL EVALUATES VALUES GIVEN THE VALUE TYPE AND I.D.
C   THEN ,DEPENDING ON THE VALUE OF A(20) CONTROL IS RETURNED
C   TO THE ORIGINAL ROUTINE.A(20)=1 ==- GO TO 500.
C     EVALUATE DECISION TYPE AND VALUE
C       IA=1     VAL=IB
C      IA=2      VAL=FN(IB)
C      IA=3      VAL=V(IB)
C      IA=4      VAL=P(IB)
C      IA=5      VAL=C(IB)
C      IA=6      FACILITY DECISION RULE
C      IA=7      STORAGE DECISION RULE
C      IA=8      QUEUE DECISION RULE
 316  GO TO (4001,4002,4003,4004,4005,4006,4007,4008,4099),IA
 4099 ERROR(11)=1
      GO TO 1000
 4001 A(13)=IB
      GO TO 499
 4002 A(13)=IA
      A(16)=IB
      A(61)=10
      A(8)=26
      GO TO 9999
 4003  A(16)=IB
      A(13)=99193
      CALL  SIMT09(A)
      IF(A(13).NE.99193)GO TO 4031
      ERROR(12)=1
      GO TO 1000
 4031 GO TO 499
 4020 GO TO 499
 4004 IF(IB.LE.5.AND.IB.GT.0)GO TO 4021
      ERROR(13)=1
      GO TO 1000
 4021 A(13)=A(IB)
      GO TO 499
 4005 IF(IB.LE.25.AND.IB.GE.0)GO TO 4022
      ERROR(14)=1
      GO TO 1000
 4022 A(13)=A(IB+20)
      GO TO 499
C   FACULTY DECISION RULES.....ALL ENTITIES ARE BETWEEN 0 & A(64)
 4006  T1=0
      T2=A(64)+1
      GO TO 4090
C  STORAGE DECISION RULES.....ALL ENTITIES ARE BETWEEN A(63)-1 & 0
 4007 T1=A(63)-1
      T2=0
      GO TO 4090
C   QUEUE DECISION RULES.....ALL ENTITIES ARE BETWEEN A(62)-1 & A(63)
```

```
0473      4008  T1=A(62)-1
0474            T2=A(63)
0475            GO TO 4090
0476      4422  VAL=A(13)
0477      499   IT=A(20)
0478            GO TO(509,1027,1401,1101,1600,1601,1700,1702,1800,1801,
               12100,2101,2770,2781,9304),IT
0479            ERROR(77)=1
0480            GO TO 1000
0481      509   JR=IR
0482            NENT=A(13)
0483      507   A(6)=NENT
0484            GO TO 200
          C  BUILD WORK TABLE OF NEXT ENTITY CANDIDATES
0485      4090  N=A(11)
0486            LENT=A(6)
          C DECISION RULE:MX6:COL1=LENT;COL2=RULE TYPE;COL3=RULE I.D.
0487       4097 DR=A(71)
0488            DO 4091 I=1,DR
0489            IF(LENT.NE.MX6(I,1))GO TO 4091
0490            IF(IA.NE.MX6(I,2))GO TO 4091
0491            IF(IB.EQ.MX6(I,3))GO TO 4092
0492       4091 CONTINUE
0493            ERROR(15)=1
0494            GO TO 1000
          C PUT ENTITIES INTO MX8
0495       4092 DO 4093 J=4,20
0496            IF(MX6(I,J).EQ.0)GO TO 4094
0497            MX8(1,J-3)=MX6(I,J)
0498       4093 CONTINUE
0499       4094 IF(J.GT.4)GO TO 4096
0500            ERROR(16)=1
0501            GO TO 1000
0502       4096 KT=J-4
0503            GO TO 4063
          C 4063   TEST THAT WORK TABLE VALUES AGREE WITH DECISION TYPE
0504      4063  DO 4064 I=1,KT
0505            ENT=MX8(1,I)
0506            IF(ENT.GT.T1.AND.ENT.LT.T2)GO TO 4064
0507            ERROR(17)=1
0508            GO TO 1000
0509      4064  CONTINUE
          C SEND TO GPSS FOR EVALUATION
0510            IP=IA-5
0511            GO TO (4110,4111,4124),IP
          C  PUT SYSTEM FAC # INTO MX8(3,I)
0512      4110  IH=A(67)
0513            DO 4250 I=1,KT
0514            ENT=MX8(1,I)
0515            DO 4251 J=1,IH
0516            IF(ENT.EQ.MX2(J,1))GO TO 4252
0517      4251  CONTINUE
0518            A(18)=ENT
0519            ERROR(18)=1
0520            GO TO 1000
0521      4252  MX8(3,I)=J
0522      4250  CONTINUE
0523            GO TO (4120,4121,4121,4130),IB
```

```
C  PUT SYSTEM STR # INTO MX8(3,I)
4111  IS=A(68)
      DO 4253 I=1,KT
      ENT=MX8(1,I)
      DO 4254 J=1,IS
      IF(ENT.EQ.MX3(J,1))GO TC 4255
4254  CONTINUE
      A(18)=ENT
      ERROR(19)=1
      GO TO 1000
4255  MX8(3,I)=J
4253  CONTINUE
      GO TO (4122,4122,4123,4123,4123,4130),IB
4130  ERROR(21)=1
      GO TO 1000
C FACILITY AVAILABILITY
4120  A(61)=22
      GO TO 4119
C   FACILITY UTILIZATICN
4121  A(61)=23
      GO TO 4119
C  STORAGE UTILIZATICN
4122  A(61)=24
      GO TO 4119
C  STORAGE CONTENTS
4123  A(61)=25
      GO TO 4119
C QUEUE CCNTENTS
C  PUT SYSTEM QUEUE # INTO MX8(3,I)
4124  IQ=A(70)
      DO 4256 I=1,KT
      ENT=MX8(1,I)
      DO 4257 J=1,IQ
      IF(ENT.EQ.MX5(J,1))GO TC 4258
4257  CONTINUE
      A(18)=ENT
      ERROR(20)=1
      GO TO 1000
4258  MX8(3,I)=J
4256  CONTINUE
      A(61)=26
      GO TO 4119
4119  A(8)=19
      A(13)=IA
      A(16)=IB
      A(19)=KT
      GO TO 9999
C RETURN FROM GPSS...MX8 NOW CONTAINS VALUES FOR COMPARISON
4065  IP=A(13)-5
      IB=A(16)
      KT=A(19)
      GO TO(4066,4067,4C68,4069),IP
4069  ERROR(78)=1
      GO TO 1000
C FACILITY COMPARISONS
4066  GO TO (4070,4071,4072,4073),IB
4C73  ERROR(21)=1
      GO TO 1000
```

```
      C CHOOSE FIRST AVAIL. FAC I.E. =  ZERO
      4070  DO 4078 I=1,KT
            IF(MX8(2,I).EQ.0)GO TO 4079
      4078  CONTINUE
      4080  I=1
      4079  NENT=MX8(1,I)
            A(6)=NENT
            IC=MX8(3,I)
            GO TO 200
      C LOWEST UTILIZED FACILITY
      4071  MIN=MX8(2,1)
            KM=1
            DO 4081 I=1,KT
            IF(MIN.LT.MX8(2,I))GO TO 4081
            MIN=MX8(2,I)
            KM=I
      4081  CONTINUE
            IF(KM)4082,4082,4083
      4082  ERROR(79)=1
            GO TO 1000
      4083  I=KM
            GO TO 4079
      C HIGHEST UTILIZED FACILITY
      4072  MAX=MX8(2,1)
            KM=1
            DO 4084 I=1,KT
            IF(MAX.GT.MX8(2,I))GO TO 4084
            MAX=MX8(2,I)
            KM=I
      4084  CONTINUE
            IF(KM)4082,4082,4083
      C...........END OF FACILITY EVALUATIONS.............
      C     STORAGE EVALUATIONS
      4067  GO TO(4101,4102,4101,4102,4103),IB
      4103  ERROR(21)=1
            GO TO 1000
      C LEAST FULL STORAGE
      C LEAST UTILIZED
      4101  GO TO 4071
      C   MOST FULL STORAGE
      C   HIGHEST UTILIZATION
      4102  GO TO 4072
      C...........END OF STORAGE EVALUATIONS...............
      C      QUEUE EVALUATIONS
      4068  GO TO (4104,4105,4106),IB
      C  SHORTEST QUEUE
      4104  GO TO 4071
      C  LONGEST QUEUE
      4105  GO TO 4072
      4106  ERROR(21)=1
            GO TO 1000
      C...........END OF QUEUE EVALUATIONS
      C   END OF SIMULATION,OUTPUT SIM STATS,ETC......
      2200  X=X
      C  ENTER GPSS TO OBTAIN SIMULATION STATISTICS
            A(61)=32
            A(8)=22
            GO TO 9999
```

```
 2201 CALL SIMT03(A,MX2,MX3,MX5)
      WRITE(IPR,2245)
 2245 FORMAT('1')
      WRITE(IPR,3069) (I,A(I),I=1,5),(J,A(J+20),J=1,25),A(96)
      WRITE(IPR,2246) A(95),A(94)
 2246 FORMAT('0THE SIMULATION HAS BEEN TERMINATED BY',I6,
     1' TERMINATIONS FOR CONTROL BLOCK #',I4,'.')
C     GO TO GPSS FOR FINAL TERMINATION OF RUN
      A(61)=14
      GO TO 9999
      END
```

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	6C0	SIMT09	6C4	SIMT03	6C8				

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	6E0	IPR	6E4	M1	6E8	M2	6EC	M3	6F0
M4	6F4	M5	6F8	M6	6FC	M7	700	M8	7C4
K	708	IDK	70C	J	710	N	714	M	718
ENT	71C	IRX	720	IR	724	IA	728	IB	72C
N2	730	NENT	734	LENT	738	N1	73C	FAC	740
IH	744	IX	748	QUE	74C	IQ	750	STR	754
IS	758	JS	75C	IBR	760	IF	764	T1	768
T2	76C	VAL	770	IT	774	JR	778	DR	77C
KT	780	IP	784	IC	788	MIN	78C	KM	790
MAX	794	X	798						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	79C	MX1	7A0	MX2	7A4	MX3	7A8	MX4	7AC
MX5	7B0	MX6	7B4	MX7	7B8	MX8	7BC	MT	7C0
ERROR	888	MX9	A18						

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
8888	A1C	99	A34	4298	A3A	4296	A6F	3069	AA1
2784	B33	1003	BE6	4299	C39	4297	C68	9950	CA5
2245	CBB	2246	CC0						

```
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,MAP
*OPTIONS IN EFFECT*  NAME = SIMT01  , LINECNT =       60
*STATISTICS*    SOURCE STATEMENTS =      623,PROGRAM SIZE =     13102
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

```
0001          SUBROUTINE SIMT03(A,MX2,MX3,MX5)
0002          IMPLICIT INTEGER(A)
CCC3          DIMENSION A(100),MX2(50,15),MX3(50,2C),MX5(50,10),X(50,2)
0004          IPR=6
        C ROUTINE TO OUTPUT FACILITY,STORAGE & QUEUE STATISTICS.
0005          IH=A(67)
CCC6          IF(IH.EQ.0)GO TO 4
0007          DO 10 I=1,IH
C0C8          X(I,1)=MX2(I,8)
0009       10 X(I,1)=X(I,1)/1CCC.
0010          WRITE(IPR,100) (MX2(I,1),MX2(I,7),X(I,1),MX2(I,9),I=1,IH)
C011      100 FORMAT('1',50X,'*** FACILITY STATISTICS ***'//5X,'FACILITY NO.',
             15X,'TRANSACTION ENTRIES',5X,'AVERAGE UTILIZATICN',5X,
             2'AVERAGE TIME/TRANSACTION'//(6X,I8,14X,I8,15X,F5.3,18X,I8))
0012        4 IS=A(68)
0013          IF(IS.EQ.0)GO TO 5
C014          DO 11 I=1,IS
0015          X(I,2)=MX3(I,15)
0016       11 X(I,2)=X(I,2)/1CCC.
0017          WRITE(IPR,200) (MX3(I,1),MX3(I,14),X(I,2),MX3(I,16),MX3(I,18),
             1MX3(I,17),MX3(I,2),MX3(I,19),I=1,IS)
0018      200 FORMAT('1',50X,'*** STORAGE STATISTICS ***'//
             15X,'STORAGE NO.',6X,'TOT. UNITS ',6X,'AVERAGE',6X,'AVERAGE',
             26X,'AVERAGE TIME/',6X,'MAXIMUM',6X,'CAPACITY',7X,'CURRENT',/24X,
             3'ENTERED'
             4,6X,'UTILIZATICN',4X,'CCNTENTS',5X,'TRANSACTION',8X,'CONTENTS',
             520X,'CONTENTS'//
             6(5X,I8,9X,I8,10X,F5.3,4X,I8,10X,I8,8X,I8,7X,I8,7X,I8))
0019        5 IQ=A(70)
0020          IF(IQ.EQ.0)GO TO 99
0021          WRITE(IPR,300) (MX5(I,1),MX5(I,5),MX5(I,6),MX5(I,7),MX5(I,8),
             1MX5(I,9),MX5(I,10),I=1,IQ)
0022      300 FORMAT('1',50X,'*** QUEUE STATISTICS ***'//
             17X,'QUEUE NO.',7X,'TRANSACTION',7X,'AVERAGE',7X,'MAXIMUM',7X,
             2'NUMBER OF',7X,'AVERAGE TIME/',7X,'AVE.TIME/TRANS'/
             326X,'ENTRIES',7X,'CONTENTS',7X,'CONTENTS',5X,'ZERO ENTRIES',
             45X,'TRANSACTICN',7X,'(NC ZERO ENTRIES)'//
             5(7X,I8,9X,I8,7X,I8,7X,I8,8X,I8,11X,I8,9X,I8))
0023       99 RETURN
0024          END
```

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	C8								

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IPR	CC	IH	D0	I	D4	IS	D8	IQ	DC

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	E0	MX2	E4	MX3	E8	MX5	EC	X	F0

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
100	280	200	316	300	416				

```
*OPTIONS IN EFFECT*  ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,MAP
*OPTIONS IN EFFECT*  NAME = SIMT03  , LINECNT =       60
*STATISTICS*    SOURCE STATEMENTS =        24,PROGRAM SIZE =     2224
*STATISTICS*  NO DIAGNOSTICS GENERATED
```

APPENDIX E

MAIN GPSS PROGRAM

```
REALLOCATE BLO,600,FAC,50,STO,50,QUE,50,LOG,60,TAB,0,FUN,20
REALLOCATE VAR,0,FSV,100,HSV,0,CHA,0,GRP,0,BVR,0,FMS,10,HMS,0
REALLOCATE MAC,10,COM,50000,XAC,200
```

```
BLOCK
NUMBER  *LOC    OPERATION  A,B,C,D,E,F,G              COMMENTS                                   CARD
                                                                                                 NUMBER
           1    MATRIX     X,1,1
           2    MATRIX     X,15,50
           3    MATRIX     X,20,50
           4    MATRIX     X,20,50
           5    MATRIX     X,10,50
           6    MATRIX     X,20,20
           7    MATRIX     X,1,1
           8    MATRIX     X,20,5
           9    MATRIX     X,1,1
        * ASSIGN PARAMETERS 1 TO 20 TO SAVEVALUES 1 TO 20
         PREP   STARTMACRO
                SAVEVALUE  1,*1
                SAVEVALUE  2,*2
                SAVEVALUE  3,*3
                SAVEVALUE  4,*4
                SAVEVALUE  5,*5
                SAVEVALUE  6,*6
                SAVEVALUE  7,*7
                SAVEVALUE  8,*8
                SAVEVALUE  9,*9
                SAVEVALUE  10,*10
                SAVEVALUE  11,*11
                SAVEVALUE  12,*12
                SAVEVALUE  13,*13
                SAVEVALUE  14,*14
                SAVEVALUE  15,*15
                SAVEVALUE  16,*16
                SAVEVALUE  17,*17
                SAVEVALUE  18,*18
                SAVEVALUE  19,*19
                SAVEVALUE  20,*20
                ENDMACRO
        * REASSIGN THE PARAMETERS 1 TO 20 FROM THE SAVEVALUES 1 TO 20
         ASSGN  STARTMACRO
                ASSIGN     1,X1
                ASSIGN     2,X2
                ASSIGN     3,X3
                ASSIGN     4,X4
                ASSIGN     5,X5
                ASSIGN     6,X6
                ASSIGN     7,X7
                ASSIGN     8,X8
                ASSIGN     9,X9
                ASSIGN     9,X9
                ASSIGN     10,X10
                ASSIGN     11,X11
                ASSIGN     12,X12
                ASSIGN     12,X12
                ASSIGN     13,X13
                ASSIGN     14,X14
                ASSIGN     15,X15
                ASSIGN     16,X16
                ASSIGN     17,X17
                ASSIGN     18,X18
                ASSIGN     19,X19
```

```
                ASSIGN      20,X20
                ENDMACRO
          * ASSIGN THE VALUES OF THE FORT SUBROUTINE IN ARG A,AND THE MATRIX
          *  NUMBERS OF THE NINE MATRICES TO BE PASSED
           FORT1 STARTMACRO
                SAVEVALUE   50,#A
                SAVEVALUE   51,#B
                SAVEVALUE   52,#C
                SAVEVALUE   53,#D
                SAVEVALUE   54,#E
                SAVEVALUE   55,#F
                SAVEVALUE   56,#G
                SAVEVALUE   57,#H
                SAVEVALUE   58,#I
                SAVEVALUE   59,#J
                ENDMACRO
          * EVALUATE SCALARS,FUNCTIONS,PARAMETERS,VARIABLES,COUNTERS
           EVAL  STARTMACRO
                TEST E      *13,K1,#B              SCALAR=A(*16)
                ASSIGN      13,*16
                TRANSFER    ,#A
           #B   TEST E      *13,K2,#C        FUNCTION
                SAVEVALUE   91,FN*16
                ASSIGN      13,X91
                TRANSFER    ,#A
           #C   TEST E      *13,K3,#D        VARIABLE
                ASSIGN      13,K99193
           FORT1 MACRO      K3,K1,K2,K3,K4,K5,K6,K7,K8,K9
           PREP  MACRO
                HELP        TEST
           ASSGN MACRO
                TEST NE     *13,K99193,GPR33
                TRANSFER    ,#A
           #D   TEST E      *13,K4,#E        PARAMETER
                ASSIGN      13,P*16
                TRANSFER    ,#A
           #E   TEST E      *13,K5,#F        COUNTER
                ASSIGN      16+,K20
                ASSIGN      13,X*16
                TRANSFER    ,#A
           #F   ASSIGN      9,K11            ERROR INVALIC ARGUMENT WHEN USED IN
                TRANSFER    ,GPR35                   THIS CONTEXT.
                ENDMACRO
           1    STORAGE     500
                INITIAL     X68,1
           1    FUNCTION    RN1,C6
          C,1/.2,2/.4,3/.6,6/.8,8/1.0,11
           2    FUNCTION    RN2,D5
          C,0/.5,21/.7,24/.9,27/1.0,30
           3    FUNCTION    RN3,D5
          .1,7/.3,8/.5,9/.9,10/1.0,11
          * SYSTEM INITIALIZATION
1          GPR00 GENERATE   ,,,1,,20
                RESET
2               ASSIGN      8,K17
3               SAVEVALUE   97,K0
4               SAVEVALUE   97-,K1
```

```
 5                TRANSFER    ,GPR01
          * INITIALIZE STORAGES
 6        GPR02   SAVEVALUE   99,K0
 7        STINT   SAVEVALUE   99+,K1
 8                TEST LE     X99,X68,GPR18
 9                ENTER       X99,MX3(3,X99)
10                TRANSFER    ,STINT
          * USER GENERATE STATEMENTS WILL BE CONCATENATED HERE:::::
11                GENERATE    7,1,0,1000,,20
12                ASSIGN      8,K18
13                ASSIGN      6,K100
14                TRANSFER    ,GPR01
15                GENERATE    7,0,0,100,,20
16                ASSIGN      8,K18
17                ASSIGN      6,K200
18                TRANSFER    ,GPR01
19        GPR27   ASSIGN      11,MX4(5,*7)        SPLIT CONTROL BLOCK
20                SPLIT       *11,GPA27
21                ASSIGN      8,K23
22                TRANSFER    ,GPR01
23        GPA27   ASSIGN      8,K24               COPIES GO TO 1026 IN FORT
24                TRANSFER    ,GPR01
          * GATHER CONTROL BLOCK.......
25        GPR17   TEST E      MX4(7,*7),X97,GPA17
26                SPLIT       K1,GPB17
27        GPA17   GATE LS     *7
28                MSAVEVALUE  4-,8,*7,K1
29                TEST E      MX4(8,*7),K0,GPR01
30                LOGIC R     *7
31                TRANSFER    ,GPR01
32        GPB17   LOGIC S     *7
33                TRANSFER    ,GPR18
34        GPR28   ASSIGN      13,F*13             FACILITY AVAIL=0
35                TRANSFER    ,GPR01
          *
36        GPR29   ASSIGN      13,Q*13             QUEUE CONTENTS
37                TRANSFER    ,GPR01
          *
38        GPR30   ASSIGN      13,S*13             STORAGE CONTENTS
39                TRANSFER    ,GPR01
          *
40        GPR22   SAVEVALUE   99,K0               FAC AVAIL=MX8
41        GPA22   SAVEVALUE   99+,K1
42                TEST LE     X99,*19,GPR01
43                ASSIGN      18,MX8(X99,3)
44                ASSIGN      17,MX2(10,*18)      TEST IF FAC HAS BEEN INITIALIZED
45                TEST E      *17,K0,GPB22        BY THE SEIZING BY A TRANS.
46                MSAVEVALUE  8,X99,2,K0
47                TRANSFER    ,GPA22
48        GPB22   PRINT       99,99,X
49                MSAVEVALUE  8,X99,2,F*18
50                TRANSFER    ,GPA22
          *
51        GPR23   SAVEVALUE   99,K0               FACILITY UTILIZATION=MX8
52        GPA23   SAVEVALUE   99+,K1
53                TEST LE     X99,*19,GPR01
54                ASSIGN      18,MX8(X99,3)
```

```
55            ASSIGN      17,MX2(10,*18)  TEST IF FAC IS INITIALIZED...     170
56            TEST E      *17,K0,GPB23                                      171
57            MSAVEVALUE  8,X99,2,K0                                        172
58            TRANSFER    ,GPA23                                            173
59      GPB23 MSAVEVALUE  8,X99,2,FR*18                                     174
60            TRANSFER    ,GPA23                                            175
       *                                                                    176
61      GPR24 SAVEVALUE   99,K0           STORAGE UTILIZATION=MX8           177
62      GPA24 SAVEVALUE   99+,K1                                            178
63            TEST LE     X99,*19,GPR01                                     179
64            ASSIGN      18,MX8(X99,3)                                     180
65            MSAVEVALUE  8,X99,2,SR*18                                     181
66            TRANSFER    ,GPA24                                            182
       *                                                                    183
67      GPR25 SAVEVALUE   99,K0           STORAGE CONTENTS=MX8              184
68      GPA25 SAVEVALUE   99+,K1                                            185
69            TEST LE     X99,*19,GPR01                                     186
70            ASSIGN      18,MX8(X99,3)                                     187
71            MSAVEVALUE  8,X99,2,S*18                                      188
72            TRANSFER    ,GPA25                                            189
       *                                                                    190
73      GPR26 SAVEVALUE   99,K0           QUEUE CONTENTS                    191
74      GPA26 SAVEVALUE   99+,K1                                            192
75            TEST LE     X99,*19,GPR01                                     193
76            ASSIGN      18,MX8(X99,3)                                     194
77            MSAVEVALUE  8,X99,2,Q*18                                      195
78            TRANSFER    ,GPA26                                            196
       *                                                                    197
79      GPR11 QUEUE       *7              *** QUEUE PROCESSING ***          198
80            ASSIGN      17,K2           FREE PREVIOUS ENTITY              199
81            TRANSFER    ,FREE                                             200
82      QUEX  ASSIGN      14,K2                                             201
83            ASSIGN      15,*7                                             202
84            TRANSFER    ,GPR01                                            203
       *                                                                    204
85      GPR12 ENTER       *7,*13           STORAGE PROCESSING?TEST FOR STR FUL  205
86            ASSIGN      17,K1           FREE PREVIOUS ENTITY              206
87            TRANSFER    ,FREE                                             207
88      STRX  TRANSFER    ,GPA19                                            208
       *                                                                    209
89      GPR13 SEIZE       *7              FACILITY PROCESSING               210
90            ASSIGN      17,K0                                             211
91            TRANSFER    ,FREE           FREE PREVIOUS PHYSICAL ENTITY     212
92      FACX  ASSIGN      13,MX2(2,*7)    EVALUATE PROCESSING TIME          213
93            ASSIGN      16,MX2(3,*7)                                      214
        EVAL  MACRO       GPA13,GPB13,GPC13,GPD13,GPE13,GPF13               215
94            TEST E      *13,K1,GPB13                                      215
95            ASSIGN      13,*16                                            215
96            TRANSFER    ,GPA13                                            215
97      GPB13 TEST E      *13,K2,GPC13                                      215
98            SAVEVALUE   91,FN*16                                          215
99            ASSIGN      13,X91                                            215
100           TRANSFER    ,GPA13                                            215
101     GPC13 TEST E      *13,K3,GPD13                                      215
102           ASSIGN      13,K99193                                         215
        FORT1 MACRO       K3,K1,K2,K3,K4,K5,K6,K7,K8,K9                     215
103           SAVEVALUE   50,K3                                             215
```

```
104          SAVEVALUE   51,K1                                  215
105          SAVEVALUE   52,K2                                  215
106          SAVEVALUE   53,K3                                  215
107          SAVEVALUE   54,K4                                  215
108          SAVEVALUE   55,K5                                  215
109          SAVEVALUE   56,K6                                  215
110          SAVEVALUE   57,K7                                  215
111          SAVEVALUE   58,K8                                  215
112          SAVEVALUE   59,K9                                  215
       PREP  MACRO                                              215
113          SAVEVALUE   1,*1                                   215
114          SAVEVALUE   2,*2                                   215
115          SAVEVALUE   3,*3                                   215
116          SAVEVALUE   4,*4                                   215
117          SAVEVALUE   5,*5                                   215
118          SAVEVALUE   6,*6                                   215
119          SAVEVALUE   7,*7                                   215
120          SAVEVALUE   8,*8                                   215
121          SAVEVALUE   9,*9                                   215
122          SAVEVALUE   10,*10                                 215
123          SAVEVALUE   11,*11                                 215
124          SAVEVALUE   12,*12                                 215
125          SAVEVALUE   13,*13                                 215
126          SAVEVALUE   14,*14                                 215
127          SAVEVALUE   15,*15                                 215
128          SAVEVALUE   16,*16                                 215
129          SAVEVALUE   17,*17                                 215
130          SAVEVALUE   18,*18                                 215
131          SAVEVALUE   19,*19                                 215
132          SAVEVALUE   20,*20                                 215
133          HELP        TEST                                   215
       ASSGN MACRO                                              215
134          ASSIGN      1,X1                                   215
135          ASSIGN      2,X2                                   215
136          ASSIGN      3,X3                                   215
137          ASSIGN      4,X4                                   215
138          ASSIGN      5,X5                                   215
139          ASSIGN      6,X6                                   215
140          ASSIGN      7,X7                                   215
141          ASSIGN      8,X8                                   215
142          ASSIGN      9,X9                                   215
143          ASSIGN      9,X9                                   215
144          ASSIGN      10,X10                                 215
145          ASSIGN      11,X11                                 215
146          ASSIGN      12,X12                                 215
147          ASSIGN      12,X12                                 215
148          ASSIGN      13,X13                                 215
149          ASSIGN      14,X14                                 215
150          ASSIGN      15,X15                                 215
151          ASSIGN      16,X16                                 215
152          ASSIGN      17,X17                                 215
153          ASSIGN      18,X18                                 215
154          ASSIGN      19,X19                                 215
155          ASSIGN      20,X20                                 215
156          TEST NE     *13,K99193,GPR33                       215
157          TRANSFER    ,GPA13                                 215
158    GPD13 TEST E      *13,K4,GPE13                           215
```

```
159                ASSIGN     13,P*16                                                       215
160                TRANSFER   ,GPA13                                                        215
161         GPE13  TEST E     *13,K5,GPF13                                                  215
162                ASSIGN     16+,K20                                                       215
163                ASSIGN     13,X*16                                                       215
164                TRANSFER   ,GPA13                                                        215
165         GPF13  ASSIGN     9,K11                                                         215
166                TRANSFER   ,GPR35                                                        215
167         GPA13  ADVANCE    *13                                                           216
168                ASSIGN     15,*7            *15=PREVIOUS LAST PHYSICAL ENTITY            217
169                ASSIGN     14,K0                                                         218
170                TRANSFER   ,GPR01                                                        219
            *                                                                               220
171         GPR10  ASSIGN     13,*13           EVALUATE ATTRIBUTE ROUTINE                   221
            EVAL   MACRO      GPA10,GPB10,GPC10,GPD10,GPE10,GPF10                           222
172                TEST E     *13,K1,GPB10                                                  222
173                ASSIGN     13,*16                                                        222
174                TRANSFER   ,GPA10                                                        222
175         GPB10  TEST E     *13,K2,GPC10                                                  222
176                SAVEVALUE  91,FN*16                                                      222
177                ASSIGN     13,X91                                                        222
178                TRANSFER   ,GPA10                                                        222
179         GPC10  TEST E     *13,K3,GPD10                                                  222
180                ASSIGN     13,K99193                                                     222
            FORT1  MACRO      K3,K1,K2,K3,K4,K5,K6,K7,K8,K9                                 222
181                SAVEVALUE  50,K3                                                         222
182                SAVEVALUE  51,K1                                                         222
183                SAVEVALUE  52,K2                                                         222
184                SAVEVALUE  53,K3                                                         222
185                SAVEVALUE  54,K4                                                         222
186                SAVEVALUE  55,K5                                                         222
187                SAVEVALUE  56,K6                                                         222
188                SAVEVALUE  57,K7                                                         222
189                SAVEVALUE  58,K8                                                         222
190                SAVEVALUE  59,K9                                                         222
            PREP   MACRO                                                                    222
191                SAVEVALUE  1,*1                                                          222
192                SAVEVALUE  2,*2                                                          222
193                SAVEVALUE  3,*3                                                          222
194                SAVEVALUE  4,*4                                                          222
195                SAVEVALUE  5,*5                                                          222
196                SAVEVALUE  6,*6                                                          222
197                SAVEVALUE  7,*7                                                          222
198                SAVEVALUE  8,*8                                                          222
199                SAVEVALUE  9,*9                                                          222
200                SAVEVALUE  10,*10                                                        222
201                SAVEVALUE  11,*11                                                        222
202                SAVEVALUE  12,*12                                                        222
203                SAVEVALUE  13,*13                                                        222
204                SAVEVALUE  14,*14                                                        222
205                SAVEVALUE  15,*15                                                        222
206                SAVEVALUE  16,*16                                                        222
207                SAVEVALUE  17,*17                                                        222
208                SAVEVALUE  18,*18                                                        222
209                SAVEVALUE  19,*19                                                        222
210                SAVEVALUE  20,*20                                                        222
211                HELP       TEST                                                          222
```

```
      ASSGN MACRO                                                          222
212         ASSIGN     1,X1                                                222
213         ASSIGN     2,X2                                                222
214         ASSIGN     3,X3                                                222
215         ASSIGN     4,X4                                                222
216         ASSIGN     5,X5                                                222
217         ASSIGN     6,X6                                                222
218         ASSIGN     7,X7                                                222
219         ASSIGN     8,X8                                                222
220         ASSIGN     9,X9                                                222
221         ASSIGN     9,X9                                                222
222         ASSIGN     10,X10                                              222
223         ASSIGN     11,X11                                              222
224         ASSIGN     12,X12                                              222
225         ASSIGN     12,X12                                              222
226         ASSIGN     13,X13                                              222
227         ASSIGN     14,X14                                              222
228         ASSIGN     15,X15                                              222
229         ASSIGN     16,X16                                              222
230         ASSIGN     17,X17                                              222
231         ASSIGN     18,X18                                              222
232         ASSIGN     19,X19                                              222
233         ASSIGN     20,X20                                              222
234         TEST NE    *13,K99193,GPR33                                    222
235         TRANSFER   ,GPA10                                              222
236   GPD10 TEST E     *13,K4,GPE10                                        222
237         ASSIGN     13,P*16                                             222
238         TRANSFER   ,GPA10                                              222
239   GPE10 TEST E     *13,K5,GPF10                                        222
240         ASSIGN     16+,K20                                             222
241         ASSIGN     13,X*16                                             222
242         TRANSFER   ,GPA10                                              222
243   GPF10 ASSIGN     9,K11                                               222
244         TRANSFER   ,GPR35                                              222
245   GPA10 TRANSFER   ,GPR01                                              223
      *                                                                    224
246   GPR18 TERMINATE                                                      225
      *                                                                    226
247   GPR19 ASSIGN     17,K3                                               227
248         TRANSFER   ,FREE                                               228
249   GPA19 TERMINATE                                                      229
      *                LEAVE CONTROL BLOCK                                 230
250   GPR20 ASSIGN     17,K5              FREE PREV ENTITY                 231
251         TRANSFER   ,FREE                                               232
252   GPA20 ASSIGN     14,K1              LAST ENT = STR TYPE              233
253         ASSIGN     15,*7              SYSTEM STR #=*7                  234
254         ASSIGN     6,*13              PRESENT ENTITY                   235
255         TRANSFER   ,GPR01                                              236
      *                                                                    237
      *                                                                    238
256   GPR01 ASSIGN     1,*1               BRANCH TO FORTRAN CONTROL        239
      FORT1 MACRO      K1,K1,K2,K3,K4,K5,K6,K7,K8,K9                       240
257         SAVEVALUE  50,K1                                               240
258         SAVEVALUE  51,K1                                               240
259         SAVEVALUE  52,K2                                               240
260         SAVEVALUE  53,K3                                               240
261         SAVEVALUE  54,K4                                               240
```

```
262              SAVEVALUE  55,K5                                          240
263              SAVEVALUE  56,K6                                          240
264              SAVEVALUE  57,K7                                          240
265              SAVEVALUE  58,K8                                          240
266              SAVEVALUE  59,K9                                          240
         PREP    MACRO                                                     241
267              SAVEVALUE  1,*1                                           241
268              SAVEVALUE  2,*2                                           241
269              SAVEVALUE  3,*3                                           241
270              SAVEVALUE  4,*4                                           241
271              SAVEVALUE  5,*5                                           241
272              SAVEVALUE  6,*6                                           241
273              SAVEVALUE  7,*7                                           241
274              SAVEVALUE  8,*8                                           241
275              SAVEVALUE  9,*9                                           241
276              SAVEVALUE  10,*10                                         241
277              SAVEVALUE  11,*11                                         241
278              SAVEVALUE  12,*12                                         241
279              SAVEVALUE  13,*13                                         241
280              SAVEVALUE  14,*14                                         241
281              SAVEVALUE  15,*15                                         241
282              SAVEVALUE  16,*16                                         241
283              SAVEVALUE  17,*17                                         241
284              SAVEVALUE  18,*18                                         241
285              SAVEVALUE  19,*19                                         241
286              SAVEVALUE  20,*20                                         241
287              HELP       TEST                                           242
         ASSGN   MACRO                                                     243
288              ASSIGN     1,X1                                           243
289              ASSIGN     2,X2                                           243
290              ASSIGN     3,X3                                           243
291              ASSIGN     4,X4                                           243
292              ASSIGN     5,X5                                           243
293              ASSIGN     6,X6                                           243
294              ASSIGN     7,X7                                           243
295              ASSIGN     8,X8                                           243
296              ASSIGN     9,X9                                           243
297              ASSIGN     9,X9                                           243
298              ASSIGN     10,X10                                         243
299              ASSIGN     11,X11                                         243
300              ASSIGN     12,X12                                         243
301              ASSIGN     12,X12                                         243
302              ASSIGN     13,X13                                         243
303              ASSIGN     14,X14                                         243
304              ASSIGN     15,X15                                         243
305              ASSIGN     16,X16                                         243
306              ASSIGN     17,X17                                         243
307              ASSIGN     18,X18                                         243
308              ASSIGN     19,X19                                         243
309              ASSIGN     20,X20                                         243
         *                                                                 244
310              TEST NE    X61,K1,GPR01    GPSS BRANCH & PROCESSING CONTROL  245
311              TEST NE    X61,K2,GPR02                                   246
         *       TEST NE    X61,K3,GPR03                                   247
         *       TEST NE    X61,K4,GPR04                                   248
         *       TEST NE    X61,K5,GPR05                                   249
         *       TEST NE    X61,K6,GPR06                                   250
```

```
      *       TEST NE      X61,K7,GPRC7
      *       TEST NE      X61,K8,GPRC8
      *       TEST NE      X61,K9,GPR09
312           TEST NE      X61,K10,GPR10
313           TEST NE      X61,K11,GPR11
314           TEST NE      X61,K12,GPR12
315           TEST NE      X61,K13,GPR13
316           TEST NE      X61,K14,GPR14
      *       TEST NE      X61,K15,GPR15
      *       TEST NE      X61,K16,GPR16
317           TEST NE      X61,K17,GPR17
318           TEST NE      X61,K18,GPR18
319           TEST NE      X61,K19,GPR19
320           TEST NE      X61,K20,GPR20
321           TEST NE      X61,K21,GPR21
322           TEST NE      X61,K22,GPR22
323           TEST NE      X61,K23,GPR23
324           TEST NE      X61,K24,GPR24
325           TEST NE      X61,K25,GPR25
326           TEST NE      X61,K26,GPR26
327           TEST NE      X61,K27,GPR27
328           TEST NE      X61,K28,GPR28
329           TEST NE      X61,K29,GPR29
330           TEST NE      X61,K30,GPR30
331           TEST NE      X61,K31,GPR31
332           TEST NE      X61,K32,GPR32
333           TEST NE      X61,K33,GPR33
      *       TEST NE      X61,K34,GPR34
      *       TEST NE      X61,K35,GPR35
334           ASSIGN       9,K60         ERROR:SYSTEM:REQUESTED ROUTINE
335           TRANSFER     ,GPR35            NON-EXISTANT.
      *
336     GPR31 SAVEVALUE    96,C1
337           TRANSFER     ,GPR01        CLOCK TIME=*13
      *    END OF RUN COLLECT STATS....
338     GPR32 ASSIGN       17,K4         FREE PREV PHYS ENTITY
339           SAVEVALUE    96,C1         ASSIGN CLOCK TO X96
340           TRANSFER     ,FREE
341     GPF32 ASSIGN       18,K0         FACILITY STATS
342     GPA32 ASSIGN       18+,K1
343           TEST LE      *18,X67,GPB32
344           MSAVEVALUE 2,8,*18,FR*18   UTILIZATION
345           MSAVEVALUE 2,7,*18,FC*18   COUNT
346           MSAVEVALUE 2,9,*18,FT*18   AVE TIME/TRANS
347           TRANSFER     ,GPA32
      *
348     GPB32 ASSIGN       18,K0         STORAGE STATS
349     GPC32 ASSIGN       18+,K1
350           TEST LE      *18,X68,GPD32
351           MSAVEVALUE 3,14,*18,SC*18  COUNT
352           MSAVEVALUE 3,15,*18,SR*18  UTIL.
353           MSAVEVALUE 3,16,*18,SA*18  AVE CONTENTS
354           MSAVEVALUE 3,17,*18,SM*18  MAX CONTENTS
355           MSAVEVALUE 3,18,*18,ST*18  AVE TIME/TRANS
356           MSAVEVALUE 3,19,*18,S*18   CURRENT CONTENTS
357           TRANSFER     ,GPC32
      *
```

```
358      GPD32 ASSIGN      18,K0              QUEUE STATS
359      GPE32 ASSIGN      18+,K1
360            TEST LE     *18,X70,GPR01
361            MSAVEVALUE  5,5,*18,QC*18      CCUNT
362            MSAVEVALUE  5,6,*18,QA*18      AVE CONTENTS
363            MSAVEVALUE  5,7,*18,QM*18      MAX CONTENTS
364            MSAVEVALUE  5,8,*18,QZ*18      ZERO ENTRIES
365            MSAVEVALUE  5,9,*18,QT*18      AVE TIME/TRANS
366            MSAVEVALUE  5,10,*18,QX*18     AVE TIME (NO ZEROS)
367            TRANSFER    ,GPE32
         *
368      GPR33 ASSIGN      9,K12              UNDEFINED VARIABLE REFERENCED
369            TRANSFER    ,GPR35
         *
370      GPR35 ASSIGN      8,K25              *8=25 >> 1004 ERROR IN FORT
371            TRANSFER    ,GPR01
         *
372      FREE  TEST NE     *14,K0,FRE0        FREE LAST PHYSICAL ENTITY
373            TEST NE     *14,K1,FRE1
374            TEST NE     *14,K2,FRE2
375            TEST NE     *14,X97,FRE99      FIRST PHYSICAL ENTITY
376            ASSIGN      9,K81              SYSTEM ERROR 81
377            TRANSFER    ,GPR35
378      FRE0  RELEASE     *15                *15=FAC # (GPR13)
379            TRANSFER    ,FRE99
380      FRE1  ASSIGN      13,K0              STR-QUE. TYPE STORAGE
381            ASSIGN      13,MX3(6,*15)      EVAL UNITS LEAVING STR
382            ASSIGN      16,MX3(7,*15)
         EVAL  MACRO       FRE4,FRE10,FRE20,FRE30,FRE40,FRE50
383            TEST E      *13,K1,FRE10
384            ASSIGN      13,*16
385            TRANSFER    ,FRE4
386      FRE10 TEST E      *13,K2,FRE20
387            SAVEVALUE   91,FN*16
388            ASSIGN      13,X91
389            TRANSFER    ,FRE4
390      FRE20 TEST E      *13,K3,FRE30
391            ASSIGN      13,K99193
         FORT1 MACRO       K3,K1,K2,K3,K4,K5,K6,K7,K8,K9
392            SAVEVALUE   50,K3
393            SAVEVALUE   51,K1
394            SAVEVALUE   52,K2
395            SAVEVALUE   53,K3
396            SAVEVALUE   54,K4
397            SAVEVALUE   55,K5
398            SAVEVALUE   56,K6
399            SAVEVALUE   57,K7
400            SAVEVALUE   58,K8
401            SAVEVALUE   59,K9
         PREP  MACRO
402            SAVEVALUE   1,*1
```

```
403    SAVEVALUE  2,*2     336
404    SAVEVALUE  3,*3     336
405    SAVEVALUE  4,*4     336
406    SAVEVALUE  5,*5     336
407    SAVEVALUE  6,*6     336
408    SAVEVALUE  7,*7     336
```

```
409              SAVEVALUE  8,*8                                                    336
410              SAVEVALUE  9,*9                                                    336
411              SAVEVALUE  10,*10                                                  336
412              SAVEVALUE  11,*11                                                  336
413              SAVEVALUE  12,*12                                                  336
414              SAVEVALUE  13,*13                                                  336
415              SAVEVALUE  14,*14                                                  336
416              SAVEVALUE  15,*15                                                  336
417              SAVEVALUE  16,*16                                                  336
418              SAVEVALUE  17,*17                                                  336
419              SAVEVALUE  18,*18                                                  336
420              SAVEVALUE  19,*19                                                  336
421              SAVEVALUE  20,*20                                                  336
422              HELP       TEST                                                    336
           ASSGN MACRO                                                              336
423              ASSIGN     1,X1                                                    336
424              ASSIGN     2,X2                                                    336
425              ASSIGN     3,X3                                                    336
426              ASSIGN     4,X4                                                    336
427              ASSIGN     5,X5                                                    336
428              ASSIGN     6,X6                                                    336
429              ASSIGN     7,X7                                                    336
430              ASSIGN     8,X8                                                    336
431              ASSIGN     9,X9                                                    336
432              ASSIGN     9,X9                                                    336
433              ASSIGN     10,X10                                                  336
434              ASSIGN     11,X11                                                  336
435              ASSIGN     12,X12                                                  336
436              ASSIGN     12,X12                                                  336
437              ASSIGN     13,X13                                                  336
438              ASSIGN     14,X14                                                  336
439              ASSIGN     15,X15                                                  336
440              ASSIGN     16,X16                                                  336
441              ASSIGN     17,X17                                                  336
442              ASSIGN     18,X18                                                  336
443              ASSIGN     19,X19                                                  336
444              ASSIGN     20,X20                                                  336
445              TEST NE    *13,K99193,GPR33                                        336
446              TRANSFER   ,FRE4                                                   336
447        FRE30 TEST E     *13,K4,FRE40                                            336
448              ASSIGN     13,P*16                                                 336
449              TRANSFER   ,FRE4                                                   336
450        FRE40 TEST E     *13,K5,FRE50                                            336
451              ASSIGN     16+,K20                                                 336
452              ASSIGN     13,X*16                                                 336
453              TRANSFER   ,FRE4                                                   336
454        FRE50 ASSIGN     9,K11                                                   336
455              TRANSFER   ,GPR35                                                  336
456        FRE4  TEST G     *13,S*15,GPR21                                          337
457              ASSIGN     12,*13                                                  338
458              ASSIGN     13,S*15                                                 339
459              ASSIGN     9,*8                                                    340
460              ASSIGN     8,K27                                                   341
461              ASSIGN     19,C1                                                   342
462              TRANSFER   ,GPR01                                                  343
463        GPR21 LEAVE      *15,*13                                                 344
464              TRANSFER   ,FRE99                                                  345
```

```
465    FRE2  DEPART     *15            QUEUF
466    FRE99 TEST NE    *17,K0,FACX
467          TEST NE    *17,K1,STRX
468          TEST NE    *17,K2,QUEX
469          TEST NE    *17,K3,GPA19   REMOVE & TERMINATE NON RUN END TRANS
470          TEST NE    *17,K4,GPF32   TERMINATE RUN
471          TEST NE    *17,K5,GPA20   LEAVE -FREE PREV ENTITY
472          ASSIGN     9,K94          SYSTEM ERROR 94
473          TRANSFER   ,GPR35
      * INSERT ALL NEW ROUTINES BEFORE 'GPR14'
474    GPR14 TERMINATE  1              FINAL SIMULATICN TERMINATICN
             START      1,NP
             SIMULATE
             END
```

APPENDIX F

ASSEMBLER ROUTINE

SYMBOL	TYPE	ID	ADDR	LENGTH	LD ID
LINK	SD	01	000000	00014A	
SIMT01	ER	02			
SIMT03	ER	03			
SIMT09	ER	04			

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                        F08APR70    6/21/71

000000                              1 LINK     CSECT
000000                              2          USING LINK,15
000000 9028 F118            00118   3          STM   2,8,SAVE
000004 0570                         4          BALR  7,0
000006                              5          USING *,7
                                    6          DROP  15
000006 581A 0018            00018   7          L     1,24(10)               GET ADDR OF CONTROL WORD
00000A 5821 0400            00400   8          L     2,1024(1)          R2=ADDR OF SAVEVALUE BASE
00000E 4122 0004            00004   9          LA    2,4(2)             R2=ADDR OF 1ST SAVEVALUE
000012 5020 70C2            000C8  10          ST    2,PARMLIS          PARM1=FIRST SAVEVALUE
000016 5862 00C4            000C4  11          L     6,196(2)         R6=CONTENTS OF SAVEVALUE 50
00001A 5851 0414            00414  12          L     5,1044(1)          R5=MATRIX CONTROL BASE
00001E 5832 00C8            000C8  13          L     3,200(2)           R3=CONTENTS OF SAVEVALUE 51
000022 4C30 7142            00148  14          MH    3,=H'24'            R3=R3*24
000026 5835 3000            00000  15          L     3,0(5,3)           R3=ADDR OF MATRIX(X51)
00002A 5030 70C6            000CC  16          ST    3,PARMLIS+4        PARM2=MATRIX(X51)
00002E 5832 00CC            000CC  17          L     3,204(2)           R3=CONTENTS OF SAVEVALUE 52
000032 4C30 7142            00148  18          MH    3,=H'24'            R3=R3*24
000036 5835 3000            00000  19          L     3,0(5,3)           R3=ADDR OF MATRIX(X52)
00003A 5030 70CA            000D0  20          ST    3,PARMLIS+8        PARM3=MATRIX(X52)
00003E 5832 00D0            000D0  21          L     3,208(2)           R3=CONTENTS OF SAVEVALUE 53
000042 4C30 7142            00148  22          MH    3,=H'24'            R3=R3*24
000046 5835 3000            00000  23          L     3,0(5,3)           R3=ADDR OF MATRIX(X53)
00004A 5030 70CE            000D4  24          ST    3,PARMLIS+12       PARM4=MATRIX(X53)
00004E 5832 00D4            000D4  25          L     3,212(2)           R3=CONTENTS OF SAVEVALUE 54
000052 4C30 7142            00148  26          MH    3,=H'24'            R3=R3*24
000056 5835 3000            00000  27          L     3,0(5,3)           R3=ADDR OF MATRIX(X54)
00005A 5030 70D2            000D8  28          ST    3,PARMLIS+16       PARM5=MATRIX(X54)
00005E 5832 00D8            000D8  29          L     3,216(2)           R3=CONTENTS OF SAVEVALUE 55
000062 4C30 7142            00148  30          MH    3,=H'24'            R3=R3*24
000066 5835 3000            00000  31          L     3,0(5,3)           R3=ADDR OF MATRIX(X55)
00006A 5030 70D6            000DC  32          ST    3,PARMLIS+20       PARM6=MATRIX(X55)
00006E 5832 00DC            000DC  33          L     3,220(2)           R3=CONTENTS OF SAVEVALUE 56
000072 4C30 7142            00148  34          MH    3,=H'24'            R3=R3*24
000076 5835 3000            00000  35          L     3,0(5,3)           R3=ADDR OF MATRIX(X56)
00007A 5030 70DA            000E0  36          ST    3,PARMLIS+24       PARM7=MATRIX(X56)
00007E 5832 00E0            000E0  37          L     3,224(2)           R3=CONTENTS OF SAVEVALUE 57
000082 4C30 7142            00148  38          MH    3,=H'24'            R3=R3*24
000086 5835 3000            00000  39          L     3,0(5,3)           R3=ADDR OF MATRIX(X57)
00008A 5030 70DE            000E4  40          ST    3,PARMLIS+28       PARM8=MATRIX(X57)
00008E 5832 00E4            000E4  41          L     3,228(2)           R3=CONTENTS OF SAVEVALUE 58
000092 4C30 7142            00148  42          MH    3,=H'24'            R3=R3*24
000096 5835 3000            00000  43          L     3,0(5,3)           R3=ADDR OF MATRIX(X58)
00009A 5030 70E2            000E8  44          ST    3,PARMLIS+32       PARM9=MATRIX(X58)
00009E 5832 00E8            000E8  45          L     3,232(2)           R3=CONTENTS OF SAVEVALUE 59
0000A2 4C30 7142            00148  46          MH    3,=H'24'            R3=R3*24
0000A6 5835 3000            00000  47          L     3,0(5,3)           R3=ADDR OF MATRIX(X59)
0000AA 5030 70E6            000EC  48          ST    3,PARMLIS+36       PARM10=MATRIX(X59)
0000AE 4180 7132            00138  49          LA    8,SUBROUT
0000B2 0660                        50          BCTR  6,0
0000B4 8860 0002            00002  51          SLA   6,2
0000B8 4110 70C2            000C8  52          LA    1,PARMLIS
0000BC 58F6 8000            00000  53          L     15,0(6,8)
0000C0 9828 7112            00118  54          LM    2,8,SAVE
0000C4 07FF                        55          BR    15
```

1292

```
  LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F08APR70   6/21/71
000008                              56 PARMLIS  DS    20F
000118                              57 SAVE     DS    8F
000138 00000000                     58 SUBROUT  DC    V(SIMT01)
00013C 00000000                     59          DC    V(SIMT03)
000140 00000000                     60          DC    V(SIMT09)
000000                              61          END   LINK
000148 0018                         62                =H'24'
```

6/21/71

POS.ID	REL.ID	FLAGS	ADDRESS
01	02	1C	000138
01	03	1C	00013C
01	04	1C	000140

```
SYMBOL    LEN   VALJE  DEFN     REFERENCES                                                          6/21/71

LINK      00001 000000 00001    0002  0061
PARMLIS   00004 0000C8 00056    0010  0016  0020  0024  0028  0032  0036  0040  0044  0048  0052
SAVE      00004 000118 00057    0003  0054
SUBROUT   00C04 000138 00058    0049


NO STATEMENTS FLAGGED IN THIS ASSEMBLY
*STATISTICS*      SOURCE RECORDS (SYSIN) =     61
*OPTIONS IN EFFECT*   LIST, NODECK, LOAD, NORENT, XREF, NOTEST, ALGN, OS, LINECNT =  55
   88 PRINTED LINES
```

APPENDIX G

EDIT PROGRAM

```
  1       MAIN: PROC OPTIONS(MAIN);
  2       DCL (I1,I2,I,TP,CC,CR,L,LK,JI,KI,KI1,J,K,IX,ER,SC,SR,XK,LF,CMX,FMX,
          QMN,SMN,L4,FJ,FK,FR,FC,CR,QC,RR,RC,QM,FM,SM,CM,QUEM(50),STOM(50),
          FAM(50),COUM(50),LNF(20),VR(50),VVR(50),ILV,LV,BR,T,FI)BIN FIXED,
          (PD,XP,XS,XR,CBC,MX4(50,10),MX3(50,12),MX2(50,6),MX5(50,3),
  3       MX6(20,20))REAL FIXED DECIMAL ; MX2=0;MX3=0;MX4=0;MX5=0;MX6=0;
  8          DCL  (XX,A(56)) CHAR(1),(CV,CF) BIN FIXED,(GF(50),GV(50))REAL
          FIXED DECIMAL,
                 (GEN,AG8,A09,AG10,AG11,TRN,STR,FN1,FUN,VAR,BLK,FUNX,
                  VR1,SRR,A010,A011,A08,FL,VR2,AG,AG12,GEM,AG9)
  9              CHAR(80);DCL C8 CHAR(16);
 10          DCL  SYSIN  FILE INPUT STREAM,SYSPRINT FILE OUTPUT STREAM,
           (DISK1,DISK2,DISK3,DISK4,DISK5,DISK6) FILE OUTPUT  STREAM;
 11        DCL (CMN,FMN,CMX,SMX)REAL FIXED DECIMAL;
 12         DCL IHESARC ENTRY(FIXED BINARY(31,0)), EX FIXED BINARY(31,0);
 13       CMN=99999;FMN=0;CMX=-99999;SMX=-99999;CF=0;CV=0;
           /*   DEFINITION OF CHARACTER SET                              */
 19        A(1)=' '; A(2)=','; A(3)='/'; A(4)='*'; A(5)='('; A(6)=')'; A(7)=':';
 26        A(8)=';'; A(9)='>'; A(10)=' '; A(11)='#'; A(12)='='; A(13)='$';
 32        A(14)='&'; A(15)='|'; A(16)='@'; A(17)=' '; A(18)='1'; A(19)='2';
 38        A(20)='3'; A(21)='4'; A(22)='5'; A(23)='6'; A(24)='7'; A(25)='8';
 44        A(26)='9'; A(27)='0'; A(28)='+'; A(29)='-'; A(30)='A'; A(31)='B';
 50        A(32)='C'; A(33)='D'; A(34)='E'; A(35)='F'; A(36)='G'; A(37)='H';
 56        A(38)='I'; A(39)='J'; A(40)='K'; A(41)='L'; A(42)='M'; A(43)='N';
 62        A(44)='O'; A(45)='P'; A(46)='Q'; A(47)='R'; A(48)='S'; A(49)='T';
 68        A(50)='U'; A(51)='V'; A(52)='W'; A(53)='X'; A(54)='Y'; A(55)='Z';
 74        A(56)='.';      /* DEFINITION OF GENERATE CARDS */
 75             GEM='          GENERATE
 76                     ';  GEN=GEM;
 77             A09='          ASSIGN      6,K
 78                     ';     AG9=A09;     /* LAST ENTITY=A(6) */
 79             A08='          ASSIGN      8,K18
 80                     ';     AG8=A08;
 81             TRN='          TRANSFER    ,GPR01
                        ';
 82             SRR='          STORAGE
 83                     ';        STR=SRR;
 84             FUNX='          FUNCTION
                        ';
 85             BLK ='
 86                     ';     FUN=FUNX; FN1=BLK;   VAR=' '; VR1=' ';
 90       FI=0; IV=0;ILV=0;LV=0;CMX=0;FMX=0;CMN=0;SMN=0;DR=0;XS=0;
100       /* OPEN FILES*/SM=0;RR=0;SR=0;CR=0;QR=0;FR=0;FM=0;QM=0;CM=0;EX=0;
110         OPEN  FILE(SYSIN)  INPUT STREAM,
            FILE(SYSPRINT) OUTPUT STREAM,
                 FILE(DISK1)  OUTPUT  STREAM,
                 FILE(DISK2)  OUTPUT  STREAM,
                 FILE(DISK3) OUTPUT STREAM,
                 FILE(DISK4)  OUTPUT  STREAM,
                 FILE(DISK5) OUTPUT  STREAM,
111              FILE(DISK6) OUTPUT  STREAM;  SUBSTR(VR1,7,20)=
            'SUBROUTINE SIMT09(A)';
112         PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
114         SUBSTR(VR1,7,14)='INTEGER A(100)';
115       PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
117        SUBSTR(VR1,7,22)='DIMENSION X(100),V(20)';
118         PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
```

```
120              SUBSTR(VR1,7,8)='NV=A(16)';PUT FILE(DISK5) EDIT(VR1) (A(80));
122            VR1=' '; SUBSTR(VR1,7,21)='IF(NV.GT.A(72))RETURN';
124             PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
126            SUBSTR(VR1,7,12)='DO 1 I=1,100';
127             PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
129             SUBSTR(VR1,5,11)='1 X(I)=A(I)';
130             PUT FILE(DISK5) EDIT(VR1) (A(80)); VR1=' ';
132                ON ENDFILE(SYSIN) BEGIN; CLOSE FILE(SYSIN);
135                                         GO TO EDIT;
136                                  END;  CALL PRTHD;
138           START1:I=0;
139                 CALL READ80(FL);
140           START:I=I+1;
141                 IF SUBSTR(FL,I,81-I)=' ' THEN GO TO START1;
143                 XX=SUBSTR(FL,I,1);
144                 CALL CHAD(XX,I1,I2,A);
145           STAR2:IF I1>5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1; END;
151                 IF I=80 THEN DO; CALL READ80(FL); I=0; GO TO START; END;
157                 IF I1=1 THEN GO TO START;
159                 IF I1=2|I1=3 THEN DO;ER=1;CALL ERRB(ER,I,EX);GO TO START1;END;
165                 IF XX='F' THEN GO TO FVALUE;
167                 IF XX='Q' THEN GO TO QUEUE;
169                 IF XX='S' THEN GO TO STORAGE;
171                 IF XX='V' THEN GO TO VARIABLE;
173                 IF XX='C' THEN GO TO CONTROL;
175                 IF XX='D' THEN GO TO DECISION;
177                           ELSE DO; ER=2; CALL ERRB(ER,I,EX); GO TO START1;
181                             END;
182           FVALUE:I=I+1; XX=SUBSTR(FL,I,1);
184                 IF XX='N'| XX='U' THEN GO TO FUNCTION;
186                                  ELSE GO TO FACILITY;
187            READ80:PROC(FL);
188                 DCL FL CHAR(80), SYSIN FILE INPUT STREAM, SYSPRINT FILE OUTPUT
189           STREAM;  GET FILE(SYSIN)  EDIT(FL) (A(80));
190                    PUT FILE(SYSPRINT) EDIT(FL) (COLUMN(15),A(80));
191                  END READ80;
192            ERRA:PROC(ER,I,EX);
193              DCL EX FIXED BINARY(31,0);
194                DCL (I,ER) BIN FIXED, SYSPRINT FILE STREAM  OUTPUT;
195                    PUT FILE(SYSPRINT)
                   EDIT('ERROR..:NUMBER',ER,' PARAMETER =',I)
              (COLUMN(10),A(14),F(4),A(12),F(4));
196                    ER=0;EX=EX+1;
198                  END ERRA;
199            ERRB:PROC(ER,I,EX);
200                DCL (ER,I) BIN FIXED, EX FIXED BINARY(31,0);
201                    EX=EX+1;I=I-1;
203                    PUT FILE(SYSPRINT) EDIT('ERROR..:NUMBER',ER,' IN COLUMN ',I,
          ' OF THE ABOVE INPUT LINE') (COLUMN(10),A(14),F(4),A(11),F(2),A(24));
204             ER=0;   END ERRB;
206            CHAD:PROC(XX,I1,I2,A);
207                 DCL   (XX,A(56))CHAR(1),(I1,I2) BIN FIXED;
208                 DO I2 = 1 TO 56;
209                    IF XX = A(I2) THEN GO TO ID1;
211                 END;
212                 I1=99 ;
213                 GO TO ID2;
```

```
214        ID1: IF I2 < 18 THEN DO; I1=1; GO TO ID2; END;
219             IF I2 < 28 THEN DO; I1=2; GO TO ID2; END;
224             IF I2 < 30 THEN DO; I1=3; GO TO  ID2: END;
229             IF I2 < 55 THEN I1=4; ELSE I1=5;
232       ID2: END CHAD;  /* CHARACTER IDENTITY  */
         /*DELIMITERS I1=1/NUMERICS I1=2/SIGN I1=3/ALPHA I1=4/PERIOD I1=5*/
233       PRTHD:PROC;                 PUT FILE(SYSPRINT) EDIT(' ') (PAGE,A(1));
235                 PUT FILE(SYSPRINT) EDIT('AN ELEMENTARY SIMULATION SYSTEM',
         'THIS SIMULATION SYSTEM REQUIRES THAT THE PHYSICAL SYSTEM TO BE SIMULAT
         ED','BE DEFINE BY THE FOLLOWING MODEL DEFINITION STATEMENTS:-',
         ' DECISION RULE (D).....NEXT ENTITY CANDICATES & SPECIAL DEC. RULES FOR
          CHOICE  ',
              ' FACILITY (F)        ...DESCRIPTION OF THE FACILITY              ',
              ' QUEUE     (Q)       ...DESCRIPTION OF THE QUEUE                 ',
              ' STORAGE  (S)        ...DESCRIPTION OF THE STORAGE               ',
              ' CONTROL BLOCK  (C) ...DESCRIPTION OF THE CONTROL BLOCK          ',
              ' FUNCTION (FN)       ...DESCRIPTION OF THE FUNCTION              ',
              ' VARIABLE EQUATION(V)..DESCRIPTION OF THE EQUATION               ',
              ' MODEL DEFINITION INPUT SUMMARY AND EDIT')
              (COLUMN(35),A(31),SKIP,COLUMN(15),A(72),SKIP,COLUMN(15),A(56),
              SKIP,COLUMN(20),A(91),SKIP,COLUMN(20),A(63),SKIP,COLUMN(20),A(63),
              SKIP,COLUMN(20),A(63),SKIP,COLUMN(20),A(63),SKIP,COLUMN(20),A(63),
            SKIP,COLUMN(20),A(63),SKIP,COLUMN(20),A(63),SKIP,COLUMN(20),A(39));
236       PUT FILE(SYSPRINT) EDIT('....5...10....5...20....5...30....5...40....5
         ...50....5...60....5...70....5...80') (SKIP,COLUMN(15),A(80));
237           END PRTHD;
238      FUNCTION:L=0; LF=1;XK=0;L4=6;
242        FUN1:I=I+1;
243             XX=SUBSTR(FL,I,1);
244             CALL CHAD(XX,I1,I2,A);
245             IF I1>5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1;END;
251             IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
256                                                GO TO FUN1;
257                                            END;
258             IF I1  ¬=    1 THEN GO TO FUN1;
260      FUN2:I=I+1;
261            IF SUBSTR(FL,I,81-I)=' ' & L=L4 THEN GO TO FUNEND;
263             XX=SUBSTR(FL,I,1);
264             CALL CHAD(XX,I1,I2,A);
265             IF I1 >5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1;END;
271             IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
276                                                GO TO FUN2;
277                                           END;
278                                     ELSE GO TO FNNEND;
279             IF I1 =1 THEN GO TO FUN2;
281             J=I; JI=I; L=L+1;
284             IF I1 = 2 THEN GO TO FUN3;
286             IF I1 = 4 THEN IF L >  L4 THEN GO TO      FNNEND;
289                       ELSE DO;ER=4;CALL ERRB(ER,I,EX);GO TO START1;END;
294             IF I1=3 THEN GO TO FUN3;
296             IF I1=5 THEN GO TO FUN3;
298                      ELSE DO; ER=3; CALL ERRB(ER,I,EX);GO TO START1;END;
303      FUN3:
                I=I+1;
304             XX=SUBSTR(FL,I,1);
305             CALL CHAD(XX,I1,I2,A);
306             IF I1>5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1;END;
```

```
312                IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
317                                                END;
318                                          ELSE GO TO FNNEND;
319                IF I1=2|I1=5 THEN GO TO FUN3;
321                IF I1=4 THEN GO TO FNNEND;
323                IF I1¬=1 THEN DO;ER=1; CALL ERRB(ER,I,EX);GO TO START1;END;
329                K=I-J;XP=SUBSTR(FL,J,K);IF L=1 & K>2 THEN DO; ER=5;
334                           CALL ERRB(ER,I,EX);GO TO START1;END;
337                          IF L=1 THEN DO;
339                             SUBSTR(FUN,2,K)=SUBSTR(FL,J,K);
340                               FI=FI+1;MX6(20,FI)=XP;
342                               IF FI>20 | XP>20  THEN DO;ER=5;
345                                  CALL ERRB(ER,I,EX);GO TO START1;END;
348                             GO TO FUN2; END;
350                IF L= 2 THEN DO;XR=XP;IF XP=1 THEN DO;IX=20;
356                                                   SUBSTR(FUN,19,IX)='P';
357                                                   GO TO FUN2;
358                                                END;
359                             IF XP = 2 THEN DO;IX=21;
362                                                   SUBSTR(FUN,19,IX)='RN';
363                                                   GO TO FUN2;
364                                                END;
365                             IF XP = 3 THEN DO;IX=20;
368                                                   SUBSTR(FUN,19,IX)='X';
369                                                   GO TO FUN2;
370                                                END;
371                             IF XP=4  THEN  DO;IX=21;
374                                                   SUBSTR(FUN,19,IX)='CI';
375                                                   GO TO FUN2;
376                                                END;
377                   ELSE DO;ER=6;CALL ERRB(ER,I,EX);GO TO START1;END;
382                                END;
383                IF L=3 THEN DO;IF XR= 4 THEN GO TO FUN3;
387             IF XR=1 THEN DO;IF (XP<6 & XP>0)THEN GO TO FUN5;
391                      ELSE DO;ER=7;CALL ERRB(ER,I,EX);GO TO START1;END;END;
397             IF XR=2 THEN DO;IF (XP<9 & XP>0)THEN GO TO FUN5;
401                      ELSE DO;ER=8;CALL ERRB(ER,I,EX);GO TO START1;END;END;
407              IF XR=3 THEN DO;IF (XP<26 & XP>0)THEN DO;XP=XP+20;
412                                                      K=2;GO TO FUN5;
414                                                   END;
415                      ELSE DO;ER=9;CALL ERRB(ER,I,EX);GO TO START1;END;END;
421                  ELSE DO;ER=6;CALL ERRB(ER,I,EX);GO TO START1;END;
426        FUN5:C8=XP;SUBSTR(FUN,IX,K+1)=SUBSTR(C8,9-K,K) || ',';
428                IX=IX+K+1;GO TO FUN2;
430                          END;
431                IF L=4 THEN DO;IF XP=1 THEN DO;
435                                                SUBSTR(FUN,IX,1)='C';
436                                                IX=IX+1;
437                                                GO TO FUN2;
438                                             END;
439                          IF XP=2 THEN DO;
441                                                SUBSTR(FUN,IX,1)='D';
442                                                IX=IX+1;
443                                                GO TO FUN2;
444                                             END;
445                   ELSE DO;ER=10;CALL ERRB(ER,I,EX);GO TO START1;END;
450                           END;
```

```
                IF L=5 THEN DO;L4=XP*2+5;
                               SUBSTR(FUN,IX,K)=SUBSTR(FL,J,K);
                                 PUT FILE(DISK3) EDIT(FUN) (A(80)); FUN=FUNX;
                                 FX=1;XY=1;
                                GO TO FUN2;
                          END;
            IF L > L4 THEN DO;ER=11;CALL ERRB(ER,I,EX);GO TO START1;END;
            IF L>=6 THEN DO;
               IF XY=1 THEN DO;
                      IF FX¬=1 THEN DO;SUBSTR(FN1,FX,1)='/';FX=FX+1;END;
                      SUBSTR(FN1,FX,K+1)=SUBSTR(FL,J,K) || ',';
                       FX=FX+K+1;XY=0;GO TO FUN2;
                             END;
                         ELSE DO;/*LAST VAL WAS X(XY=1),THIS IS Y(XY=0)*/
                             XY=1;SUBSTR(FN1,FX,K)=SUBSTR(FL,J,K);
                              FX=FX+K;
                              END;/*IF L =L4 THEN GO TO FNXEND;*/
                      IF FX>=60 THEN DO;FX=1;
                                   PUT FILE(DISK3) EDIT(FN1) (A(80));
                                 FN1=' ';GO TO FUN2;
                                     END;
                                  ELSE GO TO FUN2;
                       END;
        FUNEND: FNNEND:
              IF L =L4 THEN GO TO        FNXEND;
                          ELSE DO; ER=4; CALL ERRB(ER,I,EX);GO TO START1;END;
        FNXEND: PUT FILE(DISK3) EDIT(FN1) (A(80));  FN1=' ';
              GO TO START;
        STORAGE:SR=SR+1; SC=0; L=0; LK=9;
        STR1:I=I+1;IF SR>50THEN DO;ER=16;CALL ERRB(ER,I,EX);GO TO START1;END;
              XX=SUBSTR(FL,I,1);
              CALL CHAD(XX,I1,I2,A);
              IF I1 > 5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
              IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL);
                                                I=0; GO TO STR1;
                                            END;
            ELSE DO; ER=12;CALL ERRB(ER,I,EX);GO TO START1;END;
              IF I1 ¬= 1 THEN GO TO STR1;
        STR2:I=I+1; XX=SUBSTR(FL,I,1);
             IF SUBSTR(FL,I,81-I)=' ' THEN IF L=LK THEN GO TO STZEND;
                  ELSE DO; ER=12; CALL ERRB(ER,I,EX); GO TO START1; END;
              CALL CHAD(XX,I1,I2,A);
              IF I1 > 5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
              IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
                                                 GO TO STR2;  END;
                                   ELSE  DO; L=L+1;
                                           GO TO STZEND; END;
              IF I1 = 1 THEN GO TO STR2;
              J=I;
              JI=I;
              L=L+1;
              IF L>LK THEN DO; ER=13; CALL ERRB(ER,I,EX); GO TO START1; END;
              IF I1 = 2 THEN GO TO STR3;
              IF I1 = 4 THEN GO TO SLZTEST;
         IF I2=29 THEN IF L=1 | L=11 THEN GO TO STR3;
                           ELSE DO;ER=14;CALL ERRB(ER,I,EX);GO TO STR2;END;
        STR3:I=I+1;
```

```
568                 XX=SUBSTR(FL,I,1);
589                 CALL CHAD(XX,I1,I2,A);
590                 IF I1 >5 THEN DO; ER=3; CALL ERRB(ER,I,EX); GO TO START1; END;
596                 IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
601                                               GO TO STR3;
602                                          END;
603                                     ELSE GO TO SLZTEST;
604                 IF I1=2 THEN GO TO STR3;
606                 IF I1=4 THEN GO TC SLZTEST;
608                 IF I1¬=1 THEN DO; ER=1; CALL ERRB(ER,I,EX); GO TO START1; END;
614                 SC=SC+1; KI=I-JI; K=I-J;
617                               IF L=1 THEN DO; XP= SUBSTR(FL,J,K);
620                                               IF XP>0 THEN XP=-XP;
622                                               SM=SM+1;
623               IF XP>SMX THEN SMX=XP;IF XP<SMN THEN SMN=XP;
627                                               STOM(SM)=XP;
628                                               MX3(SR,SC)=XP;
629                 K=1;IF SR>9 THEN K=2;    C8=SR;
633                  SUBSTR(STR,2,K)=SUBSTR(C8,10-K,K);
634                                               GC TO STR2; END;
636                 IF L= 2 THEN DO; XP=SUBSTR(FL,J,K);
639                                MX3(SR,SC)=XP;
640                                SUBSTR(STR,19,KI)=SUBSTR(FL,JI,KI);
641                                GO TO STR2;
642                             END;
643                        ELSE DO; XP=SUBSTR(FL,J,K); MX3(SR,SC)=XP;
646                                GC TO STR2;
647                             END;
648           SLZTEST:
649                 IF L= LK THEN GO TO STZEND;
650                           ELSE DC; ER=15; CALL ERRB(ER,I,EX); END;
654           STZEND:
                    PUT FILE(DISK1) EDIT(STR) (A(80));/*DISK1*/
655                 STR=SRR;
656                 GO TO START1;
657           CCNTROL:TP=0; CC=0; CR=CR+1; L=0; LK=10;
662           CNT1:I=I+1;IF CR>50 THEN DO;ER=24;CALL ERRB(ER,I,EX);GO TO START1;
668                                    END;
669                 XX=SUBSTR(FL,I,1);
670                 CALL CHAD(XX,I1,I2,A);
671                 IF I1 > 5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1;END;
677                 IF I = 80 THEN IF XX='X' THEN   DO;  CALL READ80(FL);I=0;
682                                                 GO TO CNT1;
683                        END; ELSE DO;ER=20;CALL ERRB(ER,I,EX);GO TO START1;END;
689                 IF I1¬= 1 THEN GO TO CNT1;
691           CNT2: I=I+1; XX=SUBSTR(FL,I,1); CALL CHAD(XX,I1,I2,A);
                 /*IF L=LK THEN GO TC CBZEND;*/
694                 IF I1 > 5 THEN DO;ER=3; CALL ERRB(ER,I,EX); END;
699                 IF I = 80 THEN IF XX ='X' THEN DO; CALL READ80(FL); I=0;
704                                                GO TO CNT2;
705                                          END;
706                                     ELSE DO; IF L >=LK THEN GO TO CBZEND;
709                                                     ELSE DO; ER=20;
711                /* NOT ALL FIELDS COMPLETE */             CALL ERRB(ER,I,EX);
712                                                   GO TO START1;
713                                                                END;
714                                          END;
```

```
715                 IF I1 = 1 THEN GO TO CNT2;
717                 J=I;
718                 JI=I ;
719                 L=L+1;
720                 IF L > LK THEN DO; ER=21;
723          /* TOO MANY FIELDS*/    CALL ERRB(ER,I,EX);
724                                 GO TO START1;
725                              END;
726                 IF I1 =2 | I1=5  THEN GO TO CNT3;
728                 IF I2 = 28 THEN DO; JI=J-1; GO TO CNT3; END;
733                 IF I1 = 4 THEN IF L>= LK THEN GO TO CBZEND;
736                                                ELSE DO; ER=80;
738                                           CALL ERRB(ER,I,EX);GO TO START1;
740                                                     END;
741                 IF I2 = 29  THEN GO TO CNT3;
743                            ELSE DO; ER=20;
745                                       CALL ERRB(ER,I,EX);
746           GO TO START1;       END;
748         CNT3: I=I+1;
749                 XX=SUBSTR(FL,I,1);
750                 CALL CHAD(XX,I1,I2,A);
751                 IF I1 > 5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
756                 IF I = 80 THEN DO; IF XX='X' THEN DO; CALL READ80(FI); I=0;
762                                                       GO TO CNT3;
763                                                     END;
764                              IF L >=LK THEN GO TO CBZEND;
766                          ELSE DO; ER=20; CALL ERRB(ER,I,EX); END; .
770                            END;
771                 IF I1 =2 | I1=5  THEN GO TO CNT3;
773                 IF I1 = 4 THEN IF L >=LK THEN GO TO CBZEND;
776                            ELSE DO;ER=20;CALL ERRB(ER,I,EX);GO TO START1;END;
781                 IF I1¬= 1 THEN DO; ER=1; CALL ERRB(ER,I,EX);GO TO START1;END;
787                 CC=CC+1; K=I-J; KI=I-JI; XP=SUBSTR(FL,JI,KI);
791                 IF L = 1 THEN DO;XJ=JI;
794           IF XP<CMN THEN CMN=XP; IF XP>CMX THEN CMX=XP;
798                                 CM=CM+1;
799                                 COBM(CM)=XP;
800                                 MX4(CR,CC)=XP;
801                                 KII=KI;
802                                 GO TO CNT2; END;
804                 IF L= 2  THEN DO; TP=XP;
807                                 IF TP = 1 THEN DO;
809                                    SUBSTR(AG9,22,KII)=SUBSTR(FL,XJ,KII);
810                                                 MX4(CR,CC)=TP;
811                                                  GO TO CNT2;
812                                               END;
813                              END;
814                 IF TP¬= 1
815                       THEN DO;
816                              IF TP=3|TP=14|TP=5|TP=6|TP=4 THEN LK=6;
818                             IF  TP=2  THEN LK=7;
820                              IF TP=7|TP=8|TP=9  THEN LK=9;
822                       IF TP=11 THEN LK=2;IF TP=12 | TP=13 | TP=10 THEN LK=4;
826                              IF TP>14 THEN DO;ER=22;CALL ERRB(ER,I,EX);
830          /* INVALID CNTBLK TYPE */          GO TO START1; END;
832                              IF L>LK THEN DO;ER=21;
835                                         CALL ERRB(ER,I,EX);
```

```
836                                          GO TO START1;
837                                       END;
838                                  ELSE DO;MX4(CR,CC)=XP;
840                                          GO TO CNT2;
841                                       END;
842                         END;
843              IF TP= 1
844                     THEN DO; LK=9;
846                   IF L=3 THEN DO;
848                        IF XP=1 THEN DO;IX=19; GT=1;XR=1;XS=XP;
854                                      GO TO CNT4;
855                                  END;
856                         IF XP=2 THEN DO;IX=19; GT=2;XR=2;XS=XP;
862                                       SUBSTR(GEN,IX,2)='FN';
863                                          IX=IX+2;
864                                       GO TO CNT4;
865                                    END;
866                         IF XP>2|XP=0 THEN DO; ER=23;
869                                         CALL ERRB(ER,I,EX);
870                                         GO TO START1;
871                                    END;
872                           END;
873              IF L=4 THEN DO;
875          IF XP<0 THEN DO;ER=27;CALL ERRB(ER,I,EX);GO TO START1;END;
881          IF XR=2 & XP<20 THEN DO;ER=26;CALL ERRB(ER,I,EX);GO TO START1;
886                                  END;
       /* XR=1 MEANS CONSTANT GEN TIME,THUS NEXT ARG (THE RANGE/2)MUST
          BE LESS THEN THE MEAN OF THE RECTANGULAR DISTR.*/
887                         SUBSTR(GEN,IX,KI)=SUBSTR(FL,JI,KI);
888                         IX=IX+KI;
889        IF GT=1 THEN DO;XS=XP;GO TO CNT4;END;
894         IF GT=2 THEN DO;CF=CF+1;GF(CF)=XP;GO TO CNT4;END;
900                         GO TO CNT4;
901                     END;
902              IF L=5 THEN DO;IF XR=1 & XS<XP THEN DO;ER=28;CALL ERRB(ER,I,EX);
908                                                     GO TO START1;END;
910                         GO TO CNT5;
911                     END;
912             IF  L=6 |
913                 L=7 THEN GO TO CNT5;
914              IF L=8 THEN DO;
916                          SUBSTR(GEN,IX,4)=',,20';
917                                    IX=IX+4;
918                          SUBSTR(AG11,23,KI)=SUBSTR(FL,JI,KI);
919                          GC TO CNT4;
920                     END;
921               IF L=9 THEN GO TO CNT4;
923              IF L > 9 THEN DC; ER=21;
926                           CALL ERRB(ER,I,EX);
927                            GO TO START1;
928                       END;
929                    ELSE GO TO CBZEND;
930              CNT5:
                       SUBSTR(GEN,IX,1)=',';
931                    IX=IX+1;
932                    SUBSTR(GEN,IX,KI)=SUBSTR(FL,JI,KI);
933                    IX=IX+KI;
```

```
 934                  CNT4:MX4(CR,CC)=XP; GO TO CNT2;
 936                               END;
 937          CBZEND:IF TP>1 THEN GO TO START;
 939                 SUBSTR(AG12,22,KI)=SUBSTR(FL,JI,KI);
 940                 PUT FILE(DISK4) EDIT(GEN) (A(80));  /*DISK4 */
 941                 PUT FILE(DISK4) EDIT(AG8) (A(80));
 942                 PUT FILE(DISK4) EDIT(AG9) (A(80));
 943                 PUT FILE(DISK4) EDIT(TRN) (A(80));
 944                 GEN=GEM; AG9=AD9;
 946                 GO TO START;
 947          QUEUE:
 948                 L=?; LK=3; QR=QR+1; QC=0 ;
 951                 IF QR >50 THEN DO; ER=30; CALL ERRB(ER,I,EX); END;
 956          QUE1:
                     I=I+1;
 957                 IF I=80  THEN DO; CALL READ80(FL); X=0;  GO TO QUE1; END;
 963                 XX=SUBSTR(FL,I,1);
 964                 CALL CHAD(XX,I1,I2,A);
 965                 IF I1>5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
 970                 IF I1¬=1 THEN GO TO QUE1;
 972          QUE2:I=I+1;
 973              IF SUBSTR(FL,I,81-I)=' ' THEN
 974                IF L=LK THEN GO TO START1;
 976                        ELSE DO;ER=31;CALL ERRB(ER,I,EX); GO TO START1; END;
 981                 XX=SUBSTR(FL,I,1);
 982                 CALL CHAD(XX,I1,I2,A);
 983                 IF I1 > 5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
 988                 IF I =80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
 993                                                      GO TO QUE2;
 994                                                  END;
 995                                             ELSE DO; L=L+1; GO TO QUEZEND; END;
 999                 IF I1=1 THEN GO TC QUE2;
1001                 IF I1=4 THEN GO TC QUEZEND;
1003              IF I1¬=2&I1¬=3 THEN DO;ER=1;CALL ERRB(ER,I,EX);GO TO START1;END;
1009               L=L+1; J=I; QC=QC+1;
1012               IF L>LK THEN DO;ER=31;CALL ERRB(ER,I,EX);GO TO START1;END;
1018          QUE3: I=I+1;
1019                 XX=SUBSTR(FL,I,1);
1020                 IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0;
1025                                                    GO TO QUE3;
1026                                                END;
1027                                           ELSE GO TO QUEZEND;
1028                 CALL CHAD(XX,I1,I2,A);
1029                 IF I1>5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
1034                 IF I1=2 THEN GO TO QUE3;
1036                 IF I1=4 THEN GO TO QUEZEND;
1038                 IF I1¬=1 THEN DO; ER=1; CALL ERRB(ER,I,EX); END;
1043           K=I-J; XP=SUBSTR(FL,J,K);
1045            IF L=1 THEN DO;IF XP>0 THEN XR=-XP;ELSE XR=XP;
1050                                QM=QM+1;
1051                                IF XR<QMN THEN QMN=XR;
1053                                  IF XR>QMX THEN QMX=XR;
1055                                 QUEM(QM)=XR;
1056                                 MX5(QR,QC)=XR;
1057                                  GO TO QUE2;
1058                               END;
1059            IF L =2 THEN DO;MX5(QR,QC)=XP;GO TO QUE2;END;
```

```
1064          IF L=3 THEN DO;MX5(QR,QC)=XP;
1067                   IF XP<0 & MX5(QR,QC-1)¬=1 THEN
1068                      DO;ER=32;CALL ERRB(ER,I,EX);
1071                        GO TO START1;
1072                      END;        ELSE GO TO QUE2;
1074                END;
1075       QUEZTEST:  QUEZEND:
1076           IF L = LK THEN GO TO STAR2;
1077                  ELSE DO;ER=31;CALL ERRB(ER,I,EX);GO TO START1;END;
1082       FACILITY:
1083           L=0; LK=5; FR=FR+1; FC=0;
1086           IF FR > 50 THEN DO; FR=40; CALL ERRB(ER,I,EX); END ;
1091       FAC1:      /* 700*/
               I=I+1;
1092           IF I =80 THEN DO;CALL READ80(FL); I=0; GO TO FAC1; END;
1098           XX=SUBSTR(FL,I,1);
1099           CALL CHAD(XX,I1,I2,A);
1100           IF I1 >5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1;END;
1106           IF I1¬=1 THEN GO TO FAC1;
1108       FAC2:I=I+1;
1109          IF SUBSTR(FL,I,81-I)=' '
1110              THEN IF L=LK THEN GO TO START1;
1112                     ELSE DO;ER=41;CALL ERRB(ER,I,EX); GO TO START1;END;
1117           XX=SUBSTR(FL,I,1);
1118           CALL CHAD(XX,I1,I2,A);
1119           IF I1 >5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
1124           IF I=80 THEN IF XX='X' THEN DO; CALL READ80(FL); I=0; GO TO FAC2;
1130                                      END;
1131                                ELSE DO; GO TO FACZTEST; END;
1134           IF I1=1 THEN GO TO FAC2;
1136           IF I1=4 THEN GO TO FACZTEST;
1138           IF I1¬=2&I1¬=3 THEN DO;ER=1;CALL ERRB(ER,I,EX);GO TO START1;END;
1144           L=L+1; J=I; FC=FC+1;
1147           IF L>LK THEN DO;ER=42;CALL ERRB(ER,I,EX);GO TO START1;END;
1153       FAC3:
               I=I+1;
1154           XX=SUBSTR(FL,I,1);
1155           CALL CHAD(XX,I1,I2,A);
1156           IF I1 >5 THEN DO; ER=3; CALL ERRB(ER,I,EX); END;
1161           IF I =80 THEN IF  XX='X' THEN DO; CALL READ80(FL); I=0;
1166                                          GO TO FAC3;
1167                                      END;
1168                                 ELSE GO TO FACZTEST;
1169           IF I1 =2 THEN GO TO FAC3;
1171           IF I1 =4 THEN GO TO FACZTEST;
1173           IF I1¬=1 THEN DO; ER=1; CALL ERRB(ER,I,EX); END;
1178                  ELSE DO;
1179                         K=I-J;
1180                        XP= SUBSTR(FL,J,K);
1181        IF L<4 & XP<0 THEN DO;ER=43;CALL ERRB(ER,I,EX);GO TO START1;END;
1187        IF L=5 & MX2(FR,FC-1)¬=1 & XP<0
1188                  THEN DO;ER=43;CALL ERRB(ER,I,EX);GO TO START1;END;
1193        IF L=1 THEN DO;FM=FM+1;IF XP>FMX THEN FMX=XP;GO TO FAC4;END;
           /* IF L=2 3 4 5 PASS THRU TO FAC4  */
1200        FAC4:MX2(FR,FC)=XP; GO TO FAC2;
1202       FACZTEST:
1203           IF L = LK THEN GO TO STAR2;
```

```
1204          IF L<LK THEN DO;ER=41;CALL ERRB(ER,I,EX); GO TO START1;END;
1210          IF L>LK THEN DO;ER=42;CALL ERRB(ER,I,EX);GO TO START1;END;
1216      VARIABLE: /*ISOLATE VARIABLE NUMBER */  VAR=' '; T=0;
1218       IV=0; BR=0; I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1223       IF I1=3| I1=4 | I1=5| I1=1 THEN DO;ER=60;CALL ERRB(ER,I,EX);
1227                                               GO TO START1; END;
1229       J=I;                                         /*1700*/
1230      VAR1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1233          IF I1=2 THEN GO TO VAR1;
1235      VAR2:K=I-J; SUBSTR(VAR,7,K+4)='V(' || SUBSTR(FL,J,K) || ')=';
1237          IV=7+K+4; XP=SUBSTR(FL,J,K);
1239         DO L=1 TO LV; IF (XP=VR(L)) THEN DO;ER=61;CALL ERRB(ER,I,EX);
1244                                                   GO TO START1;END;
1246         END;
1247         LV=LV+1; VR(LV)=XP;
1249         IF LV>19 THEN DO;ER=62;CALL ERRB(ER,I,EX);GO TO START1;END;
1255       IF I2¬=12 THEN DO;ER=63;CALL ERRB(ER,I,EX);GO TO START1;END;
1261      VAR5:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A); /*1800*/
1264      VAR3: /* IDENTIFY VARIABLES,OPERATORS AND SPECIAL CHARACTERS  */
1265          IF IV >70 THEN DO;ER=64;CALL ERRB(ER,I,EX);GO TO START1;END;
1270          IF I1 =4 THEN GO TC ALPHA;
1272          IF I1=2 THEN GO TO NUMERIC;
1274         IF I1=5 THEN GO TO NUMA;
1276         IF I2=3|I2=4|I2=28|I2=29 THEN GO TO OPERATOR;
1278          IF I2=5 THEN GO TO BRACKETO;
1280          IF I2=6 THEN GO TO BRACKETC;
1282      VAREND:IF BR¬=0 THEN DO; ER=65;CALL ERRB(ER,I,EX);END;
1287            IF T=0 THEN DO;ER=66;CALL ERRB(ER,I,EX);END;
1292            IF BR¬=0|T=0 THEN GO TO START;  /*1900*/
1294           PUT FILE(DISK5) EDIT(VAR) (A(80));
1295           GO TO START;
1296      ALPHA: /*FIND FULL EXTENT OF WORC AND ANALYSE */
             J=I;
1297      ALPH1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1300         IF I1=4 THEN GO TC ALPH1;
1302         IF I1¬=2 THEN DO;ER=76;CALL ERRB(ER,I,EX);GO TO START1;END;
1308         K=I-J;
1309         IF K=1 THEN DO;
1311                  IF SUBSTR(FL,J,K)='C' THEN GO TO VCNT;
1313                   IF SUBSTR(FL,J,K)='P' THEN GO TO VPAR;   /*2000*/
1315                   IF SUBSTR(FL,J,K)='V' THEN GO TO VVAR;
1317                                       ELSE DO;ER=67;CALL ERRB(ER,I,EX);
1320                                               GO TO START1;END;
1322                     END;
1323          IF K=2 THEN DO; IF SUBSTR(FL,J,K)='LN' THEN GO TO VLN;
1327                   ELSE DO;FR=68;CALL ERRB(ER,I,EX);GO TO START1;END;
1332                    END;
1333        IF K=3 THEN DO;IF SUBSTR(FL,J,K)='EXP'|SUBSTR(FL,J,K)='LOG'|
                             SUBSTR(FL,J,K)='SIN'|SUBSTR(FL,J,K)='COS' |
1336                     SUBSTR(FL,J,K)='TAN' THEN GO TO VSFN;
1337           ELSE DO;ER=69;CALL ERRB(ER,I,EX);GO TO START1;END;
1342                   END;
1343         IF K=4 THEN DO;IF SUBSTR(FL,J,K)='SQRT' | SUBSTR(FL,J,K)='TANH'
             | SUBSTR(FL,J,K)='SINH' | SUBSTR(FL,J,K)='COSH' |
1346          SUBSTR(FL,J,K)='ATAN' THEN GO TO VSFN;
1347          ELSE DO;ER=70;CALL ERRB(ER,I,EX);GO TO START1;END;
1352                   END;
```

```
1353          IF K=5 THEN DO;IF SUBSTR(FL,J,K)='ARSIN' | SUBSTR(FL,J,K)='ARCOS'
1356         | SUBSTR(FL,J,K)='COTAN' THEN GO TO VSFN;
1357          ELSE DO;ER=71;CALL ERRB(ER,I,EX);GO TO START1;END;
1362               END;
        /* INSERT CODE FOR VALID ALPHA WORDS LONGER THEN 5*/
1363                    ELSE DO;ER=72;CALL ERRB(ER,I,EX);GO TO START1; END;
1368    VLN:IF T¬=0 THEN DO;ER=73;CALL ERRB(ER,I,EX);GO TO START1;END;
1374           IF I2¬=5 THEN DO;ER=74;CALL ERRB(ER,I,EX);
1378                                    GO TO START1;END;
1380          BR=BR+1; SUBSTR(VAR,IV,5)='ALOG('; IV=IV+5; GO TO VAR5;
1384   VSFN: IF T¬=0 THEN DO;ER=73;CALL ERRB(ER,I,EX);GO TO START1;END;
1390         IF I2¬= 5  THEN DO;ER=74;CALL ERRB(ER,I,EX);
1394                                         GO TO START1;END;
1396          BR=BR+1;IF SUBSTR(FL,J,K)='LOG'
1398              THEN DO;SUBSTR(VAR,IV,7)='ALOG10(';IV=IV+7;END;
1402               ELSE DO;SUBSTR(VAR,IV,K+1)=SUBSTR(FL,J,K+1);IV=IV+K+1;END;
1406          GO TO VAR5;
1407   VCNT: /*IDENTIFY COUNTER NUMBER  */
1408           IF T¬=0 THEN DO;ER=75;CALL ERRB(ER,I,EX);GO TO START1;END;
1413                ELSE T=1;
1414           IF I1¬=2 THEN DO;ER=76;CALL ERRB(ER,I,EX);GO TO START1;END;
1420         J=I;
1421   VCNT1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1424          IF I1=2 THEN GO TO VCNT1;
1426          K=I-J; XP=SUBSTR(FL,J,K);
1428         IF XP<1 | XP>30 THEN DO;ER=77;CALL ERRB(ER,I,EX);GO TO START1;END;
1434          SUBSTR(VAR,IV,K+6)= 'X(' || SUBSTR(FL,J,K) ||'+20)';
1435           IV=IV+K+6; GO TO VAR3;
1437   VPAR: /*IDENTIFY PARAMETER NUMBER  */
1438        IF T¬=0 THEN DO;ER=75;CALL ERRB(ER,I,EX);GO TO START1;END;
1443                ELSE T=1;                  /*2200*/
1444         IF I1¬=2 THEN DO;ER=76;CALL ERRB(ER,I,EX);GO TO START1;END;
1450         J=I;
1451   VPAR1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1454         IF I1=2 THEN GO TO VPAR1;
1456          K=I-J; XP=SUBSTR(FL,J,K);
1458          IF XP<1 | XP>5 THEN DO;ER=78;CALL ERRB(ER,I,EX);GO TO START1;END;
1464         SUBSTR(VAR,IV,K+3)='X(' || SUBSTR(FL,J,K) || ')';
1465        IV=IV+K+3;  GO TO VAR3;
1467   VVAR: /* IDENTIFY VARIABLE NUMBER  */
1468          IF T¬=0 THEN DO;ER=75;CALL ERRB(ER,I,EX);GO TO START1;END;
1473                  ELSE T=1;
1474         J=I;
1475   VVAR1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1478         IF I1=2 THEN GO TO VVAR1;
1480          K=I-J;  XP=SUBSTR(FL,J,K);ILV=ILV+1; VVR(ILV)=XP;
1484           SUBSTR(VAR,IV,K+3)= 'V(' || SUBSTR(FL,J,K) || ')';
1485          IV=IV+K+3; GO TO VAR3;
1487   NUMA:PD=1;
1488   NUMERIC: /*CONSTANT-NUMERIC */
1489          IF T¬=0 THEN DO;ER=79;CALL ERRB(ER,I,EX);GO TO START1;END;
1494                  ELSE T=1;               /*2400*/
1495          J=I;
1496   NUM1:I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);
1499     IF I1=2 THEN GO TO NUM1; IF I1=5 THEN IF PD=0 THEN GO TO NUM1;
1504                  ELSE DO;ER=80;CALL ERRB(ER,I,EX);GO TO START1;END;
1509   PD=0; K=I-J; SUBSTR(VAR,IV,K)=SUBSTR(FL,J,K); IV=IV+K; GO TO VAR3;
```

```
1514          OPERATOR:IF T¬=1 THEN DO;ER=81;CALL ERRB(ER,I,EX);GO TO START1;END;
1520               J=I; I=I+1;XX=SUBSTR(FL,I,1);CALL CHAD(XX,I1,I2,A);T=0;
1525               IF XX='*' & SUBSTR(FL,J,1)='*' THEN DO;I=I+1;XX=SUBSTR(FL,I,1);
1529               K=2; CALL CHAD(XX,I1,I2,A);GO TO OPER1;END;K=I-J;
1534               IF I1=4|I1=2 THEN GO TO OPER1;
1536               IF I2=3|I2=28|I2=4|I2=29 THEN DO;ER=82;CALL ERRB(ER,I,EX);
1540                                               GO TO START1;END;
1542          OPER1:SUBSTR(VAR,IV,K)=SUBSTR(FL,J,K);
1543               IV=IV+K; GO TO VAR3;
1545          BRACKETO:/*OPEN BRACKET MUST FOLLOW AN OPERATOR OR AN OPEN BRACKET */
1546             IF T¬=0 THEN DO;ER=83;CALL ERRB(ER,I,EX);GO TO START1;END;
1551            SUBSTR(VAR,IV,1)='(';  IV=IV+1; BR=BR+1; GO TO VAR5;
1555          BRACKETC:/*CLOSED BRACKET MUST FOLLOW A VARIABLE OR CLOSED BRACKET*/
1556            IF T¬=1 THEN DO;ER=84;CALL ERRB(ER,I,EX);GO TO START1;END;
1561               SUBSTR(VAR,IV,1)=')'; IV=IV+1; BR=BR-1; GO TO VAR5;
1565          DECISION:DR=DR+1;DC=0;L=0;XR=0;       /*2600*/
1569               IF DR>19 THEN DO;ER=90;CALL ERRB(ER,I,EX);GO TO START1;END;
1575          DEC1:I=I+1;
1576                IF I=80 THEN DO;CALL READ80(FL);I=0;GO TO DEC1;END;
1582                XX=SUBSTR(FL,I,1); CALL CHAD(XX,I1,I2,A);
1584                IF I1>5 THEN DO;ER=3;CALL ERRB(ER,I,EX); GO TO START1;END;
1590                IF I1¬=1 THEN GO TO DEC1;
1592          DEC2:I=I+1;
1593                IF SUBSTR(FL,I,80-I)=' '  THEN GO TO START1;
1595                XX=SUBSTR(FL,I,1); CALL CHAD(XX,I1,I2,A);
1597                IF I=80 THEN IF XX='X' THEN DO;CALL READ80(FL);I=0;GO TO DEC2;
1603                                              END;
1604                                      ELSE GO TO START1;
1605                IF I1=1 THEN GO TO DEC2;
1607                IF I1=4 THEN GO TO STAR2;
1609                IF I1¬=2&I1¬=3 THEN DO;ER=1;CALL ERRB(ER,I,EX);GO TO START1;END;
1615          J=I; L=L+1;DC=DC+1;IF DC>50 THEN DO;ER=91;CALL ERRB(ER,I,EX);
1622                                               GO TO START1; END;
1624          DEC3:I=I+1;XX=SUBSTR(FL,I,1);
1626                IF I=80 THEN IF XX='X' THEN DO;CALL READ80(FL); I=0;GO TO DEC3;
1632                                              END;       /*2800*/
1633                                      ELSE GO TO START1;
1634                CALL CHAD(XX,I1,I2,A);
1635                IF I1>5 THEN DO;ER=3;CALL ERRB(ER,I,EX);GO TO START1; END;
1641                IF I1=2 THEN GO TO DEC3;
1643                IF I1¬=1 THEN DO;ER=1;CALL ERRB(ER,I,EX);GO TO START1;END;
1649                          ELSE DO;K=I-J;
1651                                  XP=SUBSTR(FL,J,K);
1652                         IF DC=2 THEN DO;IF (XP>=6) & (XP<=8) THEN GO TO DEC4;
1656                                                                ELSE DO;
1657                                    ER=92;CALL ERRB(ER,I,EX);GO TO START1;END;
1661                                  END;
1662           IF DC=3 THEN DO;
1664                        IF XR=6THEN
1665                                  IF XP>0 & XP<4 THEN GO TO DEC4;
1667                                              ELSE DO;ER=93;GO TO ERDD;END;
1671                         IF XR=7 THEN
1672                                    IF XP>0 & XP<5 THEN GO TO DEC4;
1674                                                ELSE DO;ER=94;GO TO ERDD;END;
1678                        IF XR=8 THEN
1679                                  IF XP>0 & XP<3 THEN GO TO DEC4;
1681                                              ELSE DO;ER=95;GO TO ERDD;END;
```

```
1685                     ELSE ER=92;
1686         ERDD:CALL ERRB(ER,I,EX);GO TO START1;
1688                   END;
1689                      DEC4:MX6(DR,DC)=XP;XR=XP;
1691                           GO TO DEC2;
1692          EDIT:
               /*TEST IF ALL REFERENCED VARIABLES ARE DEFINED */
             DO I=1 TO ILV;
1693         DO J=1 TO LV;
1694           IF VVR(I)=VR(J) THEN GO TO END1;
1696         END;
1697          DO;ER=85;K=VVR(I);CALL ERRA(ER,K,EX);END;
1702         END1:END;
1703          END2: /* TEST IF VARIABLES ARE SEQUENTIALLY INCLUSIVE  */
1704        DO I = 1 TO LV;  DO J = 1 TO LV;
1705          IF VR(J)=I THEN GO TO END3;    END;
1708         ER=86;CALL ERRA(ER,I,EX);GO TO END4;
1711         END3:END;
1712         END4:
             END6:/*TEST IF FUNCTIONS REFERENCED EXIST*/
1713          DO I = 1 TO CF; DO J=1 TO FI;IF GF(I)=MX6(20,J) THEN GO TO END7;
1716        END;K=GF(I);ER=26;CALL ERRA(ER,K,EX);GO TO END8;
1721         END7:END;
1722         END8:/* TEST IF ENTITY NUMBERING OVERLAPS */
1723       IF QMX>SMN THEN DO;ER=17;I=SMX;CALL ERRA(ER,I,EX);END;
1728        IF FMX>CMN THEN DO;ER=44;I=FMX;CALL ERRA(ER,I,EX);END;
1734            PUT FILE(SYSPRINT) EDIT(' ') (PAGE,A(1));
1735            DCL ERSMT CHAR(80) INIT(' ');
1736            IF EX=0 THEN SUBSTR(ERSMT,10,24)='SIMULATION IS CONTINUED.';
1738                    ELSE SUBSTR(ERSMT,10,28)='SIMULATION IS NOT CONTINUED.';
1739       PUT FILE(SYSPRINT) EDIT('THERE ARE ',EX,' ERRORS IN THE EDIT PHASE.')
             (A(10),F(4),A(26));
1740          PUT FILE(SYSPRINT) EDIT(ERSMT) (SKIP,A(80));
1741        IF EX>0 THEN DO;  CALL THESARC(EX);  GO TO END20;END;
1746          /* CONTROL VECTOR ASSIGNMENT     */    L=1;
1747       COBM(*)=0; COBM(1)=QMN; COBM(2)=SMN; COBM(3)=FMX; COBM(4)=CMX;
1752       COBM(5)=RR; COBM(6)=FR; COBM(7)=SR; COBM(8)=CR; COBM(9)=QR;
1757          COBM(10)=DR;COBM(11)=LV;
1759       J=50; PUT FILE(DISK6) EDIT(L,J,(COBM(I) DO I=1 TO 50))(52F(4));
1761        /* FACILITY MATRIX   */ L=3;  J=5;
1763           DO I=1 TO FR;
1764             PUT FILE(DISK6) EDIT(L,J,(MX2(I,K) DO K=1 TO J)) (SKIP,7F(4));
1765            END;
1766         /*STORAGE  MATRIX  */   J=9;L=4;
1768          DO I =1 TO SR;
1769       PUT FILE(DISK6) EDIT(L,J,(MX3(I,K) DO K=1 TO J)) (SKIP,14F(4));
1770           END;
1771          /*CONTROL BLOCK MATRIX      */    J=9; L=5;
1773        DO I=1 TO CR;
1774         PUT FILE(DISK6) EDIT(L,J,(MX4(I,K) DO K=1 TO J))(SKIP,11F(4));
1775         END;
1776          /*    QUEUE MATRIX     */L=6; J=3;
1778         DO I=1 TO QR;
1779        PUT FILE(DISK6) EDIT(L,J,(MX5(I,K) DO K=1 TO J)) (SKIP,5F(4));
1780           END;
1781        /*DECISION RULES */ J=20;  L=7;
1783         DO I =1 TO 20; PUT FILE(DISK6) EDIT(L,J,(MX6(I,K) DO K=1 TO 20))
```

3040

```
1785            (SKIP,22F(4)); END;
1786      /*END OF SUBR SIMT02 */ VR1=' ';
1787       SUBSTR(VR1,7,11)='X(13)=V(NV)'; PUT FILE(DISK5) EDIT(VR1) (A(80));
1789      VR1=' ';SUBSTR(VR1,7,12)='DO 2 I=1,100';
1791       PUT FILE(DISK5) EDIT(VR1) (A(80));VR1=' ';
1793      SUBSTR(VR1,5,11)='2 A(I)=X(I)';PUT FILE(DISK5) EDIT(VR1) (A(80));
1795      VR1=' '; SUBSTR(VR1,7,6)='RETURN';PUT FILE(DISK5) EDIT(VR1) (A(80));
1798      VR1=' '; SUBSTR(VR1,7,3)='END';PUT FILE(DISK5) EDIT(VR1)(A(80));
          /* DEFINE NUMBER OF STORAGES TO GPSS  */
1801       VR1=' '; SUBSTR(VR1,8,7)='INITIAL'; SUBSTR(VR1,19,4)='X68,';
1804       K=1;IF SR>9 THEN K=2; C8=SR;
1808        SUBSTR(VR1,23,K)=SUBSTR(C8,10-K,K);
1809       PUT FILE(DISK1) EDIT(VR1) (A(80));
1810      CLOSE FILE(SYSPRINT),FILE(SYSIN),FILE(DISK1),FILE(DISK2),
                FILE(DISK3),FILE(DISK4),FILE(DISK5),FILE(DISK6);
1811      END20:END MAIN;
```

APPENDIX H

EXECUTION JOB CONTROL

STATEMENTS

```
//SAM6E2 JOB '0006,XXXX,1,T=3,I=10,L=1,C=0,R=110K','PETR',MSGLEVEL=2      JOB 976
//JOBLIB DD DSN=PISIM.A3540,VOL=SER=UM1404,UNIT=2314,DISP=SHR
//   EXEC PGM=SIM02,REGION=110K
//GO.SYSPRINT DD SYSOUT=A
//GO.DISK1 DD DSN=SAM61.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5)),DISP=(,KEEP),
//   DCB=(RECFM=F,LRECL=80,DSORG=PS)
//GO.DISK2 DD DSN=SAM62.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5)),DISP=(NEW,KEEP),
//   DCB=(RECFM=F,LRECL=80,DSORG=PS)
//GO.DISK3 DD DSN=SAM63.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5)),DISP=(NEW,KEEP),
//   DCB=(RECFM=F,LRECL=80,DSORG=PS)
//GO.DISK4 DD DSN=SAM64.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5)),DISP=(NEW,KEEP),
//   DCB=(RECFM=F,LRECL=80,DSORG=PS)
//GO.DISK5 DD DSN=SAM65.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5)),DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//GO.DISK6 DD DSN=SAM66.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//   SPACE=(TRK,(2,5),RLSE),DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=208,BLKSIZE=2080)
//GO.SYSIN DD *
IEF373I STEP /        / START 71168.1345
IEF374I STEP /        / STOP  71168.1345 CPU    0MIN 01.65SEC MAIN  98K LCS    0K
             28.24 SEC EXEC TIME      1.65 SEC CPU TIME      306 I/O COUNTS 110K REGION  98K USED

IEF375I  JOB /SAM6E2  / START 71168.1345
IEF376I  JOB /SAM6E2  / STOP  71168.1345 CPU    0MIN 01.65SEC
SAM6E2       28.24 SEC EXEC TIME      1.65 SEC CPU TIME         DATE 71.168 UNIVERSITY OF MANITOBA

 HASP-II JOB STATISTICS --      39 CARDS READ --       63 LINES PRINTED --       0 CARDS PUNCHED --    0.47 MINUTES EXECUTION TIME
 UNIVERSITY OF MANITOBA --     323 I/O COUNTS --    1,042 K BYTE-SECONDS -     0.85 UNITS        --    0.02 MINUTES CPU TIME
 ONLINE ACCOUNTING INFO --     185 JOB(S) RUN --
```

```
//JPSM2   JOB '0086,XXXX,1,T=9,L=1,I=20,R=50,C=0','PETR',MSGLEVEL=1          JOB 991
//JOBLIB DD DSN=POSIM.A3540,VOL=SER=UM1404,UNIT=2314,DISP=SHR
//    EXEC PGM=SIM02
//GO.SYSPRINT DD SYSOUT=A
//GO.DISK1 DD DSN=PESIM91.A3540,VOL=SER=UM1404,UNIT=2314,
//    DISP=(NEW,KEEP),SPACE=(TRK,(5,5)),DCB=(RECFM=F,LRECL=80,DSORG=PS)
//GO.SYSIN DD DSN=PESIM40.A3540,VOL=SER=UM1404,DISP=(OLD,KEEP),
//    UNIT=2314
//   DD  DSN=SAM61.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//      DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80,DSORG=PS)
//    DD  DSN=SAM63.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//        DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80,DSORG=PS)
//    DD  DSN=PESIM50.A3540,VOL=SER=UM1404,UNIT=2314,
//      DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80)
//    DD  DSN=SAM64.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//      DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80,DSORG=PS)
//    DD  DSN=PESIM60.A3540,VOL=SER=UM1404,UNIT=2314,
//       DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80)
//
IEF373I STEP /        / START 71168.1350
IEF374I STEP /        / STOP  71168.1351 CPU    0MIN 01.49SEC MAIN  34K LCS    OK
            88.25 SEC EXEC TIME      1.49 SEC CPU TIME     953 I/O COUNTS  52K REGION  34K USED

IEF375I  JOB /JPSM2   / START 71168.1350
IEF376I  JOB /JPSM2   / STOP  71168.1351 CPU   0MIN 01.49SEC
JPSM2       88.25 SEC EXEC TIME      1.49 SEC CPU TIME        DATE 71.168 UNIVERSITY OF MANITOBA


  HASP-II JOB STATISTICS --     20 CARDS READ --       33 LINES PRINTED --      0 CARDS PUNCHED --   1.47 MINUTES EXECUTION TIME
  UNIVERSITY OF MANITOBA --    970 I/O COUNTS --    1,315 K BYTE-SECONDS -     1.86 UNITS        --   0.02 MINUTES CPU TIME
  ONLINE ACCOUNTING INFO --    188 JOB(S) RUN --
```

```
//JPSM6E2 JOB '0085,XXXX,1,T=50,R=220K,I=70,L=3,C=0','PETR',MSGLEVEL=2   JOB 997
// EXEC ASMFC
//ASM.SYSIN DD DSN=PESIM10.A3540,VOL=SER=UM1404,UNIT=2314,
//  DISP=(OLD,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
IEF373I STEP /ASM      / START 71168.1354
IEF374I STEP /ASM      / STOP  71168.1354 CPU    0MIN 00.90SEC MAIN  52K LCS    OK
ASM          23.45 SEC EXEC TIME      0.90 SEC CPU TIME      327 I/O COUNTS  52K REGION  52K USED

// EXEC FORTGCL
//FORT.SYSIN DD DSN=PESIM21.A3540,VOL=SER=UM1404,UNIT=2314,
//    DISP=(OLD,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//    DD  DSN=SAM65.A3540,VOL=REF=ONE.MONTH,UNIT=2314,
//      DISP=(OLD,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
IEF373I STEP /FORT     / START 71168.1354
IEF374I STEP /FORT     / STOP  71168.1357 CPU    0MIN 17.44SEC MAIN 104K LCS    OK
FORT        129.71 SEC EXEC TIME     17.44 SEC CPU TIME      243 I/O COUNTS 104K REGION 104K USED

//LKED.SYSLMOD DD DSN=&USER(TEST),DISP=(NEW,PASS),UNIT=SYSDA,
//        SPACE=(TRK,(5,50,1))
IEF373I STEP /LKED     / START 71168.1357
IEF374I STEP /LKED     / STOP  71168.1357 CPU    0MIN 00.92SEC MAIN  96K LCS    OK
LKED         31.27 SEC EXEC TIME      0.92 SEC CPU TIME      408 I/O COUNTS  96K REGION  96K USED

//  EXEC GPSS360,REGION=220K
//GPSS.STEPLIB DD
//    DD DSN=&USER,DISP=(OLD,DELETE)
//GPSS.FT06F001 DD SYSOUT=A
//GPSS.FT09F001 DD DSN=SAM66.A3540,UNIT=2314,DISP=(OLD,KEEP),
//    DCB=(RECFM=FB,LRECL=208,BLKSIZE=2080),VOL=REF=ONE.MONTH
//GPSS.SYSABEND DD SYSOUT=A
//GPSS.DINPUT1 DD DSN=PESIM91.A3540,VOL=SER=UM1404,UNIT=2314,
//    DISP=(OLD,KEEP),DCB=(RECFM=F,LRECL=80)
IEF373I STEP /GPSS     / START 71168.1357
IEF374I STEP /GPSS     / STOP  71168.1403 CPU    0MIN 21.82SEC MAIN 212K LCS    OK
GPSS        368.87 SEC EXEC TIME     21.82 SEC CPU TIME     4932 I/O COUNTS 220K REGION 212K USED

IEF375I  JOB /JPSM5E2 / START 71168.1354
IEF376I  JOB /JPSM5E2 / STOP  71168.1403 CPU    0MIN 41.08SEC
JPSM6E2     553.30 SEC EXEC TIME     41.08 SEC CPU TIME        DATE 71.168 UNIVERSITY OF MANITOBA


 HASP-II JOB STATISTICS --      24 CARDS READ --    2,637 LINES PRINTED --      0 CARDS PUNCHED --   9.22 MINUTES EXECUTION TIME
 UNIVERSITY OF MANITOBA --   5,941 I/O COUNTS --   38,986 K BYTE-SECONDS -    21.68 UNITS         --   0.68 MINUTES CPU TIME
 ONLINE ACCOUNTING INFO --     189 JOB(S) RUN --
```

BIBLIOGRAPHY

1. Goldberg M.,Mittman B., "SPURT - A Simulation Package for University Research and Teaching " Paper presented to the Second Conference on Applications of Simulation, New York, New York, December 2-4, 1968.

2. Kiviat P.J. "Digital Computer Simulation : Modeling Concepts " RAND Memorandum RM-5378-PR August 1967 Page 19-22.

3. International Business Machines, "General Purpose Simulation System/360, User's Mannal " H20-0326-2

4. Dalkey N.C. "Operations Research" RAND Memorandum P-3705 October 1967

5. Churchman C.W., Ackoff R.L., Arnoff E.L. "Introduction to Operations Research" John Wiley and Sons, Inc. 1957 September 1964 8th printing

6. Peterson J.E. "Generalized GPSS Model for Grain Terminal Elevators " Paper presented to the Second Conference on Applications of Simulation , New York, New York, December 2-4, 1968

7. Steel R.G.D.,Torrie J.H. "Principles and Procedures Statistics ", McGraw-Hill Book Company, 1960

8. di Roccaferrera G.M.F. "Operations Research Models for Business and Industry",South-Western Publishing Company,1964