DEVELOPMENT OF A DYNAMIC SIMULATION MODEL FOR PLANNING PHYSICAL DISTRIBUTION SYSTEMS: FORMULATION OF THE COMPUTER MODEL

Thesis for the Degree of Ph. D.

MICHIGAN STATE UNIVERSITY

EDWARD JOSEPH MARIEN

1970

This is to certify that the

thesis entitled

DEVELOPMENT OF A DYNAMIC SIMULATION MODEL

FOR PLANNING PHYSICAL DISTRIBUTION SYSTEMS:

FORMULATION OF THE COMPUTER MODEL

presented by

Edward Joseph Marien

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in **Business**

Donald J Bowersox
Major professor

Date Nov 10, 1970

O-7639

ABSTRACT

DEVELOPMENT OF A DYNAMIC SIMULATION MODEL FOR
PLANNING PHYSICAL DISTRIBUTION SYSTEMS:
FORMULATION OF THE COMPUTER MODEL

By

Edward Joseph Marien

To provide consumers with a wide assortment of goods at the right time, in the right place and at the highest profit potential, managers have increasingly viewed their product distribution systems on an integrative basis. Total integrated costs of transportation, warehousing, holding and handling inventories, and communications can be balanced against requirements of consumers for fast and consistent service.

To achieve an effective and efficient balance between cost and customer service, product or physical distribution management is responsible for designing and administering systems to control finished goods flow and, in some cases, the flow of raw materials. The problem is how to achieve the correct balance in a changing business environment.

Given the above general class of problem, a specific need exists for the development of improved planning models to aid firms in the design of total physical distribution

systems. Such a planning model should be capable of assisting managers to achieve their objectives by determining the best points in time to implement physical distribution system modifications.

The overall objective of an ongoing research project at the Graduate School of Business, Michigan State University, is to provide such a physical distribution planning model. The model has been titled Long-Range Environmental Planning Simulator (LREPS). The research is being conducted by faculty and doctoral students. This dissertation deals with the computerization of the model on the Control Data Corporation 6500 computer using primarily the GASP IIA simulation language and FORTRAN IV.

In formulating the computerized model, an overall approach was required. This approach is discussed in the Methodology Section of Chapter I. Various programming and systems techniques also had to be reviewed for their applicability in the computerization process. These techniques, such as the use of Monte Carlo selection, pseudo-random number generation and statistical estimation procedures, are discussed as the need arises.

The result of the activities associated with this dissertation is an operational computer model. Information systems concepts were initially utilized to segregate the activities of the computer model into supporting data, operating and report generator systems. In relation to these systems, a total data base of information was defined. Data base variables were assigned mnemonics and dimensions.

After developing preliminary specifications of the computer model's subprograms and data base of information, the GASP IIA simulation language was selected for predominantly implementing the model's operating system. A set of criteria was developed for language selection. FORTRAN IV and the CDC 6500 assembly programming languages are used on a supportive basis.

In programming the activities of the mathematical model, basic building-block subprograms facilitated the overall development of the computer model. These basic building blocks are organized and discussed in relation to the four major subsystems of the computer model. The four subsystems are: (1) the Demand and Environment Subsystem for generating and allocating customer demand, (2) the Operations Subsystem for processing simulated customer orders through the physical distribution system configuration being tested, (3) the Measurement Subsystem for developing the criteria for evaluating the system configuration and (4) the Monitor and Control Subsystem for supervising and controlling the operation of the computer simulation model. In implementing these subsystems on the computer, magnetic tape, punched cards and listings are the primary means of model input and output.

The computer model is presently dependent upon MSU's large-scale CDC 6500 computer operating system. The conversion of the model to another computer operating system would be very difficult unless a sophisticated FORTRAN compiler that would include EXTENDED FORTRAN routines for tape buffering, shifting and logical instructions plus large amounts of computer memory were available. Packed computer words of memory, which are sixty-bit words on the CDC 6500 and which, in many cases, contain up to fifty pieces of information per computer word, make the possible conversion process even more difficult.

The operating system of the present computer model requires a little less than thirty-two thousand decimal, computer words of memory. Of this required amount of computer memory, the data base of information occupies seventeen thousand two hundred words. The execution time is primarily a function of a firm's sales dollar forecast and average sales dollars per customer order. Considering the consumer-oriented firms who might use LREPS, the computer run times per maximum ten-year simulation cycle might be unreasonable if they have large sales forecasts and small customer orders.

Preliminary validation of model results for one year's sales history plus initial, experimental use of the model for assisting the management of the project's industrial research supporter indicate the model is sound and reliable.

DEVELOPMENT OF A DYNAMIC SIMULATION MODEL FOR PLANNING PHYSICAL DISTRIBUTION SYSTEMS: FORMULATION OF THE COMPUTER MODEL

By

Edward Joseph Marien

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Marketing and Transportation

1970

67287

To my lovely wife, Janet

ACKNOWLEDGEMENTS

Since a Michigan State University research team developed LREPS, acknowledgements are given to team members Dr. Donald J. Bowersox, our faculty director, and doctoral candidates, O. K. Helferich, R. T. Rogers, M. L. Lawrence, V. K. Prasad, P. Gilmour and F. W. Morgan, Jr. They assisted not only in conceptualizing the alternative methods for accomplishing certain computer tasks but also in working out the details.

The Michigan State University Computer Laboratory was a very helpful source of information and assistance in the development of the various procedures concerning the use of their available computer operating systems. I am also particularly grateful to Dr. Charles F. Wrigley, Director of the Computer Institute of Social Science Research, who provided a large source of programming personnel from which part of the LREPS pool of programmers was drawn.

The industrial research supporter must be acknowledged for his kind consideration and assistance. Much of the data for the model was prepared on their computer systems which alleviated much work at MSU. They also greatly assisted in conceptualizing and implementing the system model.

My doctoral committee was composed of Dr. Donald A. Taylor, Chairman of the Department of Marketing and Transportation at MSU, Dr. Thomas J. Manetsch, Associate Professor of Electrical Engineering and System Science at MSU, and Dr. Donald J. Bowersox, Professor of Marketing and Transportation at MSU, who was the chairman of my committee. Dr. Bowersox must also be particularly acknowledged for his assistance in the development of my dissertation. Being fairly strong in the use of computers and developing a deeper understanding of physical distribution concepts, I relied heavily upon his guidance in designing the universal aspects of the computer model.

Mrs. Ann Brown and Mrs. Irene Orr must be thanked for typing and preparing the dissertation in final form. Their knowledge of the rules and guidelines for dissertation preparation made my task much easier.

My wife Janet must be sincerely thanked for typing many rough drafts of this dissertation, spending many lonely hours while I was writing plus giving me the incentive to persevere to its completion.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure		Page

Figure		Page

Figure | Page

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

Scope of the Research

To provide consumers with a wide assortment of goods at the right time, in the right place and at the highest profit potential, managers have increasingly viewed their product distribution systems on an integrative basis. Costs of transportation, warehousing, holding and handling inventories and communications should be considered from a total, integrated viewpoint. Requirements of customers for fast and consistent service, plus a rapid pace of technological improvements throughout the entire distribution system add to the plight of management in designing efficient and effective distribution systems.

Management has the responsibility of not only designing better distribution systems for supplying goods to their customers but also administering these systems within their associated channel structures. Bowersox, Smykay and LaLonde define physical distribution management as follows:

> Physical distribution management is defined as that responsibility to design and administer systems to control raw material and finished goods flow.[1]

This definition stresses finished goods distribution and the physical supply of the raw materials for finished goods' manufacturing.

Physical distribution matters also extend beyond the limits and controls of the individual firm. To properly plan a physical distribution system, total costs and overall service levels experienced by all members of the channel(s) of distribution should be considered. To do otherwise could lead to suboptimization in the attainment of channel as well as firm objectives and goals.

One of the main problems, therefore, that confronts firms is the balancing of the costs of physical distribution against the desired level of customer service. In order to achieve market goals, the level of realized, actual customer service must accomplish the target level while keeping within the total cost constraint.

Given these types of questions, considerations and problems, a specific need exists for the development of improved planning models to aid firms in the design of total physical distribution systems. Such a planning model should be capable of assisting managers in analyzing, over time, the effect of alternative system configurations within the dynamics of a changing marketplace and the internal business environment. The planning model should be capable of assisting the managers in the determination of the best time to implement physical distribution system modifications. Test evaluations of proposed system changes should also be capable of analysis with respect to sensitivity to different assumptions concerning sales, costs, technological improvements and new product introductions.

The overall objective of research conducted at Michigan State University was to provide such a physical distribution planning model. The model has been titled Long-Range Environmental Planning Simulator (LREPS). This dissertation deals with the computerization of the model on the Control Data Corporation 6500 computer using primarily the GASP IIA simulation language and FORTRAN IV.

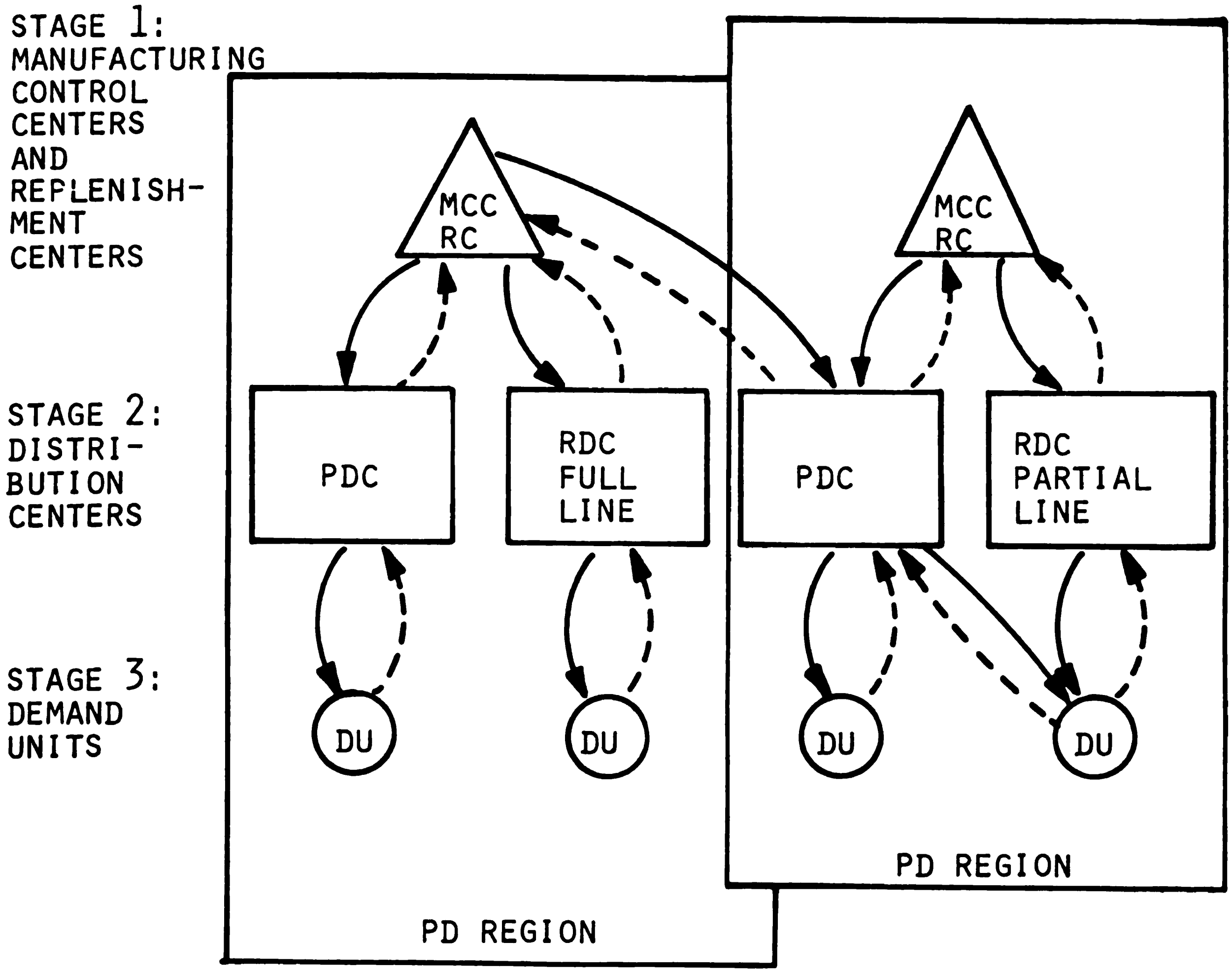
Situational Analysis

The dimensions of the physical distribution system developed in LREPS extend the channel of distribution from the production line to the point of ownership transfer. Included are five basic elements of the physical distribution operating system. The five elements are classified as follows:

1) the addition, deletion and modification of distribution center facilities for the holding and handling of finished goods inventories;
2) the transportation of finished goods from manufacturing centers or supply points to distribution centers and then on to the point of ownership transfer;
3) the communication and processing of customer orders;
4) the physical picking and preparation of orders for shipment dispatch to customers;
5) the holding and controlling of the level of finished goods inventories to supply customer needs consistent with various inventory policies.

In terms of the channel structure these five basic elements of the operating system are considered at three stages of activities. A graphical view of this system is shown in Figure 1. The three stages of activities are:

1) the Manufacturing Control Center (MCC) at which production is accomplished and inventoried at a Replenishment Center (RC);
2) the Distribution Center (DC) in which products are located adjacent to the marketplace;



------INFORMATION FLOW ——————PRODUCT FLOW

REGION..THE REGION IS DEFINED BY THE ASSIGNMENT OF RDCS AND DUS TO A PDC.
MCC.....EACH MANUFACTURING CENTER PRODUCES A PARTIAL LINE.
RC......REPLENISHMENT CENTERS STOCK ONLY PRODUCTS MANUFACTURED AT COINCIDENT MCC.
RDC.....REMOTE DISTRIBUTION CENTER, FULL OR PARTIAL LINE.
PDC.....PRIMARY DISTRIBUTION CENTER, EACH PDC IS FULL LINE AND SUPPLIES ALL PRODUCTS TO DUS ASSIGNED TO THE PDC REGION; PRODUCT CATEGORIES NOT STOCKED AT THE PARTIAL LINE RDCS IN THE REGION ARE ALSO SHIPPED BY THE PDC.
DU......THE DEMAND UNIT CONSISTS OF ZIP SECTIONAL CENTER(S).
CSP.....CONSOLIDATED SHIPPING POINT.

Figure 1.--Stages of the Physical Distribution Network[1]

[1]D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, Forthcoming).

3) the individual customer's demand or agglomerations of customer demands identified as the Demand Unit's (DU) stage.

The distribution center stage, Stage 2, incorporates four basic types of distribution nodes. Primary Distribution Centers (PDC's) handle a full line of products and have the potential of serving all demand units in a defined region of the firm's total market area. A distribution center that handles a full line of products and is not the primary distribution center in a market region is called a Remote Distribution Center-Full line (RDC-F). A RDC-F serves only part of the DU's in a region but it serves them entirely. A RDC that only supplies part of the product lines to its assigned DU's is called a RDC-Partial line (RDC-P). The PDC to which the RDC-P is linked will supply the other products to the assigned RDC-P's DU's. Also included in the DC stage are Consolidated Shipping Points (CSP's). CSP's are very similar to RDC-P's but handle no products. They serve as geographical aggregative points for the demand of several DU's. The total demand for several DU's is agglomerated and then shipped on a break-bulk basis to the geographical point of demand.

Given this general framework of analysis which fits most firms in the manufacture of both industrial and consumer package goods, the LREPS project team

conceptualized a simulation model. A research monograph written by the LREPS project team provides a detailed discussion of the purpose of the model, general class of problem to be modeled, the conceptual model, the capabilities of the model plus the use of simulation as the appropriate solution technique.[2] Simulation was considered most appropriate when considering the system characteristics and complexities plus the research objectives. The range of simulation is described in Figure 2 which illustrates the model's major subsystems. Each of these subsystems was conceptualized to emphasize some particular aspect of the operations of the physical distribution system.

The Demand and Environment (D&E) Subsystem focuses on the major inputs of customer mix, product mix, and order characteristics, plus forecasting and allocating sales among demand units. The Operations (OPS) Subsystem processes simulated customer orders through the major physical distribution activities or elements associated with transportation, facilities, communications, materials handling, unitization and inventory control. The Measurement (MEAS) Subsystem develops the criteria for evaluating alternative distribution system configurations. The Monitor & Control (M&C) Subsystem is the model supervisor and controller section of LREPS. The LREPS controller not only effects model feedback responses from past activities but also incorporates changes in

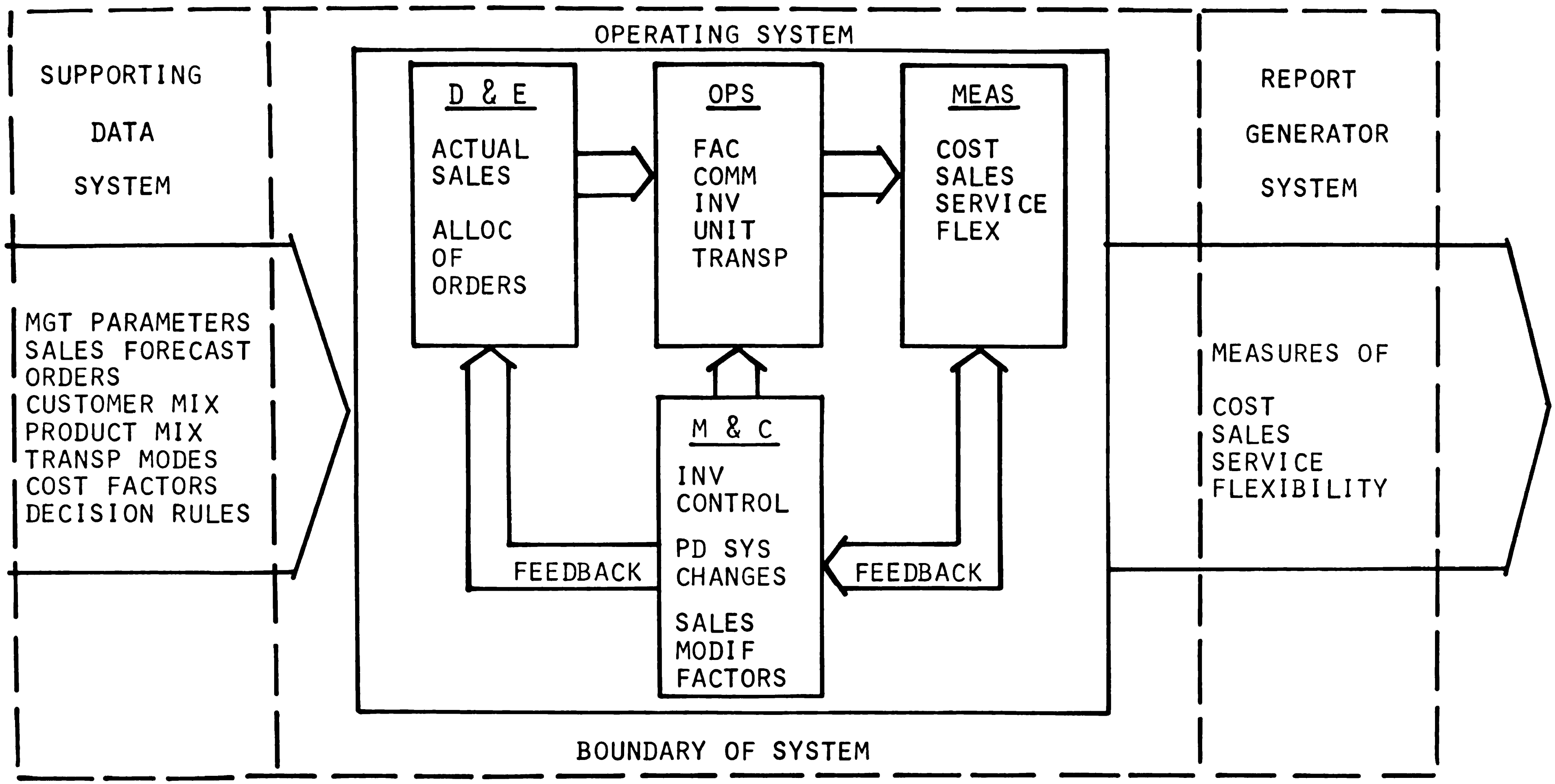


Figure 2.--LREPS Systems Model Concept[1]

[1]D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, Forthcoming.)

environmental factors and conditions. These controller actions, therefore, dynamically affect decisions concerning future facility configurations, transportation, communications, inventory policies plus DU, DC, regional and MCC sales. Each of the major subsystems is supported by a data analysis, preparation and reduction system labeled the Supporting Data System. The output is converted into managerial reports by the Report Generator (RPG) System.

System Identification

Figure 3 identifies the system design procedure or logical progression of activities utilized in the development of the LREPS simulation model. The shaded area of Figure 3 illustrates the focal point of this dissertation. In the "Problem Definition and Feasibility Study" the goals of the research were examined and substantiated via a preliminary collection and analysis of data. The outputs of this activity were a detailed problem statement, the specifications for a mathematical model and the boundary conditions on a defined set of feasible alternative solutions.[3] Given the outputs of the feasibility study, the next step was to develop an abstract representation of the mathematical model. This step emphasized the specific definition of: (1) the system boundaries and assumptions, (2) the inputs and outputs, (3) the constraints on the inputs, outputs and system configuration, (4) the criteria for evaluating alternative systems, (5) the design

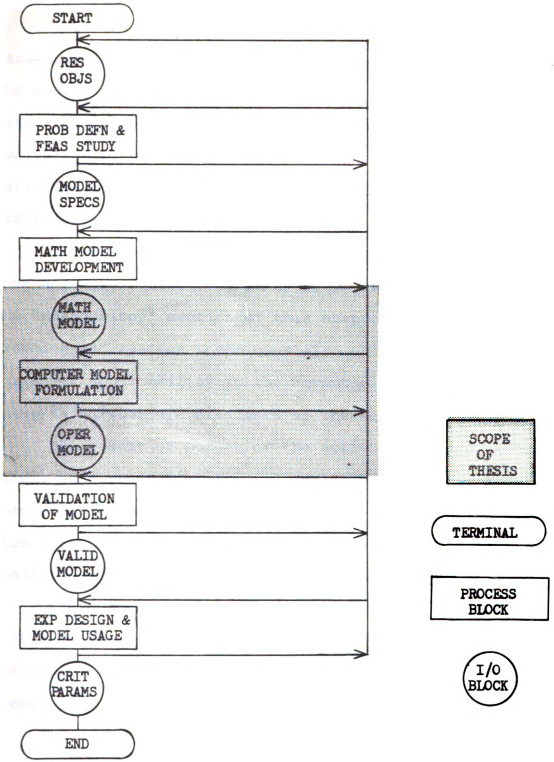


Figure 3.--LREPS Systems Design Procedure[1]

[1]D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, Forthcoming).

parameters that will be the basis for experimenting with the model, (6) the estimation of parameters via further data collection and analysis, and (7) a preliminary evaluation of model validity. The output of this mathematical modeling process served as the basis for constructing the computerized model.[4]

In formulating the computerized model, an overall approach was required. This approach is discussed in the "Methodology" section of this chapter. Various programming and systems techniques also had to be reviewed for their applicability in the computerization process. These techniques are discussed as the need arises.

The result or output of the activities associated with this dissertation is an "Operational Computer Model." This computer model serves as the input into the application areas that include detailed model validation plus sensitivity analysis and experimentation.

The output of the last two processes in the overall system design procedure is a set of critical management parameters that aid in solving the defined research problems.

Detailed Problem Statement and Researchable Questions

Analysis of the mathematical model served as the basic input for developing an operational computer model. The boundary conditions for the system and feasible solutions were defined in terms of economic, financial, social, political and physical characteristics of the system. The parameters that will be the basis for experimenting with the model, (6) the estimation of parameters via further data collection and analysis, and (7) a preliminary evaluation of model validity. The output of this mathematical modeling process served as the basis for constructing the computerized model.[4]

In formulating the computerized model, an overall approach was required. This approach is discussed in the "Methodology" section of this chapter. Various programming and systems techniques also had to be reviewed for their applicability in the computerization process. These techniques are discussed as the need arises.

The result or output of the activities associated with this dissertation is an "Operational Computer Model." This computer model serves as the input into the application areas that include detailed model validation plus sensitivity analysis and experimentation.

The output of the last two processes in the overall system design procedure is a set of critical management parameters that aid in solving the defined research problems.

Detailed Problem Statement and Researchable Questions

Analysis of the mathematical model served as the basic input for developing an operational computer model. The boundary conditions for the system and feasible solutions were defined in terms of economic, financial, social, political and physical characteristics of the system. The model's inputs were classified as either controllable or uncontrollable. The outputs of the system model included the alternative criteria for evaluating a tested physical distribution plan. The system outputs included such items as sales, costs, service and flexibility measures. Given the system characteristics, Figure 4 illustrates the various aspects considered in computerizing the mathematical model. The detailed problem was to develop a computerized model that satisfied the needs and specifications of the abstract mathematical model, the validity of the model, the model's data base requirements and stayed within the constraints of the defined computerization process.

The constraints on the inputs, outputs, optimizing parameters and the overall process of formulating the computer model were phrased in the form of researchable questions. These researchable questions, which lent direction to the computerization process, are:

1) What programming languages will facilitate the development of a reliable computer model including the supporting data system and the report generator system?
2) What model building procedures will facilitate the development of the computer model, the later sophistication of the model's defined activities, the broadening of the model to encompass

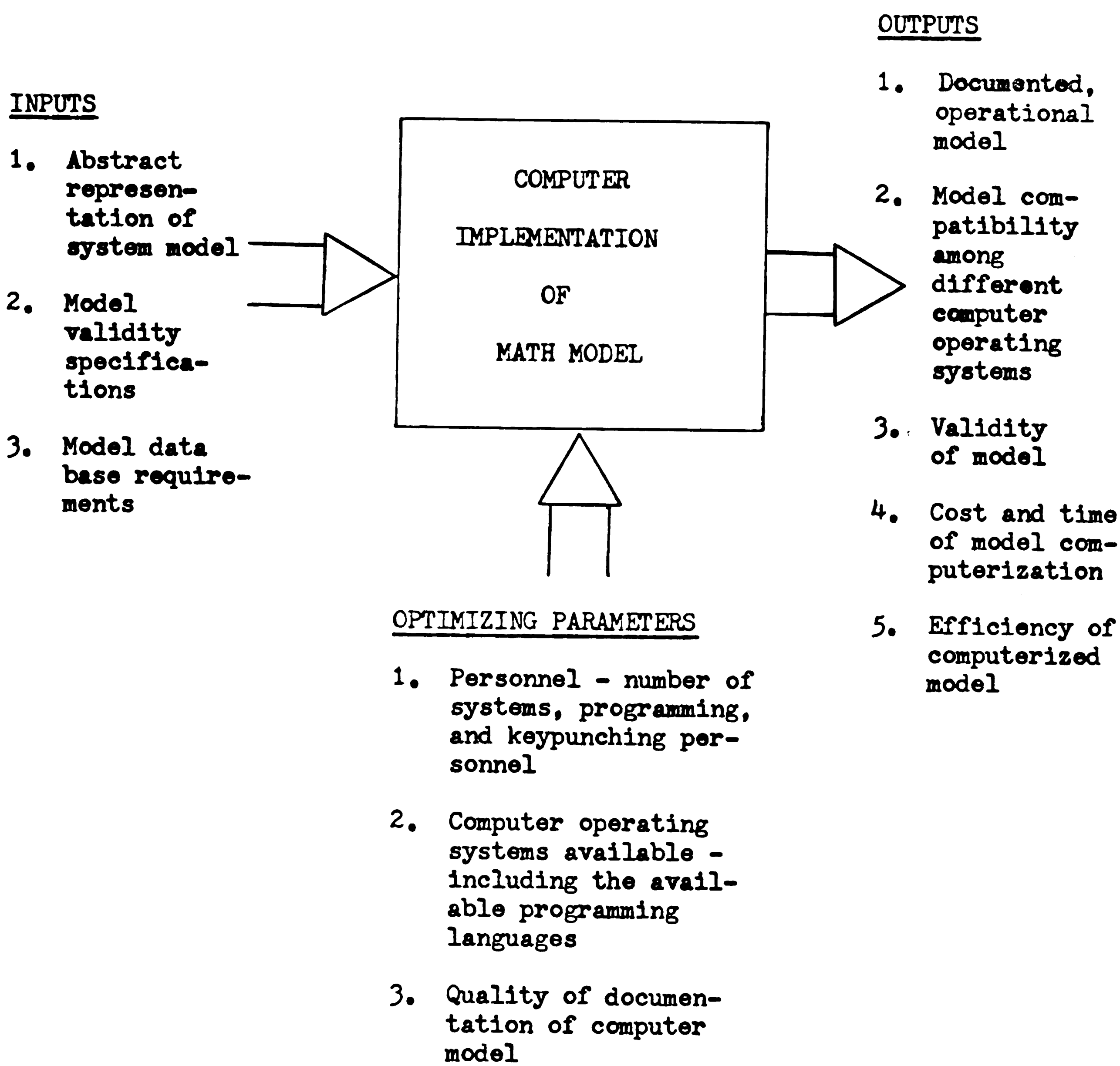


Figure 4.--Problem Input/Output Analysis

additional horizontal and vertical aspects of the total business system and satisfy the strong desire for universal applicability among packaged-goods firms?

3) What computer software and hardware features will allow the structuring of the simulation model's data base and its input/output requirements that will minimize reprogramming these structures for alternative LREPS versions?

4) What computer software and hardware features will promote the efficient processing of the great amounts of information that will accompany a model spanning a maximum ten-year planning horizon?

5) What is the most efficient way to write the model's output in order to provide printed reports of the essential data and to provide storage that can be easily accessed later as input to programs for further simulation results analysis?

6) Can a LREPS model be developed that will be highly compatible among different computer operating systems (hardware and software combined)?

7) Can a model be developed that will run on medium-size computer operating systems?

8) Can a computer program of the main operating system of LREPS be developed that will:
   a) Cycle through a ten-year planning horizon within a total elapsed computer time of 30 minutes?
   b) Fit within a computer memory limitation of 36K decimal words?
   c) Require no more than three input/output files?

Methodology

In order to develop a computerized operational model of LREPS and attempt to satisfy the above researchable questions, the following approach was developed:

1) Specify the computer system model's data base in detail using the mathematical model specifications and the broad system flowcharts.
2) Evaluate and select the predominant computer programming language for the simulator. The selection was from general compiler languages such as FORTRAN or ALGOL and from simulation languages such as GASP, FORDYN or SIMSCRIPT.
   a) A set of criteria for selecting this predominant language was developed in order to make this decision.

3) Segregate and program the computer model activities and their respective data base requirements according to these three main sections:
   a) Supporting data systems which were primarily developed for collection, preparation and reduction of data inputs to the LREPS operating system;
   b) Operating system programs which were those programs and activities directly involved in the LREPS simulator;
   c) Report generator programs for the development of the reports containing the model's results.
4) For each individual subprogram associated with the LREPS procedure:
   a) Identify the respective system model activities within the program;
   b) Identify the information flowing in and out of the subprogram plus any key endogenous variables;
   c) Block diagram the basic, logical steps for generating the program's outputs after considering alternative computer systems and programming techniques;
   d) Code the program using the most appropriate computer programming language(s);

e) Test and debug the program using a test case that had been developed;

f) Present any computerization problems, and any peculiar or unique programming techniques.

5) Combine the subprograms and their respective inputs, outputs and data base requirements, making any necessary revisions as a result of this step.

6) Test and debug the combined subsystem models to operationalize and calibrate to the actual system.

a) the D&E subsystem was first programmed and calibrated;

b) The D&E and OPS subsystems were then merged to develop the second LREPS model and it was also calibrated.

c) The D&E, OPS and MEAS subsystems were merged into the third version of LREPS which also had to be calibrated.

d) Finally all four major subsystems were merged into the first total LREPS model. This model was calibrated and served as the basis for the first series of experiments, utilizing LREPS.

The above approach required feedback at each level of development as more detailed and concrete system characteristics were formulated.

Organization of the Thesis

This dissertation is organized into nine chapters. Following this initial chapter, Chapter II presents the organization of the total computer model for LREPS. Chapter II also defines many of the important computer modeling concepts that are used throughout the dissertation. In the last sections of the second chapter, the evaluation of alternative computer programming languages plus the selection of the predominant computer programming language are discussed.

Chapters III through VI discuss each of the four major subsystems into which LREPS is segmented. Each of these chapters is subdivided into three sections. Section one is titled "Inputs, Outputs and Data Base Requirements." This subsection broadly discusses the subsystem's inputs and outputs and it presents an overview of the computer subprograms or activities that are discussed in detail in the latter sections of the chapter. The second section of each chapter, titled the "Operating System," presents the computer activities directly involved in the operation of the LREPS simulator. The "Supporting Data System" is the title of the last subsection. This subsection represents and discusses the computer activities that are required in the analysis, preparation and reduction of the data inputs.

Chapter VII describes that part of the total computer system in which the output reports are prepared. This chapter is titled "The Report Generator System" and is divided into three sections. The first section discusses the inputs to the report generator (RPG) while the last two sections describe the reports prepared in order to examine the results of a simulation cycle.

Chapter VIII presents a discussion of the operationalized total LREPS computer model. This discussion includes: 1) a description of the computer linkages between the operating system's subprograms and major subsystems; 2) the implementation procedures followed in developing the operational computer model; and 3) the results from a preliminary validation of model outputs.

Chapter IX presents the results of this thesis in relation to the researchable questions stated in Chapter I. Implications for future research are also discussed in relation to research results.

CHAPTER I--FOOTNOTES

[1] D. J. Bowersox, E. W. Smykay, and B. J. LaLonde, Physical Distribution Management (New York: The MacMillan Co., 1968), p. 5.

[2] D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, Forthcoming).

[3] Ibid.

[4] O. K. Helferich, Development of a Dynamic Simulation Model for Planning Physical Distribution Systems: Formulation of the Mathematical Model (unpublished Doctoral Dissertation, Michigan State University, 1970).

CHAPTER II

TOTAL COMPUTER MODEL CONCEPTUALIZATION

Preliminary Input/Output Analysis

At a broad level the inputs and outputs of the LREPS model were originally classified in the categories exhibited in Figure 5. This figure was developed in the conceptualization of the system boundaries and the experiments using the LREPS simulation model. The system model outputs included the criteria or target variables for evaluating alternative distribution plans. These outputs included measures of sales, service and total cost plus a measure of flexibility among different distribution plans. Flexibility was concerned with the amount of risk involved in a given plan when considering the uncertainties of the future.

The development of the specific outputs of the system model is evolutionary in nature. The initial reports from the model Report Generator were reports that summarized the results of the model for individual simulation cycles. A simulation cycle was considered one pass through a planning horizon.

TARGET VARIABLES

SALES
CUSTOMER SERVICE
PHYSICAL DISTRIBUTION SYSTEM COSTS
PHYSICAL DISTRIBUTION SYSTEM FLEXIBILITY

CONTROLLABLE VARIABLES

ORDER CHARACTERISTICS
PRODUCT MIX
NEW PRODUCTS
CUSTOMER MIX
FACILITY NETWORK
INVENTORY POLICY
TRANSPORTATION
COMMUNICATIONS
UNITIZATION

UNCONTROLLABLE VARIABLES

MARKETING ENVIRONMENT
TECHNOLOGY
ACTS OF NATURE

Figure 5.--Summary of Experimental Factor Categories[1]

[1]D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, forthcoming).

On a more detailed basis, the system outputs included part of the endogenous variables generated in the system as a result of the controllable and uncontrollable exogenous variables that were inputted into the model. Naylor, Balintfy, Burdick and Chu define the system inputs and outputs in this way:

> Endogenous variables can be best defined as the dependent or output variables of the system and are generated from the interaction of the system's exogenous and status variables according to the system's operating characteristics. Exogenous variables are the independent or input variables of the model and are assumed to have been predetermined and given independently of the system being modeled. Exogenous variables can be classified as either controllable or uncontrollable. Controllable or instrumental variables are those variables or parameters that can be manipulated or controlled by the decision makers or policy makers of the system.[1]

The system model inputs included controllable factors related to the firm's product mix, customer mix, the characteristics of the orders, the management policies related to the channel(s) of distribution plus the list of factors related to alternative physical distribution systems.

The uncontrollable factors included the assumptions concerning cost-of-living increases by area of the country, real rate increases in cost categories, growth rates concerning environmental, demographic factors such as population, and the delay times for alternative modes of transportation or communications given that the mode was selected.

Given these broad categories of inputs and outputs of the system model, the mathematical model was developed to describe the activities necessary to process and produce this information.

LREPS Computer System Flowcharts

Total Computer System Flowchart

After the system was mathematically modeled, the computer formulation of the model required a general overview of how to interrelate all the described activities into an efficient computer model. The philosophy was to develop a total computer system that would efficiently and effectively produce the desired system outputs. The essential goal in programming the model was to process the model's data with the least redundant operations possible. This goal was especially critical when considering the large amounts of information that would be manipulated in the model where the danger was to waste much computer time and effort. In line with these desires, Figure 6 was developed to conceptually direct the overall computerization of LREPS.

As viewed in Figure 6, much of the data needed within the operating system of LREPS was prepared off-line in the supporting data system. This off-line data was read into the operating system as exogenous inputs at specified periods of time. Also supplementary or

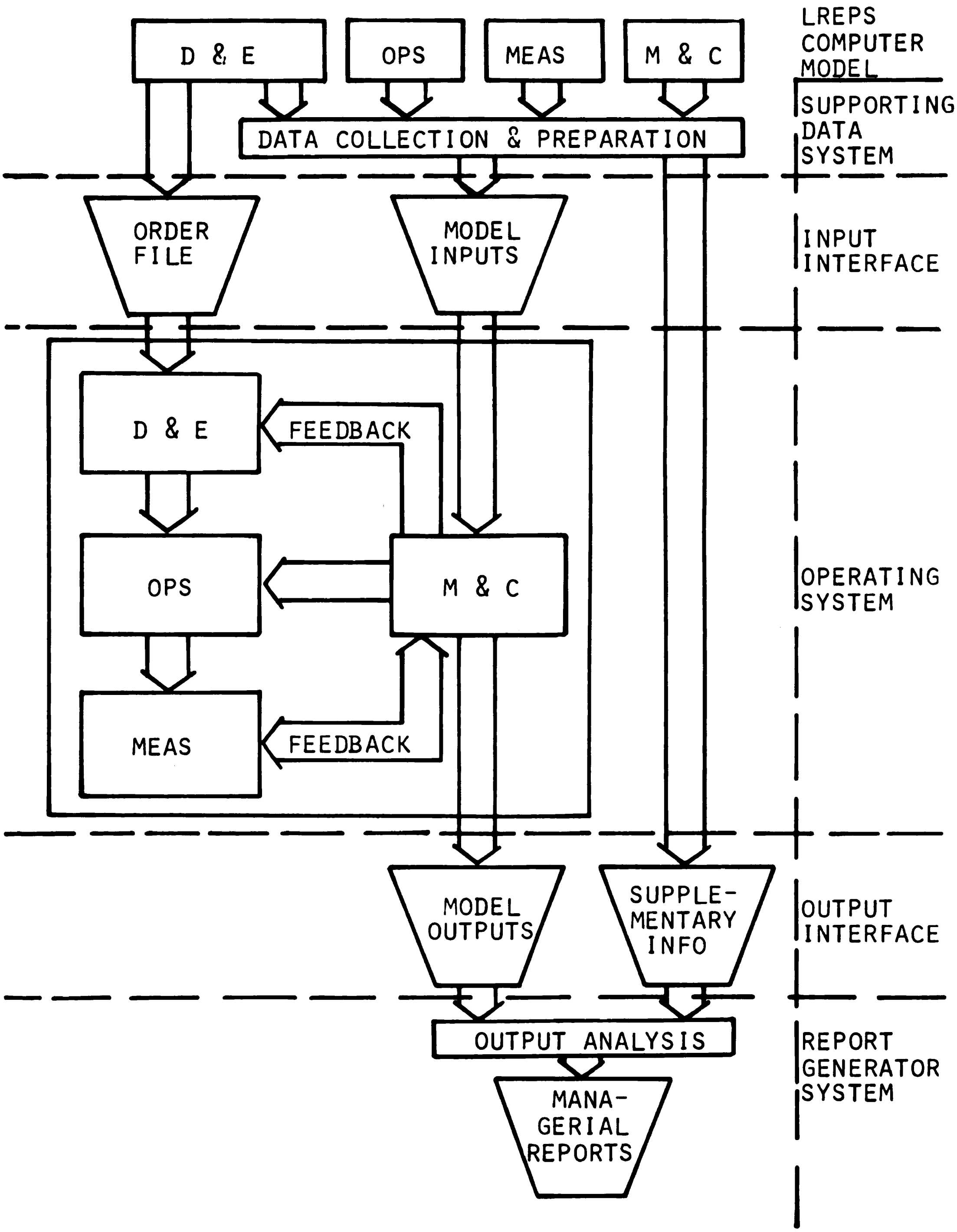


Figure 6.--LREPS Total Computer System Flowchart[1]

[1]D. J. Bowersox, et al., Dynamic Simulation of Physical Distribution Systems, Monograph (East Lansing, Michigan: Division of Research, Michigan State University, forthcoming).

or identifying information not needed in the operating system was sent directly to the report generator system.

LREPS Data Base

In developing this data manipulation concept, a total system data base had to be defined. The total data base included all the information that was needed in the system in order to develop and analyze the results of simulation cycles. This data base was further segregated into two classes. The first class of data was the common data base needed within the operating system of LREPS that was to be shared among, between, or used individually by the major subsystems of the model. The major subsystems are, as already defined, the D&E, OPS, MEAS and M&C. This first class of information is listed in Appendix 1. The other class of information was the supplementary information needed by the report generator to prepare the managerial reports.

The first class of information along with the system activities described by the mathematical model serves as the basis for discussing each of the major subsystems of the model. Each subsystem, such as the D&E, includes a discussion of:

1) The operating system activities that manipulated the endogenous variables. These generated endogenous variables served as the output of the subsystem

that went either to another subsystem or to the report generator system.

2) The supporting data system activities that generated the exogenous inputs required in the operating system.

By examining the operating system's common data base in Appendix 1, it is noted that the data was classified as to type of data, how often it was changed, whether it was an endogenous or exogenous change, what subsystem altered or set it, what subsystem used the data and the mode of the data. The mode was used to classify the data as integer, real or packed information. In addition, many of the variables were classified as exogenous for only the first versions of LREPS. Later the addition of a LREPS modular activity could produce some of these variables endogenously.

Computer System Interfaces

After the operating system's common data base was developed, a problem was how to control the inflow of exogenous inputs at specified time intervals without having each major subsystem inputting its own information. Therefore, the concept of an interface between the operating system and each of the other computer systems was developed. An interface was defined as the nodal point in data flow where two major computer systems interacted. The input interface was segmented into two parts. The first part was the order file.

Because of the potentially large amounts of information in this file given the decision had been made to work with individual customer orders, the orders were generated off-line in the D&E supporting data system and read in as a separate file. The other segment of the operating system's inputs was the file containing the remaining exogenous inputs. It should also be noted that these exogenous inputs into the operating system might have been endogenously created in a supporting data system activity. The decision was made not to incorporate certain activities directly in the operating system when efficiencies in computer processing could be achieved.

The last file of exogenous inputs was designed to agglomerate all of the model's subsystem inputs into one file that would be inputted into the M&C subsystem. The M&C was not only conceptualized as being the supervisor and controller of the model but also the gatekeeper controlling the flow of exogenous inputs into the operating system's common data base.

LREPS Detailed Computer System Flowcharts

Figure 7 illustrates in more detail the supporting data system program to collect and prepare the exogenous inputs for each specified time period. This program is directly related to a catalogued or segmented data base and provides a convenient method for preparing and controlling the inputs into the operating system. Figure 8

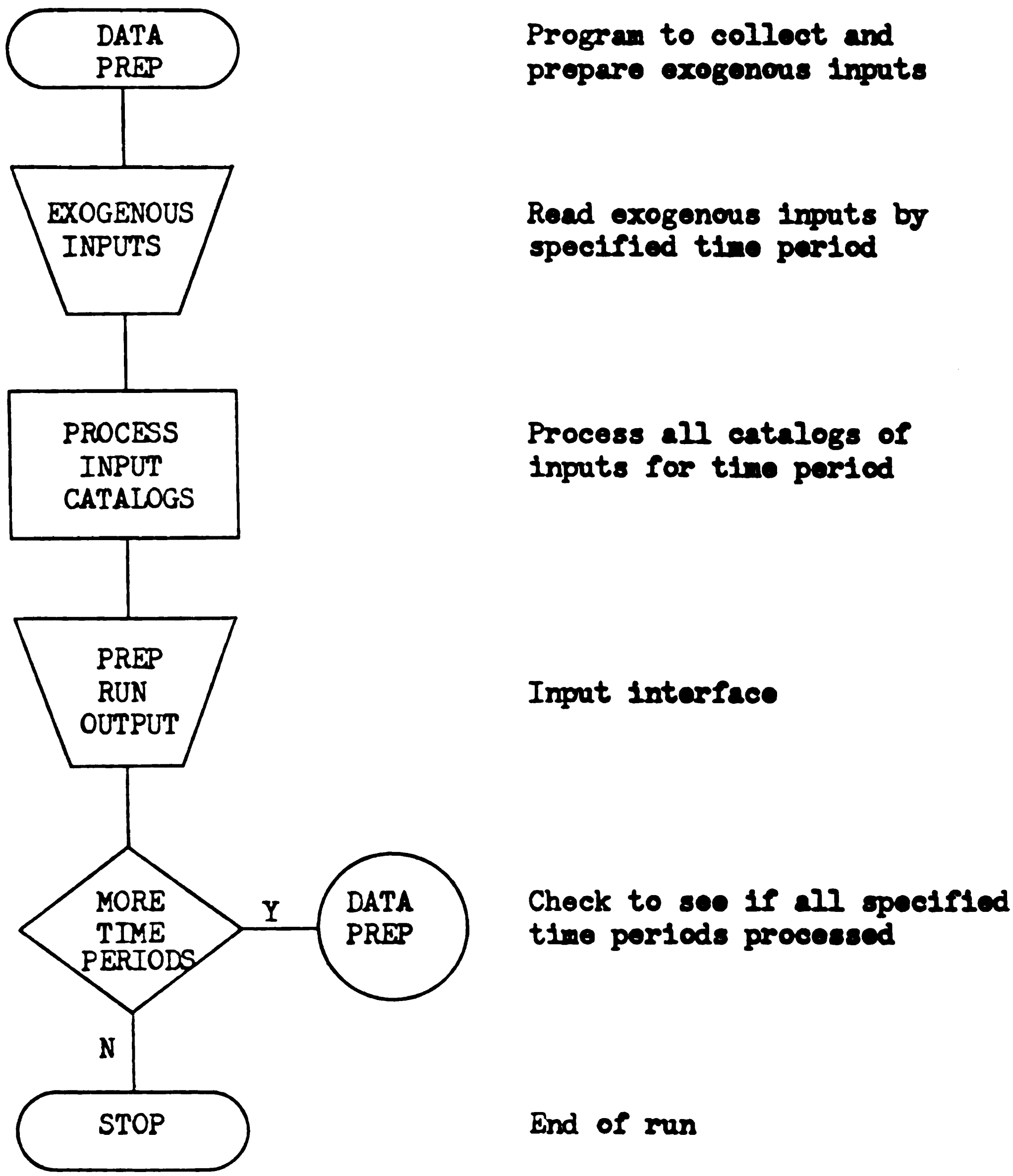


Figure 7.--Exogenous Input Preparation

illustrates the operating system's interaction not only with the input interface but also the output interface.

As Figures 6 and 8 illustrate, the M&C subsystem was conceptualized as the gatekeeper controlling the outflow of information from the operating system. M&C basically wrote an output interface containing run control information, specific distribution center information and the common data base for processing by the report generator system.

Figure 9 illustrates the activities of the report generator system in processing not only the operating system's output but also the supplementary information sent to it by the supporting data system. Both cyclic and multi-cyclic reports are conceptualized as being prepared in this computer system.

Selection of Predominant Computer Programming Language

Language Selection Criteria

The purpose of this section is to discuss the selection of GASP IIA as the computer programming language predominantly used in programming LREPS. The alternative languages evaluated are general-purpose languages such as FORTRAN or ALGOL and simulation languages such as GPSS or SIMSCRIPT. Predominance must be stressed because other programming languages besides the main language were used to process much more efficiently some of the

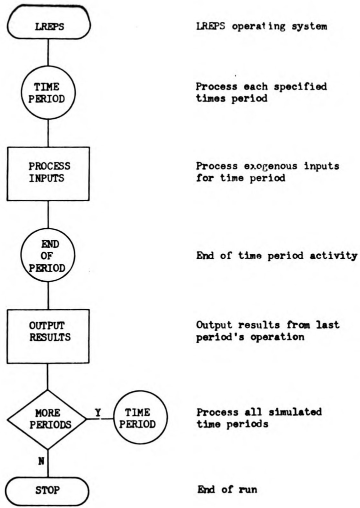


Figure 8.--Operating System's Interface with Input/Output

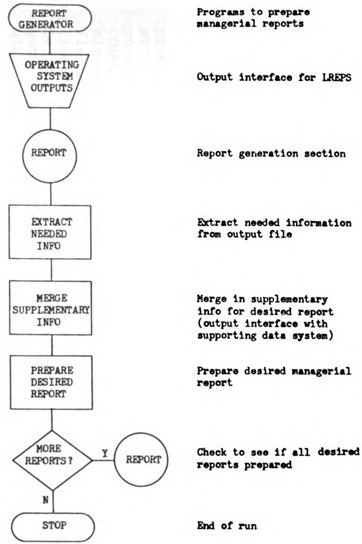


Figure 9.--Report Generator System

activities in the total-system model. The use of these other languages took place in the supporting data, operating and report generator systems.

The literature is abundant with articles comparing and evaluating alternative programming languages. Naylor, Balintfy, Burdick and Chu state:

> Clearly one way to approach programming simulation experiments is to write a special program for simulating each system to be studied in one of the well-known, general-purpose languages such as FORTRAN, ALGOL, COBOL, or IBM's PL/I. To be sure, this alternative offers the programmer maximum flexibility in (1) the design and formulation of the mathematical model of the system being studied, (2) the type and format of output reports generated and (3) the kinds of simulation experiments performed with the model.[2]

However, these authors further comment on the difficulty encountered in trying to write routines to step the model through the time periods of the model. To aid the computer implementers of simulation models and to facilitate model experimentation, simulation programming languages have been developed. These simulation languages serve essentially three vital purposes:

1) Reduce the programming task.
2) Provide conceptual guidance.
3) Allow flexibility for change.

Essentially a simulation package consists of the language used by model builders and the "run-time" routines which are present during the simulation.

After reviewing many of the important pieces of literature concerning the available programming languages and their applicability to simulation, criteria for selecting the most appropriate language for the formulation of the computer model were required. The language selected had to be appropriate given the nature of the model plus be able to manipulate efficiently and effectively the data base of information described in the previous section of this chapter. The essential nature of LREPS was a hybrid system that had both fixed- and variable-time activities occurring during the simulated planning horizon.

The set of criteria for evaluating and comparing many of the available languages are listed in Figure 10. These ten criteria were used to screen many of the available languages such as those listed in Tables II.1 and II.2 in the article by Teichroew and Lubin.[3] After the screening process, four languages were examined in detail as possible predominant programming languages. Finally, a predominant language was selected. The next sections of this chapter present a detailed discussion of the ten criteria, the four selected languages using the ten criteria and the reasons why the predominant language was selected.

The first criteria in Figure 10 refers to the structure of the programming language and how it lends assistance in getting the system model computerized.

1. PROGRAMMING LANGUAGE STRUCTURE
2. LEVEL OF REQUIRED PROGRAMMING SKILL
3. EASE OF CONVERTING LOGICAL BLOCK DIAGRAMS TO COMPUTER PROGRAM
4. SIMULATION PROCEDURES ASSOCIATED WITH
    A. INITIAL VALUES
    B. DATA GENERATION AND MANIPULATION
    C. TIME FLOW MECHANISMS
    D. OUTPUT ASSISTANCE
    E. STACKING OF A SERIES OF CYCLES
5. DEBUGGING AND ERROR ASSISTANCE OFFERED BY LANGUAGE
6. COMPUTER COSTS OF COMPILATION AND EXECUTION
7. COMPATIBILITY OF LANGUAGE AMONG DIFFERENT COMPUTER OPERATING SYSTEMS
8. AVAILABILITY OF COMPUTER HARDWARE
9. EASE OF CHANGING SYSTEM MODULES
10. APPLICATIONS OF LANGUAGE TO DATE

Figure 10.--Criteria for Selecting Predominant Computer Programming Language

Specifically, the language must be compatible with the type of system model. Teichroew and Lubin refer to this when they discuss "what flows in the simulated world?" and "what determines when a change occurs?"[4] They continue their discussion of types of system models by classifying them as either continuous-change or discrete-change models.[5] Krasnow and Merikallio discuss these different systems--continuous and discrete--and languages that can process both types of systems.[6] Tocher refers to this system structure when he discusses the organization of the program required to piece together the various blocks of the program into a consistent order so that execution will change the system configuration approximately at each instant of time.[7]

In regards to the structure of the language one must also consider the conceptual guidance provided by the language in not only computerizing but also in developing the system model. This concept was stressed by several authors where they emphasized the overlap in mathematical modeling and formulation of the computer model. Naylor, Balintfy, Burdick and Chu mention this when they discuss the languages that they compare based on language components.[8] Krasnow also discusses this when he notes:

> The user can then utilize a way of approaching his problem which both reduces the intellectual task and assures compatibility with computer requirements for implementation.[9]

Associated with conceptual guidance several authors have commented on the assistance provided in getting a simulation model computerized via written subroutines to produce the model's activities.[10,11,12] Also associated with the subroutines that are already available for systemizing the model, one must consider the language components that are available for modifying the state of the system. Krasnow mentions the characteristics of the language for arithmetic operations, set operations and logical testing and program control.[13] This is also mentioned by Naylor.[14] The last thing to be mentioned in this area is the flexibility and ease of use of the language. Krasnow and Merikallio mention the flexibility and range of using simulation languages considering simulator-defined concepts, user-defined concepts, concepts and values defined at the start of the simulation and during the simulation.[15]

The second criterion is concerned with the level of programming skill required. This criterion is essentially concerned with the ease of learning the language.[16,17] The language may be a very powerful language but only if one has used the language for several simulations. One must, therefore, consider the availability of experienced programming skill and its cost.[18,19,20] A language may be available but because it is not easy to learn or sophistication only comes with experience, one may not be able to obtain people to assist you in computerizing the model. This is especially true if one is under a tight deadline for implementation. In addition, one of the biggest problems with using many simulation languages is the lack of adequate documentation and instructional material.[21] If good instructional material exists plus the language is not difficult to learn, the programming task can be greatly reduced.[22]

The third criterion is again one that cuts across the mathematical modeling and computerization tasks. It is concerned with the ease of converting logical block diagrams to computer programs. Some of the languages offer special flowcharting symbols that help in systemizing the model plus lead directly to model computerization.[23,24] If one does a number of simulations and there is good documentation, then the use of these special flowcharting symbols can be very helpful. One problem that frequently arises is that the flowcharting symbols are not consistent between languages. Thus, this aspect promotes inflexibility in the use of different languages.

The fourth criterion is concerned with some of the special features of the language in doing some of the more or less menial tasks in programming. Given that one has determined the type of system model and the desired outputs, simulation languages offer assistance in initializing values and arrays,[25] data-generation routines,[26,27] data-manipulation operations,[28] output assistance,[29,30,31,32] and stacking of a series of cycles through a simulation computer model.[33,34] Krasnow has a good discussion of these simulation procedures in his article.[35] Krasnow and Merikallio even talk of the man-machine interaction during simulation execution that some languages are beginning to offer.[36]

The next criterion deals with the debugging and error assistance offered by the language. In getting the model to execute, many of the languages have tracing routines to help one know where he is in a simulation either at the time of errors or change in simulated time.[37,38,39] They can be very helpful given that the supporting instructional material clearly states just how to use these features. All the languages provide the normal diagnostics for errors associated with programming rules.

The sixth criterion deals with the computer costs of compilation and execution. The efficiency of a computer run is concerned not only with execution time but also the time that it takes to get the program compiled into a machine language program that can be executed on the computer. Also involved in the programming of the model are the size and the type of data structures that are available with the use of a programming language. Data structure types include simple variables, arrays and packed variables.[40,41] If a simulation model is input/output bound, the capability of multi-programming execution can also be very important in reducing computer costs.

The seventh criterion is concerned with the compatibility of the language among different computer operating systems. A computer operating system is defined as that combination of computer hardware and software that approaches the optimum use of a particular computer. Given the operating systems that are available on different sizes and makes of computers, one can be restricted by not only the available computer systems but also the compiler requirements on the system.[42] In addition, the reliability of the compiler must be considered. A compiler may be available but never effectively tested and utilized.

The eighth criterion concerns the availability of computer hardware itself. Often one is restricted by the computers that are at his disposal. Some languages are just not available on certain machines.[43,44] This criterion is also concerned with the availability and use of auxiliary equipment such as disk or tape files.[45] Massive amounts of information may require disk or tape files. Another aspect that is important in determining the availability of a certain computer system is the direct cost of using the system. If one has available a system that is partly financed by outside or other sources either because of promotional reasons of computer manufacturers or educational discounts, then this can be a major factor in determining whether one uses a certain system and programming language. Also involved in this availability of hardware is the proximity of the computer to the personnel who are programming, debugging and using the model. If one has to go many miles to use a certain system or cannot use terminals for testing and/or running the model because the operating system is not oriented toward a particular simulation language, then one may also have to be satisfied with a less powerful language.

The ninth criterion has to do with the ease of changing the system model for changes in the system being modeled or to test the results of the system for system-design modifications.[46] Although general-purpose languages allow one to model a system in more detail, they cut down on the flexibility and ease of system changes. Simulation languages promote this type of system flexibility.

One of the very important criterion for evaluating languages is a record of the applications to date using the language.[47] If others have used a language for similar undertakings, it may indicate that a language is good for certain types of problems. Further, the language may have been written to handle specific types of problems with great ease.[48]

Alternative Languages Evaluation

The purpose of this section is to discuss alternative programming languages considered for operationalizing LREPS. The following subsection discusses important factors in the selection of certain types of simulation languages consistent with the nature of the defined system model. The next subsections discuss four programming languages that were seriously considered for the computerization process.

Preliminary Screening of Languages.--The LREPS model was designed to be general or universal in nature with a high degree of flexibility in processing system entities or objects. Primary emphasis in model design was the easy adaptation of the model to a large class of multiproduct, consumer and industrial, packaged-goods firms. In serving these classes of potential users, the purposes of the system model were conceptualized as both lending direction in the strategical design and testing tactical or detailed procedures for day-to-day operations of alternative physical distribution system configurations.

Given the desire for universal applicability and for serving the general purposes specified above, the ability to process alternative combinations of system entities and dynamically change system state over simulated time was necessary. A more detailed list of system entities than that given in Figure 1 of Chapter I includes: (1) demand units of potential agglomerated customer demand,

(2) customer classes of trade, (3) tracked product items, (4) customer orders, (5) dispatched customer orders from demand units to distribution centers, (6) full- or partial-line distribution centers for processing customer orders, (7) dispatched customer shipments from distribution centers to demand units, (8) customer backorders resulting from product stockouts at distribution centers, (9) multi-product reorders from distribution centers to supplying replenishment centers, (10) multi-reorder, dispatched shipments from replenishment centers to distribution centers, (11) potential distribution center locations, (12) manufacturing control centers for replenishing product inventories of replenishment centers, (13) inventory categories for grouping similar tracked products and (14) market regions to allow for variations in regional customer demand and service requirements.

Procedures for processing system entities, either singularly or in combination, were conceptualized on a discrete-change, event basis. Preliminary system activity analysis indicated that detailed modeling procedures were necessary to: (1) randomly allocate daily customer orders to a subset of all demand units, (2) dispatch customer shipments either on a daily basis or in consolidated shipments on greater than a daily basis, (3) process reorders dispatched from distribution centers to replenishment centers, triggered by varying inventory control values and occurring on an intermittent basis, and (4) dispatch

shipments from manufacturing control centers to distribution centers on an intermittent basis which varied with the accumulated amount of product on reorder plus other time-oriented, shipment dispatch policies. The use of discrete-change, event-processing programming languages was considered appropriate if complex system activities, such as those above, were to be modeled adequately and on a detailed basis by the LREPS project team.

The discrete-change events were conceptualized as occurring at a fixed time or randomly as conditions in the model dictated. Events were also conceptualized as being processed chronologically and on a first-in, first-out (FIFO) basis. The designed events could then be processed on a detailed day-to-day or week-to-week basis dependent upon the conditions specified in the event subprograms and model input.

Since system entities were conceptualized as being processed in discrete-event activities, programming systems for processing continuous-change, simulation models were excluded from serious consideration. The excluded computer systems included analog systems that treat the processing of system entities on a continuous-flow basis and digital systems that process system entities as continuous-flow rates on a discrete-change basis. The use of such programming systems requires that system entities be treated on a macro or aggregative basis rather than on an individual basis as conceptualized in the LREPS system model. If system processing procedures had been conceptualized on a macro basis, then continuous-change simulation languages would have been seriously considered.

Considering the above factors and programming languages that were reasonably available to Michigan State University and which could process discrete events on both a fixed- and/or variable-time basis, FORTRAN, GPSS, SIMSCRIPT and GASP were seriously considered for computerizing LREPS. Each of these languages will be briefly discussed in reference to the selection criteria in Figure 10.

FORTRAN.--The first programming language considered is a common general-purpose language called FORTRAN that is not specifically structured to simulation models but is scientifically oriented. Naylor mentions that the programmer has the flexibility of being able to write almost any subroutine that he may need for a particular simulation plus he can tie the subroutines together by using a main program and calling the specific subroutines when they are needed.[49] Although one gains much more flexibility using a general-purpose language such as FORTRAN, it requires more programming skill than some of the other simulation languages. Using FORTRAN one must be able to program many of the tasks that the simulation languages already provide. Thus one can become more involved in the programming rather than designing the structure of the model. One must, however, remember the availability of scientific subroutine packages that are available for data generation and statistical analysis on most computer operating systems.

Although FORTRAN has some definite disadvantages, one must consider the fact that FORTRAN is a widely used language with which many people are familiar. It is relatively easy to learn if one has some mathematical background. In fact, most universities are using FORTRAN in their courses; thus, providing a large supply of available programmers that can assist in the programming at a reasonable cost. Also many books have been written on the use of FORTRAN which alleviates the problem of poor documentation that occurs with the use of some of the other simulation languages.

There are no special flowcharting symbols to be used in programming FORTRAN simulations; however, many computer manufacturers are now developing standard flowcharting symbols for general programming. FORTRAN does not offer any special simulation procedures but one does enjoy very flexible formatting of output. Again, this flexibility is gained only at the cost of additional programming effort and time.

There are no logic checking routines built into the language; however, the language does print error diagnostics concerning FORTRAN rules of programming. Related to the lack of logic checking routines, there are also no tracing routines to aid in debugging.

In reference to computer costs, FORTRAN programs usually have small requirements for computer memory during compilation and execution. However, this is generally dependent upon the level of FORTRAN associated with a particular computer operating system. For instance, some FORTRAN languages have the ability to manipulate packed words of computer memory in which more than one piece of information can be put in one word.[50] This level of FORTRAN is called EXTENDED FORTRAN.

Another aspect of computer cost is compilation and execution time. One may experience high compilation times using FORTRAN depending on the structure of the system model and how frequently one may wish to change the system structure. Fast execution times can be often achieved, however, because the specific model activities that have been programmed have not been restricted to the rigid structures of simulation languages. Execution times can also vary with the amount of information packing that has taken place.

FORTRAN is highly compatible among different computer systems from small-scale to and including large-scale systems and among different computer manufacturers. However, one must realize that each manufacturer does not have the same operating system. Depending on the size of the simulation, the amount of special programming that is done and the level of FORTRAN used, this highly compatible feature can be very misleading. However, most computer manufacturers are going to and stressing the use of American Standards Association (ASA) FORTRAN.

As mentioned, this language is generally available on most computer manufacturers' systems, thus there is no serious limitation of small availability of computer hardware. Concerning the present study, both Control Data Corporation and International Business Machines computers are available.

The last consideration is the applications to date using the language. Naylor notes that FORTRAN has been used for many specific applications to date in a wide diversity of disciplines. This is primarily because of its scientific orientation.[51]

GPSS.--The second language to be considered was GPSS (General Purpose Systems Simulator) developed by IBM.[52] This language is primarily structured to scheduling, waiting-line and other similar problems in which a flow of transactions or individual entities over blocked-time intervals are analyzed. Priorities from 0 to 7 can be assigned to the transactions with up to eight parameters per transaction.[53] The language is not as detailed as FORTRAN but requires gaining a good familiarity with the structured parts of the language. FORTRAN-like statements are permitted in the program for algebraic computations on system state variables.[54]

Because the language has a special set of system flowchart symbols which can be directly referenced in the computer source program, the language provides not only conceptual guidance in developing a computerized model but also facilitates the development of the computer model. The language also provides special forms for getting the model computerized which also facilitates the computerization task. The biggest advantage of this language is the availability of simulation-defined and user-defined concepts to aid the systems analyst.[55]

The language assumes that one does have some systems' background plus a knowledge of FORTRAN or some other general-purpose programming language. With this background plus the aid of good documentation and instructional material, the language is not too hard to learn. However, as with any sophisticated simulation language, to realize the full potential of the language one must use the language and learn its subtleties through continued use and under proper experienced supervision. One shot deals about once a year would not allow one to realize the full potential of this powerful language. With a shortage of knowledgeable experienced GPSS programmers, programming costs would be expected to be higher than using general-purpose language programmers.

Simulation procedures are available for aiding in the initialization of system variables, generation of data, handling of transactions on a blocked-time interval, and writing of output records. The language also allows one to stack a series of simulation cycles. Simulation testing, validation and error-checking routines are available to aid the system analysts to get the model running. The handling of transactions can be based on a fixed- or random-time interval (including zero) dependent upon the conditions in the system or the activities jump to the next future event at which time all possible activities must be reviewed.[56]

Computer times of compilation and execution are generally faster than FORTRAN programs because GPSS uses many efficient assembly-language subroutines. These subroutines are readily available for the activities desired. Because they do not have to go through the translation from simulation language to a general-purpose language to a computer machine language these compilation times can be very fast. Also because expert programmers have written the machine-language routines, they are very efficient from the standpoints of execution times plus core and operating system utilization.

One of the main problems that one confronts with using these very powerful languages is that GPSS, for instance, requires at least a large-scale IBM 360 computer system with large core plus three tape drives for compilation and execution. In addition, GPSS specifically does not have the ability to pack variables as one can do in FORTRAN and, therefore, a system model of any complexity can require much core.[57] The only way packing can be handled is to write IBM 360 FAP assembly routines to handle these activities. As has been implied, GPSS is an IBM language that is only available on large-scale systems. There are different versions of GPSS with the most powerful version, GPSS III, available on the 360's only. The earlier versions of GPSS are available on such machines as the IBM 7000 series.[58] The ready availability of IBM computers, however, makes this language a very attractive choice.

Assuming that one has available an analyst who is knowledgeable of GPSS, it is easy to change a system model for study of system behavior and for maintenance. The applications to date have centered primarily in production scheduling, waiting-line problems such as tollbooth problems and in the simulation of computer operating systems. One of the main reasons why IBM developed the language was to aid in simulating time sharing systems.

SIMSCRIPT.--The next language to be discussed is SIMSCRIPT which is an "event oriented" language that is applicable to many diverse simulation problems.[59] The SIMSCRIPT language is based upon a description of systems involving concepts denoted by entity, attribute, set state and event.[60] Rather than having to review all possible activities that could take place at a certain point of simulated time, this language jumps to the next imminent event which triggers a given activity. This activity could also generate more variable-time events or just be ended by the next event in the queue. The language has many conceptual aids in getting a system model computerized. The language is not as detailed as FORTRAN but requires gaining a good familiarity of the conceptual components of the language to use it effectively. As in GPSS, FORTRAN-like statements are permitted in the program for algebraic computations and data manipulation on system state variables.

The language, similar to GPSS, requires a system background plus some programming experience with general-purpose languages in order to learn and be able to use effectively the basic, structural parts of the language. The language has no special set of system flowchart symbols but does have special coding forms that aid in structuring and computerizing the model.

Simulation procedures are available for aiding in the initialization of system variables, generation of data, handling of event transactions on variable-time intervals and perhaps the most flexible means of writing output reports of any of the high-level simulation languages.[61,62] Special report forms allow the system analyst to design many special reports. Also available are simulation testing, validation and error-checking routines. This language has more power than GPSS in being able to pack words in core and possibly allow more efficient program execution although GPSS may be easier to implement.[63] The language offers much power from the standpoint of efficient core utilization plus an event orientation that does not have to test all possible activities at an instant of time. The language also offers machine-coded subroutines to expedite the compilation of the activities of the model. Compilation and execution times are generally faster than FORTRAN programs because of these available, efficient subroutines.

The language is a RAND corporation language that is available for computer systems of many different manufacturers such as IBM, CDC and BURROUGHS. Presently the language is not available for the IBM 360 but is available on the IBM 7000 series systems. The language does require a large-scale computer system for compilation and execution. Computer hardware, a CDC 6500, is available at MSU for this language but it is not the most sophisticated level of SIMSCRIPT. Not only is it a lower level of SIMSCRIPT but also the compiler is not fully tested and reliable.

The availability of personnel who are experienced with this language is very limited. As with GPSS, this language requires that one use the language many times in order to reap the full potential of the language. The quality of documentation and instructional material varies with the level of SIMSCRIPT that is implemented on different systems. The later versions of the language, as with GPSS, have good supporting materials.

SIMSCRIPT programs are relatively easy to change for system maintenance and study given you have knowledgeable personnel assisting in the system analysis. SIMSCRIPT has been used for a wide range of problems especially large-scale simulations that require efficient core utilization plus fast computer execution times.

GASP.--The last language to be discussed is an event-oriented language that is applicable to many, diverse simulation problems. The language is called GASP (General Activity Simulation Program) and centers around a set of twenty-three FORTRAN-compiled subroutine programs and function subprograms linked and organized by a main program known as the GASP EXECUTIVE.[64] The language is a discrete-oriented language that provides conceptual guidance in the development of the system model. The language requires a knowledge of FORTRAN and system concepts in order to program the model. The language is easy to learn if one has some mathematical background and systems and programming experience. Arithmetic operations, set operations and logical testing are provided through user-written routines that are related to the model's defined activities.

The language provides no special flowcharting symbols for conceptualizing the model. There are, however, special coding forms that have been devised to aid in this conceptualization process. These forms are related to the special simulation procedures provided by the GASP subroutines for file maintenance, data generation, input and output, automatic monitoring of program variables and conditions for error detection and debugging, selective tracing of program flows for dynamic error debugging, and programmed dumping of system variables.

The GASP subroutines plus the normally available scientific subroutine packages that accompany most computer operating systems allow one to do a number of different types of simulation and non-simulation problems. Output formatting is still a problem for special reports. The GASP system has available report summaries of the historical information concerning the events and entities passed through its files but there is no flexibility in designing the format of these reports. Reports of non-GASP arrays and constants still must be formatted and wrote via a user-written subroutine. The simulation language has available a subroutine for the user in which he can specify these reports.

The GASP language has potentially small requirements for storage during compilation and execution dependent upon the expertise and experience of the programmers doing the computerization. If packed variables and other sophistications are used, one can develop a fairly efficient program from the standpoint of core memory utilization. The GASP simulation programs would possibly take more time to compile and execute than other simulation languages because of the highly efficient, already developed assembly- or machine-language subroutines that accompany the other languages. The specific GASP subroutines are, however, efficient and are very powerful when compared to the user-written routines that could be very inefficient. This again depends upon the level of sophistication stressed in getting the model computerized. In summary, by using the available GASP subroutines for the model, one does get a more efficient program from the standpoint of core memory utilization and running time than if one had to write his own subroutines for the general simulation activities mentioned above.

The language is highly compatible among different computer systems for small scale-to and including large-scale systems and among different computer manufacturers.[65] The language has been written in ASA FORTRAN with as little extensions to specific computer manufacturer compilers as possible.

Given the FORTRAN basis of this language, it can be implemented on any of the computer operating systems that have available a FORTRAN compiler. Some modifications would have to be made for scientific subroutines, such as random-number generators, plus one would, no doubt, experience the usual problems of getting a FORTRAN program implemented on different manufacturers' machines and even machines by the same computer manufacturers. Nevertheless, it should be easily implemented.

Because the user has to write FORTRAN-based subroutines to model the activities of the system under study, GASP programs usually require more detailed changes in the program for studying system behavior and for maintenance than other higher-level simulation languages. This again depends upon the stressed modularity of the program and the number of computer subroutines that are segmented in the overall program.

Large- and small-scale applications have taken place in many areas because of the FORTRAN basis of the language and its compatibility among installations. Also promoting the use of the language is the availability of good documentation and instructional materials. The main reason it has not been used more is the lack of publicity among users.

Selected Programming Language

After considering FORTRAN, GPSS, SIMSCRIPT and GASP, the language that was selected to program LREPS was GASP. The level of GASP chosen was the IIA stage that not only incorporated integer attributes in its filing system but also floating point or real attributes.[66] Given that the system model is discrete oriented, anyone of the languages could have been used for building a computer model of LREPS. However, because GASP has already worked out the programming details of many simulation activities it was felt that to program the model in FORTRAN alone was wasteful. Therefore, FORTRAN was ruled out. Considering the higher-level languages, SIMSCRIPT received the edge because of its variable-time orientation toward the execution of the model's activities. GPSS would seem to take more execution time especially for a model that would be simulating a ten-year planning horizon. SIMSCRIPT was also appealing from the standpoint of its efficient utilization of core memory considering the large amounts of information that were to be handled. Conceptually, the flowcharting assistance given by GPSS was also very attractive. It was attractive if persons who were knowledgeable of the flowcharting procedures were available. Related to this last statement was a very important problem concerning the lack of available personnel who were knowledgeable or who had used SIMSCRIPT or GPSS. There were a few people but they were not available at a reasonable cost.

The level of programming skill for SIMSCRIPT and GPSS was also greater than for GASP. Several students in the business and engineering schools at MSU had used GASP. Thus, extensive training of personnel to begin the task of computerizing the model would not be necessary. There were, however, regularly scheduled classes off-campus at high cost where training in the use of SIMSCRIPT or GPSS was offered. No classes at MSU offered instructions in the use of SIMSCRIPT or GPSS. GASP, however, was taught in several classes and schools.

All three languages offered assistance in simulation procedures associated with initial values, data generation and manipulation, time-flow mechanisms, output assistance and stacking of a series of cycles. The languages also assisted in debugging and error tracing.

These attractive aspects of simulation languages again supported the decision not to program the model in FORTRAN alone.

There were tradeoffs back and forth between the languages concerning computer costs of compilation and execution. The main cost of running GPSS at MSU was that the LREPS project team would most likely have to pay the high costs and receive low priority of computer run testing on MSU's administrative IBM 360-40 assuming a GPSS compiler could be readily obtained. At the time MSU did not have a GPSS compiler. The University of Michigan's IBM 360-67 was available but in this case the use of the computer would result in large travel expenses. GPSS had not yet been implemented on remote terminals to debug and test routines and to use the model after it had been operationalized.

All of the languages have facilitating features for changing the system model although GPSS and SIMSCRIPT have the advantage over GASP. SIMSCRIPT has a very important advantage over both GASP and GPSS in report generation because of its flexible report generator. Applications to date have been many and varied and offer little basis for a good comparison of the three languages.

Three main factors concerning the use of SIMSCRIPT were questionable. The first factor was the lack of a reliable compiler and good instructional material for SIMSCRIPT 1.5 at MSU. The second factor, related to the first, is the lack of a compiler at MSU for SIMSCRIPT II, the latest and most sophisticated version.[67] The last and, perhaps, most important factor was the very limited supply of personnel knowledgeable in the use of SIMSCRIPT. None of the LREPS project team members had ever used SIMSCRIPT for a modeling application. The only attempt was the testing of a sample problem for the SIMSCRIPT 1.5 compiler. The sample problem was never successfully tested since the MSU compiler for the CDC 6500 was not operational.

The final decision to use GASP IIA was based on six factors:

1) The high compatibility of the FORTRAN-based, simulation language among computer operating systems. The lastest, most powerful versions of GPSS and SIMSCRIPT only have compilers for International Business Machines (IBM) computers.

2) The minimization of the cost of using the model in different geographic locations if the model could be designed to be compatible among different computer operating systems.

3) The good familiarity of the LREPS project team with the general-purpose programming language FORTRAN.

4) The ready availability of sophisticated FORTRAN programming personnel, via MSU's student body, at a reasonable cost.

5) The ready availability of MSU's research computer center to operationalize LREPS. Costly time delays of not having a computer readily available would, therefore, be avoided. It should also be noted that the university research contract did not restrict the location at which the model could be operationalized but did restrict the amount of time and money that could be used in developing LREPS.

6) GASP was the only discrete-event simulation language operational on MSU's CDC 6500. If SIMSCRIPT II had been operational at MSU, then it would have received major consideration as the predominant computer programming language. The possible use of this language is an open area of research.

In relation to factors five and six, several authors support the great importance of the ready availability of computer hardware. Tocher mentions this factor when he discusses the fact that there are many important features of each simulation language but ". . . it remains a sad fact that the choice will most likely be made by the type of machine available to him."[68] Teichroew and Lubin predict that the time will come when the languages will be on most computers and this will not be such a problem to potential users.[69] Until such time, the ready availability of computer hardware and the languages operationalized on the hardware cannot be underemphasized.

CHAPTER II--FOOTNOTES

[1]T. H. Naylor, J. L. Balintfy, D. S. Burdick and Kong Chu, Computer Simulation Techniques (N. Y., London, Sydney: John Wiley and Sons, Inc., 1968), pp. 10-11.

[2]Ibid., pp. 239-240.

[3]Daniel Teichroew and J. F. Lubin, "Computer Simulation--Discussion of Techniques and Comparisons of Languages," Communications ACM, Vol. 10 (Oct., 1966), pp. 725-726.

[4]Ibid., p. 731.

[5]Ibid., p. 726.

[6]H. S. Krasnow and R. A. Merikallio, "The Past, Present and Future of General Simulation Languages," Management Science, Vol. II (Nov., 1964), pp. 254-267.

[7]K. D. Tocher, "Review of Simulation Languages," Operations Research Quarterly, Vol. 16 (June, 1965), p. 192.

[8]Naylor, Balintfy, Burdick and Chu, p. 241.

[9]Howard S. Krasnow, "Computer Languages of System Simulation," Digital Computer User's Handbook, ed. by Melvin Klerer and Granino Korn (N. Y.: McGraw Hill Book Co., 1967), Section I, p. 261.

[10]Naylor, Balintfy, Burdick and Chu, p. 241.

[11]Krasnow and Merikallio, p. 254.

[12]P. J. Kiviat and A. A. B. Pritsker, Simulation With GASP II: A FORTRAN Based Simulation Language (Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1969), pp. 1-21.

[13]Krasnow, p. 275.

[14]Naylor, Balintfy, Burdick and Chu, p. 241.

[15] Krasnow and Merikallio, p. 254.

[16] Krasnow and Merikallio, p. 259.

[17] Teichroew and Lubin, p. 737.

[18] Naylor, Balintfy, Burdick and Chu, pp. 240-241.

[19] Teichroew and Lubin, p. 732.

[20] Teichroew and Lubin, p. 738.

[21] Teichroew and Lubin, p. 738.

[22] Krasnow, p. 261.

[23] Naylor, Balintfy, Burdick and Chu, p. 241.

[24] Teichroew and Lubin, p. 732.

[25] Naylor, Balintfy, Burdick and Chu, p. 241.

[26] Naylor, Balintfy, Burdick and Chu, p. 241.

[27] Teichroew and Lubin, p. 732.

[28] Teichroew and Lubin, p. 731.

[29] Teichroew and Lubin, p. 732.

[30] Naylor, Balintfy, Burdick and Chu, p. 241.

[31] Krasnow and Merikallio, p. 254.

[32] D. Teichroew, T. D. Truitt, and J. F. Lubin, "Discussion of Computer Simulation Techniques and Comparison of Languages," Simulation, Vol. 9 (Oct., 1967), p. 244.

[33] Teichroew and Lubin, p. 732.

[34] Krasnow and Merikallio, p. 238.

[35] Krasnow, pp. 263-264.

[36] Krasnow and Merikallio, pp. 265-267.

[37] Teichroew and Lubin, p. 733.

[38] Krasnow, p. 276.

[39] Krasnow and Merikallio, p. 238.

[40] Teichroew and Lubin, p. 730.

[41]Teichroew and Lubin, p. 733.

[42]Naylor, Balintfy, Burdick and Chu, p. 241.

[43]Teichroew and Lubin, p. 733.

[44]Naylor, Balintfy, Burdick and Chu, p. 241.

[45]Krasnow and Merikallio, p. 257.

[46]Krasnow, p. 261.

[47]Naylor, Balintfy, Burdick and Chu, p. 241.

[48]Teichroew and Lubin, p. 726.

[49]Naylor, Balintfy, Burdick and Chu, pp. 242-243.

[50]Teichroew and Lubin, p. 738.

[51]Naylor, Balintfy, Burdick and Chu, p. 242.

[52]R. Efron and G. Gordon, "A General Purpose Digital Simulation-Description," IBM Systems Journal (Vol. 3, No. 1, 1964), p. 57.

[53]Naylor, Balintfy, Burdick and Chu, p. 249.

[54]Naylor, Balintfy, Burdick and Chu, p. 248.

[55]Krasnow and Merikallio, p. 256.

[56]Teichroew and Lubin, p. 731.

[57]Teichroew and Lubin, p. 733.

[58]Teichroew and Lubin, p. 726.

[59]H. Markowitz, B. Hausner and M. Karr, SIMSCRIPT: A Simulation Programming Language (Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1963).

[60]Naylor, Balintfy, Burdick and Chu, p. 279.

[61]Krasnow, p. 276.

[62]Teichroew and Lubin, p. 739.

[63]Teichroew and Lubin, p. 728.

[64]Kiviat and Pritsker, p. 18.

[65]Teichroew and Lubin, p. 726.

[66] Kiviat and Pritsker, pp. 258-263.

[67] P. J. Kiviat, R. Villanueva, and H. M. Markowitz, SIMSCRIPT II (Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1968).

[68] Tocher, p. 191.

[69] Teichroew and Lubin, p. 738.

CHAPTER III

THE DEMAND AND ENVIRONMENT SUBSYSTEM

Inputs, Outputs and Data Base Requirements

Referring to Figure 2 in Chapter I, the inputs into the D&E subsystem are broadly defined as the sales forecast, customer orders, customer mix, product mix, decision rules and management parameters. These broad categories of inputs were manipulated within the D&E subsystem to produce allocated DU, DC and regional simulated sales. Allocated customer orders served as the inputs into the OPS subsystem to be accounted for at the DU and DC stages of the model. In addition, the effect of the allocated customer orders were processed at the MCC stage plus additional sales and service information was prepared and accounted for at all levels of the model by the OPS subsystem.

The above D&E inputs and outputs were not specific enough to computer program the activities associated with their manipulation and production. The specific inputs into the D&E subsystem and the key endogenous variables associated with the subsystem activities are noted in Appendix 1. All common data base variables marked with "U" or "S" under the column labeled D&E represent the variables that were used or altered within the system. The endogenous variables, indicated by an "N" under the TYPE column, were the basis for the derived outputs of this subsystem. The exogenous variables are those variables that must be inputted into the D&E from its supporting data system through the M&C gateway. The "U," endogenous variables represented those variables that are set in one of the other operating subsystems of LREPS. As was mentioned in Chapter II, the only exogenous variables not specifically identified as coming through the M&C gateway into the common data base were those variables associated with the order file.

The organization of this chapter focuses first on the activities in the operating system of LREPS that processed its related inputs, both exogenous and from the other subsystems, and developed the outputs for the OPS subsystem. This first section is called the OPERATING SYSTEM. The specific activities associated with the D&E subsystem are:

1) DSQGEN associated with the generation of the domestic daily sales dollar quota.

2) SLSPRC associated with the generation of the allocated customer orders for the DU's in the geographic area of the DC presently being processed.

The above activities happened on a daily basis. None of the D&E activities happened only on a monthly or quarterly basis, although more sophisticated and additional routines could have been programmed to take place on other than a daily basis.

In relation to the daily LREPS activities, the pertinent variables are defined by their respective data base name, column number, if any, plus a brief description of the variable. These pertinent variables are also classified in the routine's system flowchart. The variables are classified on the basis of inputs which were defined as the LREPS common data base variables used in the routine, "local" key endogenous variables set and used only within one routine, "subsystem" common endogenous variables set and used by more than one routine in the entire subsystem, and output variables which were LREPS common data base variables set or altered in the routine.

Each of the specific LREPS D&E modelling activities are discussed in detail by relating to the respective computer system flowchart and block diagram. A detailed activity review is necessary to clarify the activity's logical location within a certain computer program or subprogram. In addition, the implications for improvement plus the results of testing and calibration for the more complicated activities are discussed.

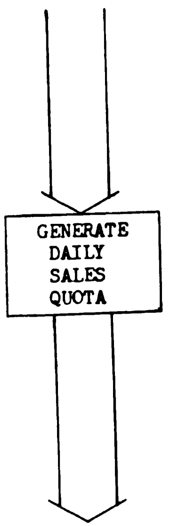
After the OPERATING SYSTEM, the next section of this chapter is the D&E's SUPPORTING DATA SYSTEM. This section presents the D&E computer activities that aided in analyzing and preparing exogenous variables for the operating system. The supporting data system computer programs are:

1) Selection of sample, actual invoices
2) Analysis of the sampled invoices
3) Analysis of the product items in the sampled invoices
4) Analysis of the product items based on annual sales
5) Development of a basis for calculating road distances
6) Sales potential correlation analysis
7) Generation of DU data
8) Selection, agglomeration and variable projection of DU's
9) Generation of the customer order file

For each program a logical block diagram of activities, the programming language(s) used, the variables or information associated with the program plus any peculiar programming problems or techniques are discussed. Implications for sophistication are also presented where it was deemed necessary because of model or program complexity.

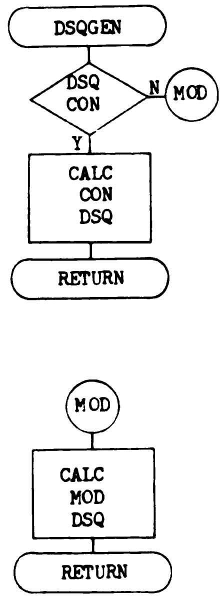
Operating System

Referring to Figures 11 to 14, the activities associated with the operating system's D&E subsystem are described graphically. In the comments section of the block diagrams, the activities associated with a particular



NAME	DESCRIPTION
INPUTS	
TDSF	Annual domestic dollar forecast
TDSFLG	Type of DSQ generated
NWKDYS	No. of simulated workdays/year
KEY ENDOGENOUS VARIABLES	
Common	
DSQ	Daily sales quota

Figure 11.--Daily Sales Quota Generation Variables



Routine to generate daily domestic sales dollar quota

Check to see if daily quota is modified variate

Calculate DSQ as a constant--annual forecast divided by number of simulated workdays

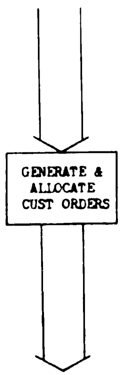
Return to Daily event activity

Modified variate generation routine

Generate DSQ as a modified variate

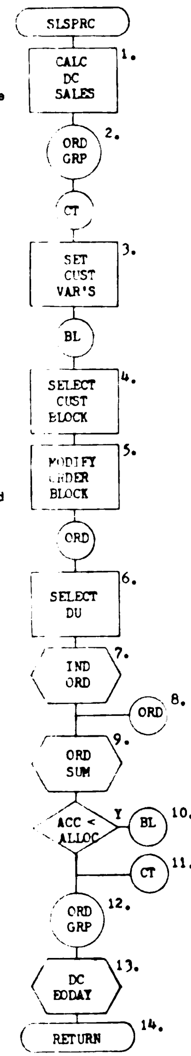
Return to DAILY event activity

Figure 12.--Calculation of Daily Sales Quota



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
ORDGRP	Group of customer orders
IREGNO	Region being processed
CSPLTP	Regional cust dollar split percentage
CSTBLK	Order blocks per cust type
NDUS	Total number of DU's being processed
DU(1)	Special customer indicator
DU(6)	DC assigned DU
DU(7)	Cumulative weighted index
DU(8)	In-solution DC pointer
DCIS(3)	Number of DU's in DC
DCIS(4)	Sales modification factor
DCIS(5)	Sum of DU weighted indices
KEY ENDOGENOUS VARIABLES	
Local	
DCSALS	DC simulated sales
ICTYPE	Cust type being processed
CSTSAL	Cust sales dollar allocation
CSTSAC	Cust sales dollar accumulator
OBM	Order block modifier
IBLKS	Number of order blocks per cust type
IPBLKS	Number of past order blocks processed
IDUNO	Demand unit being processed
Common	
DSQ	Daily domestic sales dollar quota
OUTPUTS	
IDUNO1	Demand unit selected for order
IORD	Cust order being processed
IBLOCK	Cust order block being processed
ORDGRP	Group of customer orders

Figure 13.—Normal Sales Processing Variables



D&E routine for DC normal sales processing

1. Calculate DC simulated sales as a function of forecast and sales modification factor

2. See if order group has been buffered from order file

Process sales by major customer type

3. Calculate amount of sales allocated to customer type and clear accumulator

Order block selection routine

4. Randomly select an order block from this customer's set of blocks

5. Modify order block information by factor, less than or equal to one, based upon sales allocated and accumulated

Process ten orders for customer block

6. Randomly select DU for this customer type. If invalid DU for customer type, select another DU

7. Link to OPS routine to process individual order for this DC & DU

8. Loop through all orders for order block

9. Link to OPS routine to process order block summary for this DC

10. If the accumulated sales is less than allocation, go back for another order block

11. Loop through all major customer types

12. Begin reading new order group from order file for next DC to be processed

13. Link to OPS end-of-day routine to process this in-solution DC's inventory levels

14. Return to DAILY event activity

Figure 14.—DC Daily Sales Processing Routine

subprogram are briefly discussed. In addition to the block diagram a discussion of the programming language(s) used, the LREPS variables associated with the activity, plus the identification of any peculiar programming techniques or serious programming problems are presented for each subprogram.

The first major subprogram of the D&E is the generation of the domestic daily sales dollar quota. The variables associated with this subprogram are identified in Figure 11. Three of the variables are exogenous. TDSFLG and NWKDYS were inputted at the beginning of the cycle. TDSF, the annual domestic sales dollar forecast, was inputted annually. DSQ is the generated daily sales quota which can be either a constant, a random variate, a trend modified variate, a seasoned variate or a variate based on some combination of the above. The logical activities associated with this subprogram are presented in Figure 12. This FORTRAN subprogram developed the base figure which was allocated to the in-solution distribution centers.

The other major component associated with the D&E is the normal processing of the simulated sales for each in-solution DC. There were a number of important activities associated with this subprogram. Each of the activities will be discussed in reference to the block diagram in Figure 14. The variables associated with this subprogram are listed in total in Figure 13.

Referring to Figure 14, Activity Block 1 was concerned with the generation of the DC's simulated sales. The variables associated with this activity were DSQ from the above discussed subprogram, DCIS(4) from the M&C activity associated with the development of a DC's sales forecast modification factor, DCIS(5) from the M&C subprogram associated with developing DC-DU related information, IDCNO from the M&C daily fixed-time event activity that specified the DC presently being processed, and finally DCSALS which was the output from this activity. This activity was programmed in FORTRAN according to the math model specifications.

Block 2 is a routine that was programmed in assembly language to enable the computer model to buffer the order file into the computer program for the operating system. Buffering enabled us to overlap the inputting of information while processing the remaining activities associated with the DC being processed. Presently this activity uses a very simple buffering procedure. Later a more sophisticated buffering procedure could be developed that would enable LREPS to process the order file tape at literally tape speed on the CDC 6500.

The variables associated with this buffering activity were the order group variables, ORDGRP(1) through ORDGRP(4), for all order blocks. Given that the decision had been made to work with summarized blocks of, for instance, ten or less orders, a group of order blocks was buffered in for each in-solution DC each day. An order group, which will be discussed in more detail in the SUPPORTING DATA SYSTEM of this chapter, was composed of a set number of order blocks. The type of order blocks, however, could vary between cycles. In one cycle, four blocks might have been for a certain customer type while six blocks were for another type. In another cycle, the block splits might have been four, four and two among customer types.

The total number of orders per order group was set to furnish the larger size DC's with the average number orders that they would normally process in a day. The smaller DC's would use only the required number of order blocks that they needed to meet their daily sales allocation. The usage of the order group will become clearer as the activities associated with this subprogram are developed in greater detail.

The next activity, Block 3, was programmed in FORTRAN. The output of this activity was the amount of sales to be allocated to a given customer type within the geographic area of the DC and region being processed. The activity also cleared an accumulator that enabled us to know when we reached our allocation. The variables associated with this activity were DCSALS from Block 1, CSPLTP which was an exogenous input from the D&E supporting data system, could be changed annually and specified the percentage of sales that the customer type being processed was to get from the DC's total sales allocation, IREGNO which was set in the M&C daily event, ICTYPE, CSTSAL and CSTSAC which were set in this block and outputted to the following activities.

Block 4 was the random selection of one of the order blocks for a given customer type from the DC's order group. This activity was programmed in FORTRAN and used a pseudo-random generator called RANF as the basis for selecting a block. RANF generated a pseudo-random number based on a 0,1 uniform probability distribution. The endogenous variables associated with this activity were IBLOCK which specified the block that was randomly selected, IBLKS that denoted how many blocks had been allocated for this customer type, IPBLKS which told how many of the past customer blocks had been processed and ICTYPE which identified the customer type being processed. The one exogenous input was the vector variable CSTBLK which specified the number of blocks per customer type for all specified major customers. CSTBLK was set cyclically.

Activity Block 5 was associated with what was called an order block modifier. This modifier allowed the computer model to approximate very closely the allocated sales for a customer type. This routine modified a given order block's contents if the order block's total dollar sales were greater than the difference between the allocated customer sales and the sales dollars already accounted for by previous customer orders. The variables associated with the activity were OBM which contained the order block modifier, CSTSAL and CSTSAC from Block 3, ORDGRP(1) which is an exogenous input from the order file and an order block's ORDGRP(1) through ORDGRP(4) if the contents of the order block must be modified. This activity was also programmed in FORTRAN.

Block 6 dealt with the random selection of a DU within the DC's geographic area of responsibility. A small function, FORTRAN subprogram was developed to accomplish the random selection of a DU. The function was essentially based upon a Monte Carlo selection procedure. The basis for the Monte Carlo selection was the DU's weighted indices for sales allocation. The weighted indices for all DU's in a DC area were transformed into a cumulative basis with which a random number, generated from RANF, was used to select randomly the DU that was to receive the current, allocated customer order.

After selecting the DU, one additional test was performed. If the selected DU had certain special customers given the special customer type was being processed, then the DU was acceptable. If unacceptable, another DU was selected. This cycling procedure continued until an acceptable DU was selected.

The variables associated with these activities were IDUNO and IDUNO1 which were concerned with the DU being processed, NDUS which was the total number of DU's being processed, DU(1) which specified if a certain DU had any special customers, DU(6), DU(7), DU(8), DCIS(2), DCIS(3), DCIS(5) which were the basis for the random selection of a DU and were set in the M&C subprogram associated with DC-DU link information, and IDCNO which was set in the M&C event activity associated with normal daily activities.

Block 6 could be the subject of further sophistication that considered other factors besides a DU's weighted index. Distance, past sales, past service levels and reliability plus other variables at a DU stage could be used as separate or joint bases of random selection.

Block 7 was the means of joining the D&E subsystem with the OPS subsystem. The OPS subsystem has a subprogram called INDORD that processes the allocated customer order by accounting for its dollars and weight at the DU level. This OPS subsystem routine, which also generated the required service times, will be discussed in detail in the next chapter.

Block 8 is a FORTRAN activity to check if all orders in an order block had been processed and allocated. If all orders had not been processed, another DU was selected and an order allocated to it. This activity used the endogenous variable IORD which identified how many orders had been processed out of the total number in the block.

The next block enabled the OPS subsystem to process, at the DC level, the product detail and all summarized sales information of the order block whose dollars and weight had just been allocated to the selected demand units. This OPS subprogram, called ORDSUM, will also be discussed in detail in the next chapter.

Blocks 10 and 11 were used to check the amount of dollar sales accumulated for the customer type being processed against its amount of allocated sales. If the sales dollars accumulated were not within a half percent of the allocation, then another order block had to be randomly selected. If the sales allocation had been attained, then the subprogram went on to the next activity block. These blocks are FORTRAN activities which use the endogenous variables CSTSAC, CSTSAL and ICTYPE and the exogenous variable ORDGRP(1) from the order file.

Block 12 was the activity that initiated the reading of another order group for the next DC to be processed. The activity was an assembly routine that used all the variables in ORDGRP. This activity could also be sophisticated along with the earlier mentioned buffering activity in Block 2.

Block 13 was the connection to the OPS subsystem associated with the processing of the DC end-of-day activities. These activities were processed while a new order group was read. This subprogram, called DCEODAY, will be discussed in detail in the next chapter.

Block 14 was a return to the M&C daily event that called this normal sales processing routine for each processed DC.

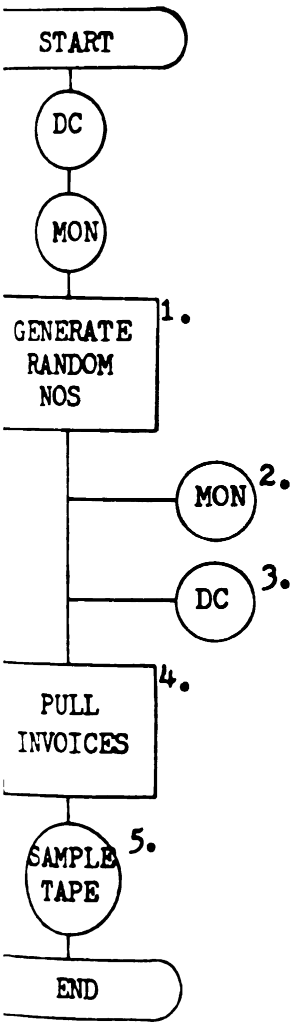
Supporting Data System

Each of the following computer activities is identified and discussed within the framework of the LREPS D&E Supporting Data System flowcharts.[1] The non-computer activities have not been discussed unless additional understanding of a computer activity was gained by describing briefly a related, non-computer activity.

The supporting computer programs are described by presenting a block diagram of the logical activities taking place within the program, the programming language(s) that were used, the operating system exogenous variables that are directly or indirectly associated with the program, other information associated with the program plus any peculiar programming techniques or problems.

Figure 15 identifies the activities associated with the selection of sample invoices from a firm's annual invoice file. This program fits within the framework of determining a firm's major customer types. Preliminary analysis of annual data had pointed out possible major customer types. Based on a stratified random sample, which is indicated in Blocks 1 to 3 in Figure 15, additional analysis was used to support the initial customer type findings. Block 4 was the pulling of the selected invoices from the annual file. Block 5 was the preparation of the "Sampled Invoice" magnetic tape.

The specific input information associated with these activities were the invoice variables that were



Selection of sample invoices from annual invoice file

Select a random sample from each of the present DC's invoice files

Select the above sample stratified on selected months

Generate invoice numbers to be selected from present DC-monthly files

Loop through all months to be sampled

Loop through present DC orders

Pull invoices for the selected major customer types

Prepare the "Sampled Invoice" tape

End of activities

Figure 15.--Selection of Sample Invoices

split between the line item detail that contained product information and the invoice summary data that contained distribution center identification, customer type identification, invoice number, geographic identification, invoice sales summaries, and the shipment dispatch date. An additional input into the selection of invoices was the random numbers generated for the specific invoices to be selected.

The random numbers were generated from a pseudo-random number generator that was available on a time-shared computing system. The preparation of the "Sampled Invoice" tape was an assembly card-to-tape routine. The output of this program was the randomly selected invoices with all the invoice detail described above. The operating system exogenous variables affected by this routine was CSTBLK, CSPLTP and ORDGRP.

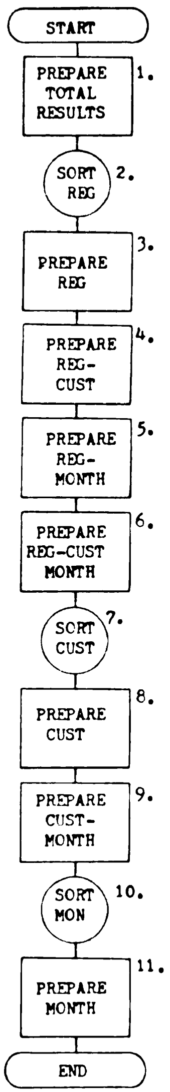
After the sample invoices had been selected, the analysis of these invoices followed the block diagram in Figure 16. This "Sampled Invoice Analysis" actually supported several areas of analysis. Block 1 analyzed the invoice summary information at a <u>total</u> level to identify the overall sample means, standard deviations, number of sample points and range of values for each item in the invoice sales summary records. Blocks 2 to 6 were concerned with the generation of the above statistical information based on a major sort of a firm's sales regions and on minor sorts of customer type and

and month. Thus this analysis supported the major customer analysis. Bocks 10 to 11 analyzed only the effect of monthly sales across the nation. This last analysis was to support the daily sales analysis, regional analysis and domestic daily sales analysis.

Based on the information generated from this FORTRAN program, decisions were made concerning different areas of supporting systems analysis. The output was only printed and subjected to manual examination. The operating system variables affected by this program are CSTBLK, CSPLTP and TDSFLG.

In analyzing a firm's product items, two supporting system programs were designed. The first program analyzed the product item detail based on the sampled invoices. The line item detail records of the "Sampled Invoice" tape were analyzed in the sequence of the logical activities shown in Figure 17. A separate listing for each type of sales information was prepared which showed the rank, the absolute and cumulative percentages of total sales per product in the tape. An additional listing showed the frequency with which each product appeared in the sample invoices.

Figure 18 shows the activities associated with the second computer program involved in the product item analysis. The basis for this second program was an analysis of a firm's total annual product sales. The input into the program was a punched card for each product that contained different sales information. A separate listing for each



Analysis of sampled invoice file

Prepare invoice results for complete sample with no minor breakdowns

Sort tape for regional analysis with minor sorts on customer type and months of year

Prepare regional results with no minor breakdowns

Prepare regional results with minor breakdown on customer type

Prepare regional results with minor breakdown on months of year

Prepare regional results with minor breakdowns on customer and month

Sort tape for customer analysis with minor breakdown by months of year

Prepare strictly customer results with no minor breakdowns

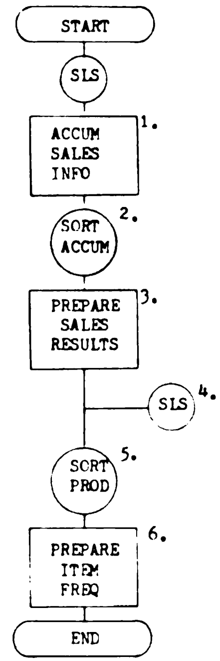
Prepare customer results with minor breakdown on month

Sort tape for monthly analysis with no minor breakdowns

Prepare monthly results only

End of analysis

Figure 16.--Sampled Invoice Analysis



Product item analysis based on "Sampled Invoice" tape

Prepare product item reports for each sales type

Accumulate the sales information for the type of sales being processed

Sort the product items in descending order based on accumulated sales information

Prepare product item results

Loop through all types of sales information

Sort the product items based on product code

Prepare the report showing how frequent an item appeared in the invoice tape

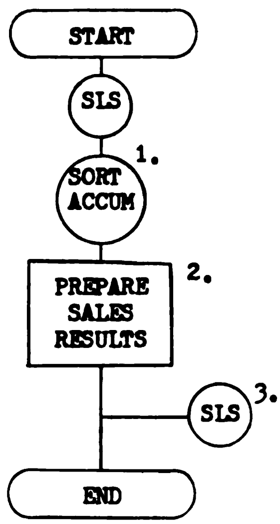
End of analysis

Figure 17.--Product Item Analysis--No. 1

type of sales information was prepared that showed the rank, the absolute and cumulative percentages of annual sales per product.

Both of these programs in the product item analysis were programmed using FORTRAN. The output listings from the runs were used in preparing the exogenous inputs TPDEM(2), ITNSP, ITNPC, PRCT(2), PRCT(3), PRCT(4), CGSCS, CUBCS, WTCU. These operating system inputs were associated with the tracked products that would be used for inventory management, control and costing analyses plus provide some of the basis for the generation of pseudo-customer orders. Each of these areas of analysis will be discussed in more detail later. New product and different order characteristic information were also merged with the results of this analysis.

Figure 19 identifies the logical activities of a program called the "Distance Check" routine. Given that the basis for a demand unit had been selected, a program was needed to check out a routine for calculating the road distances between the hub cities of these DU's. A routine for calculating the spherical, arc distances between two points was available and a road conversion factor was needed.[2] The program thus served two purposes. One was to generate a good road conversion factor and the other was to generate road distances for a comparison to actual distances.



Product item analysis based on annual sales

Prepare product item reports for each sales type

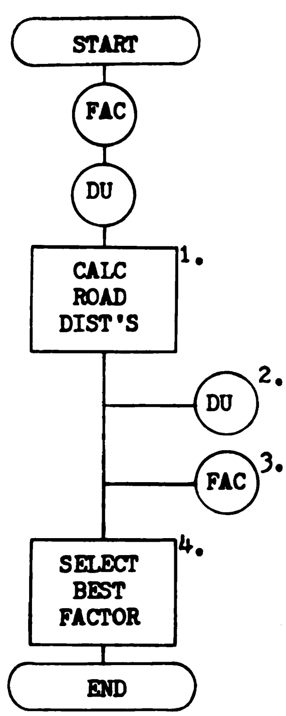
Sort the annual sales information for each product in descending order

Prepare product item results

Loop through all types of sales information

End of analysis

Figure 18.--Product Item Analysis--No. 2



Distance check routine based on a sample of DU points

Use a range of highway conversion factors

Process all combinations of DU sample points

Calculate the road distance between selected two sample points

Loop through all DU combinations

Loop through alternative road conversion factors

Select best road conversion factor by comparing calculated to actual road distances for sample DU combinations

End of analysis

Figure 19.--Distance Check Routine

The input into this FORTRAN program was a punched card containing the latitude and longitude for each sample DU's hub city, a card deck of actual road distances between the sample DU's, and a card specifying the range of road conversion factors to be considered. The output included the best road conversion factor plus the standard error between actual and calculated road distances and the array of calculated road distances for the selected conversion factor. The exogenous variables affected by this routine are the rectangular, converted X and Y coordinates of the DU's and of the potential DC's latitudes and longitudes. Specifically, the variables are DU(2), DU(3), DCP(1), and DCP(2).

Figure 20 identifies the use of a system library, statistical program for doing univariate correlation analysis. The purpose of this program was to analyze the relationship between a firm's sales and alternative independent variables. The card input to this program was composed of one card for each of the firm's geographic, sales control areas. Each card contained geographic identification information, annual firm sales and the relevant, alternative independent variables.

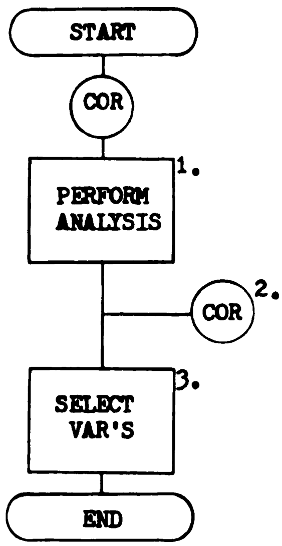
The printed output of the computer run contained the selected independent variables and the correlation coefficients between the firm's sales and the independent variables which were significant at the selected level of statistical significance.

The operating system exogenous variables specifically affected by this program were DU(4), the weighted index of DU sales potential, and DCIS(5), the basis for DC sales potential.

Figure 21 identifies a FORTRAN program that was used to generate the basic information needed at a DU level. The input into this program was sourced from three different areas. The first source of information was customer information summarized for the selected, basic demand units. Such information as the DU identifying code, summarized sales information, number of major customer types plus the number of direct competitors was compiled and summarized at the DU level.

The second source of information was DU identification information. Such items of information as the DU identifying code, the DU's hub city for denoting the point of demand concentration, the DU's latitude and longitude, the type of DU--domestic or non-domestic, and the DU agglomeration basis to an even further reduced number of DU's. This last piece of information was concerned with the reduction of counties to groups of counties, or zip sectional center areas to a reduced number of geographic areas. This reduced level of detail, if possible, allowed more efficient computer processing times.

The third source of information was the independent variable information. This source of information



Correlation analysis based on alternative independent variables and firm sales

Process each alternative independent variable

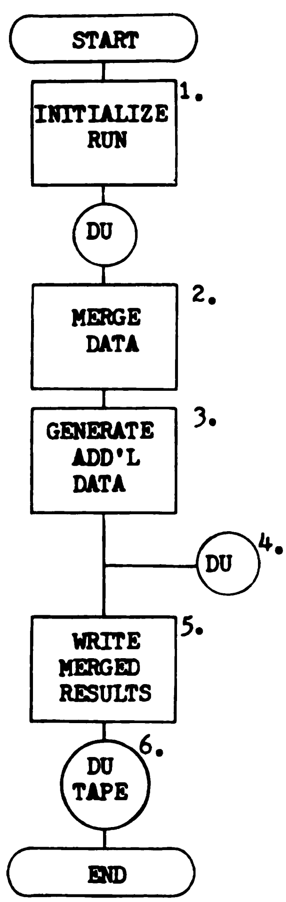
Perform correlation analysis between firm's sales and the alternative independent variable

Loop through all independent variables

Select independent variable(s) with significant correlation coefficients

End of analysis

Figure 20.--Correlation Analysis



Routine to merge and generate DU internal firm and environmental data

Initialize variables and arrays

Process all information for each customer demand unit

Merge firm and external environmental data

Generate additional summary and statistical information

Loop through all customer demand units

Write merged customer results based on DU's

Develop DU tape

End of analysis

Figure 21.--DU Data Generator

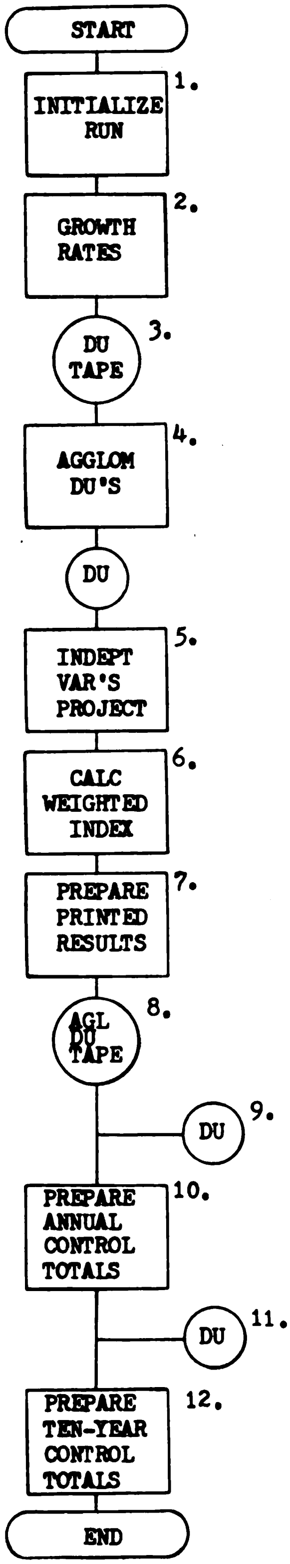
had a DU identifying code and the selected independent variables' base year totals. These base totals were used for making projections over the next ten years.

As shown in Block 2 of Figure 21, these three sources of information were merged into one file of information. In addition, Block 3 indicates that additional information was generated within the program and also merged into the original file. Finally the merged results were printed and wrote on the basic DU tape. The variables that were contained in the DU tape were DU(1), DU(2), DU(3), DU(4), DU(9) and DU(10).

Figure 22 presents the logical activities associated with 1) the selection of certain DU's in order to process only portions of a firm's total geographic sales area, 2) the further agglomeration of the basic DU's to a reduced number of manageable DU's and 3) the projection of the independent variables as the basis for preparing the DU weighted indices of sales potential for each of the years in the planning horizon.

Blocks 3 and 4 identify the activities associated with the selection and agglomeration of the basic DU's. Blocks 5 to 9 illustrate the activities of preparing the annual DU data. Blocks 10 to 12 are associated with the activities of preparing control totals for the run plus looping through a maximum ten-year planning horizon.

The inputs into the run were the basic DU tape, the selected independent variable growth rates and the selected



Routine to prepare ten-year, agglomerated DU tape

Initialize run variables and array

Initialize the selected independent variables' growth rates

Read in desired first-level demand units from demand unit tape

Agglomerate selected DU's to reduced number of DU's

Process selected, agglomerated demand units

Make projections of independent variables per demand unit

Calculate the independent variables' combined weighted index of sales potential

Prepare listing of agglomerated demand unit results for year in process

Develop agglomerated, second-level demand unit tape

Loop through all second-level demand units

Prepare annual control totals on firm and environmental data

Loop through all ten years of planning horizon

Prepare ten-year control totals

End of analysis

Figure 22.--DU Selection, Agglomeration and Projection

basic DU's to be processed in the operating system. The outputs of the run were the printed listing and the merged, selected DU tape. This was a FORTRAN program that affected the same exogenous variables as in the DU Data Generator plus variable NDUS.

Figure 23 identifies the logical activities associated with the development of the order file that was read into the D&E subsystem's operating system. As was discussed earlier, the decision had been made to use blocks of customer orders. The basic purpose of this program, called the Order File Generator, was to prepare the stream of customer orders that were allocated to the DU's and the DC's by the D&E subsystem of LREPS. One of the basic problems associated with this stream of inputs was to keep it to a feasible size. Even after the decision had been made to use blocks of orders, the amount of time that was required to process these blocks in either a sequential or random manner over a maximum of ten years was still impractical. The run time from initial tests using either tape or disk files was estimated to be approximately four hours on the CDC 6500. The size of the files that contained this information was also seen to be quite large. For a ten-year cycle, it was estimated that six or seven full reels of order blocks would not be unlikely.

Therefore, the decision was made to develop "groups" of order blocks. These groups would be composed of a

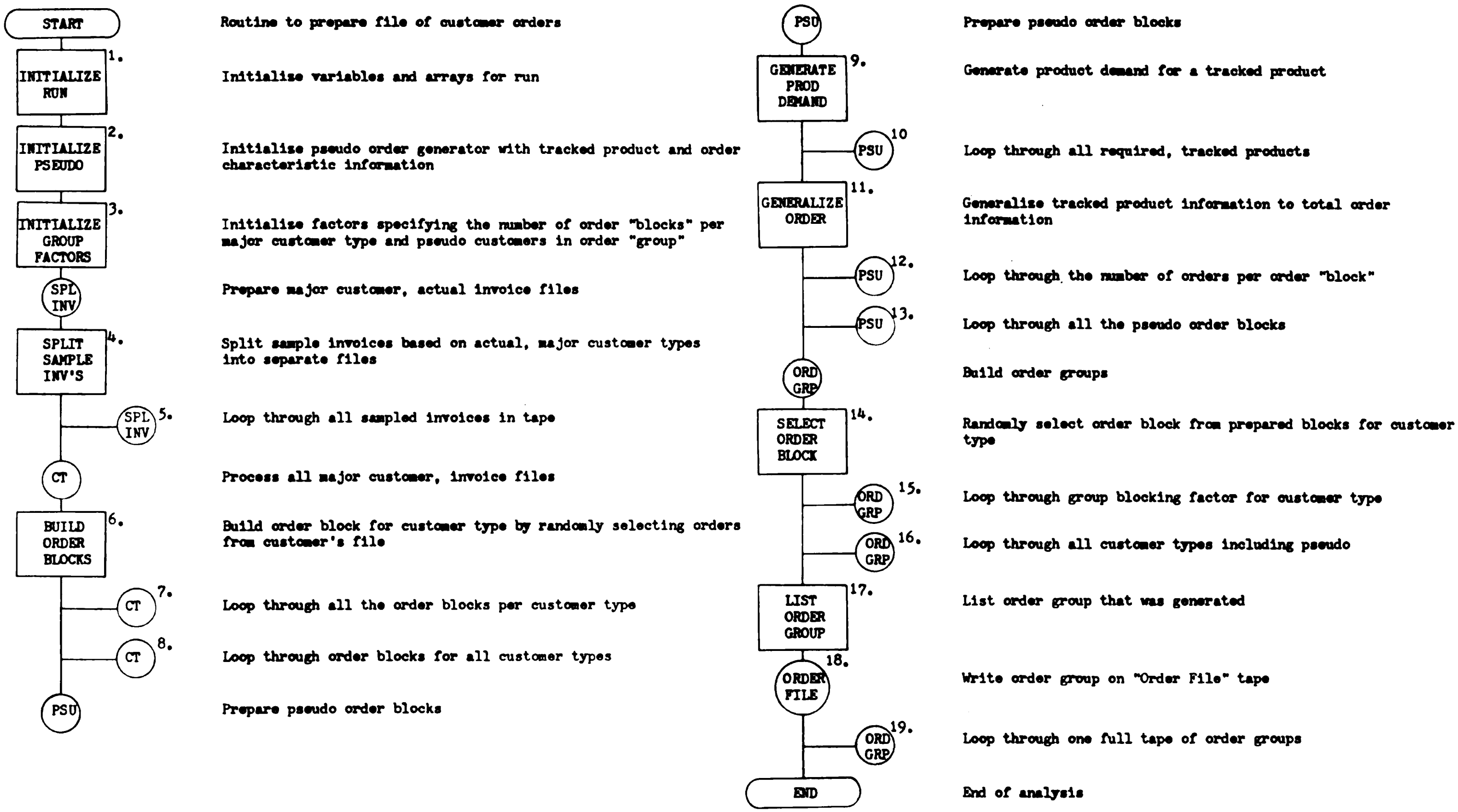


Figure 23.--Order File Generator

specific number of order blocks for each major customer type. The exact number of blocks would be based upon the amount of DC daily sales that a customer type would most likely account for over the entire planning horizon. A new group of order blocks was read for each DC in the normal day's processing.

Figure 23 shows how these order blocks and groups of order blocks were developed. Blocks 1 to 3 of Figure 23 are related to the initialization of the variables and arrays in the run. The basic input variables to the run were 1) the "Sampled Invoice" tape including both the line item and invoice summary detail, 2) tracked product information that was used to not only select the tracked products from all the products in the "Sampled Invoice" tape but also to provide the basis from which orders could be randomly generated and 3) the order group blocking factors by major customer type including the pseudo or randomly generated customer types.

Blocks 4 and 5 show the splitting of the "Sampled Invoice" tape into separate major customer type files. Blocks 6 to 8 show the random selection of a specific customer type's orders and summarizing these selected orders into blocks of orders. Each customer type was processed separately.

After the sampled invoices had been blocked into the required number of blocks, the order blocks for the pseudo-customer types were developed. Blocks 9 to 13 used individual tracked product detail composed of the model product code, the inventory category to which this product belonged, average sales volume information per order, and the assumed probability distribution type plus its parameters. The tracked product's inventory category, whether it be based on traditional ABC analysis or some other categorical basis, required data concerning the number of the tracked products in the category plus the total number of products in the category of which it was representative. The number of inventory categories was also specified.

After the tracked product information for a customer order was generated, utilizing readily available, pseudo-random variate generators for theoretical probability distributions, the information for the tracked products was generalized to the total product line for the customer order. This generalized order was then added to the order block for that pseudo-customer type. All pseudo order blocks were generated for each pseudo-customer type separately.

Blocks 14 to 19 represent the random selection of the order blocks for each customer type and the merging of these order blocks with those of the other customer types to develop an order group. Block 19, which utilized an assembly tape-write routine, shows that one full, 2400 foot reel of magnetic tape was prepared of order groups. This order file was the input into the operating system's

D&E subsystem. The tape was read by the D&E until an end-of-file occurred. The tape was then rewound and read again. This rereading of the tape was assumed to be a safe activity since the probability that a DC would process the same order group would be very, very slight.

Besides being written on the "Order File" tape, the order file was listed on the printer. The output of this program also included a tape file of the individual actual and/or pseudo-customer orders that had been summarized into order blocks. This tape file was later analyzed by a Measurement subsystem supporting computer program.

The operating system exogenous variables affected by the Order File Generator were CSTBLK and all the variables in the ORDGRP. The program was written in FORTRAN except for the assembly, tape-write routine for the order file.

CHAPTER III--FOOTNOTES

[1]Bowersox, *et al.*, Monograph.

[2]R. S. Underwood and Fred W. Sparks, *Analytic Geometry* (Boston: Houghton Mifflin Company; Cambridge: The Riverside Press, 1956), pp. 221-228.

CHAPTER IV

THE OPERATIONS SUBSYSTEM

Inputs, Outputs and Data Base Requirements

The OPERATIONS (OPS) subsystem was responsible for the processing of the allocated customer orders at the DU and DC stages and effecting whatever changes took place within the DC and MCC stages of the physical distribution (PD) system as a result of the processing of the customer orders. Stages 2 and 3 in Figure 1 of Chapter I were the areas for which customer orders were accounted. Stages 1 and 2 of the same figure were the areas in which the PD activities of transportation, communications, inventory control, unitization and facility commitment were effected. The exception to the above was the MCC stage which did not take into consideration inventory control and facility commitment.

The OPS subsystem was, therefore, responsible for processing the main physical distribution activities and developing the outputs that would serve as the bases for calculating the measures of sales, costs and service by the MEAS subsystem.

The main activities within the OPS subsystem were conceptualized as happening on both a fixed- and variable-time basis. The fixed-time activities were grouped under four areas:

1) The processing of the individual customer order which included the allocation and accounting of specific sales information at the DU level plus the generation of and accounting for customer service statistics at the DC level;
2) The processing of the order block's summarized sales and product detail information at the DC level;
3) The DC end-of-day activities in which the DC's inventory levels are checked with the pertinent inventory management policy variables to see if reorders for product replenishment should be dispatched from the DC to the supplying MCC's;
4) The DC end-of-day activities to see if any shipments must be dispatched to it from the supplying MCC's.

The variable-time activities that did not happen every day for the DC-MCC links were grouped under two areas:

1) The arrival of a multiple-product reorder at a MCC from a DC;

2) The arrival of a shipment at a DC from a supplying MCC to replenish inventoried products.

The above are the major activity breakdowns associated with the OPS subsystem. Because two of the breakdowns happened at the end of day, fixed-time activities 3 and 4 from above were grouped into one major end-of-day activity subprogram. Therefore, the OPERATING SYSTEM is organized around five major computer subprograms. These five subprograms were named INDORD for processing the individual customer order at the DU and DC stages, ORDSUM for processing the order block summary at the DC level, DCEODAY for processing the DC and MCC end-of-day activities, MCORAR for processing the arrival of an order at the MCC level, and DCSHPAR for processing the arrival of a shipment at the DC level.

Again, as in the D&E subsystem, a system flowchart of the inputs, key endogenous variables and outputs, a logical block diagram of activities, the programming language(s) used, plus any peculiar programming techniques and problems are presented and discussed for each of the above activities.

The SUPPORTING DATA SYSTEM computer activities will focus on the preparation of important DC-MCC product link information. This was the only computer, supporting activity associated with the OPS subsystem. As in the D&E subsystem, this program will be discussed within the framework of the LREPS Supporting Data System flowcharts.

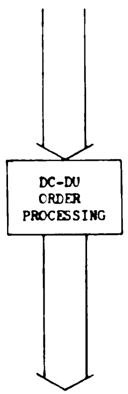
Operating System

The first component to be discussed is INDORD which is the individual processing of the customer order. Figure 24 presents the common data base variables associated with the activities in this subprogram. Figure 25 presents the logical block diagram linking the component's activities.

Activity Block 1 generated the order transmission time (T1) from the DU to the DC to which it was assigned. The activities for determining the length of time it took an order to get to the DC were the generation of a constant or average service time and then adding to this constant an element of variability. For instance, a DU may have been a constant 2 days plus a variable 0, 1 or 2 days from the DC. This resulted in a range of T1 values from 2 to 4 days.

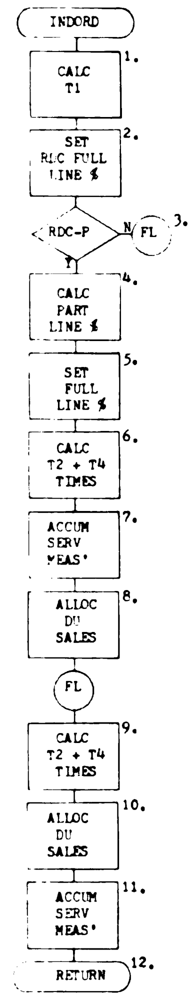
The constant service time was retrieved from data base variable DU(5) and was initially determined in a M&C routine associated with the preparation of DC-DU link information. How the constant time is determined is discussed in Chapter VI on M&C activities. The retrieval of the constant time was executed using EXTENDED FORTRAN instructions for the CDC 6500 which allowed the shifting out of the time from the packed computer word.

The variable element, if it was desired, utilized a Monte Carlo selection procedure that returned a value of 0, 1 or 2. The routine was both a FORTRAN and assembly



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
IDUNO1	Demand unit selected for order
IORD	Cust order being processed
IBLOCK	Cust order block being processed
ITNPC	Total no. of inventory categories
DCP(7)	Cust serv time variance pointers
DCP(8)	DC type + PDC assignment
PARM	Serv time variance functions
SDSM	T2 max dollar daily shipment for FLC
CSDP	Percent of FLC daily cust shipments
DCCAPC	Capacity constraint by DC size
DCIS(8)	DC size indicator
DU(5)	Cust serv time ring nos.
WTCU	Avg weight per cas unit
PRCT(1)	Partial line inv category indicator
PRCT(2)	Beginning model product code
PRCT(3)	Ending model product code
ORDGRP	Order block from group
KEY ENDOGENOUS VARIABLES	
Local	
IDCTEMP	Temporary storage for PDC no.
Common	
RDCPC	Percent of order for full-line DC
PLPC	Percent of order for RDC-P
OUTPUTS	
DU(13)	QTD $ sales for full-line-DC
DU(14)	QTD $ sales for RDC-P
DU(15)	QTD wt sales for full-line-DC
DU(16)	QTD wt sales for RDC-P
OCTDIS(1)	NOCT sales $ proportions
OCTDIS(2)	NOCT sales order proportions
DCIS(9)	Avg NOCT accumulators
DCIS(10)	Std dev NOCT accumulators
DCIS(11)	Avg OBT accumulators
DCIS(12)	Std dev OBT accumulators

Figure 24.--Individual Order Processing Variables



Routine to process an order for a specified DC & DU

Calculate the order transmission time(T1) from DU to majorly assigned DC

Set RDC full-line percent equal to one

Check to see if majorly assigned DC is partial line

Calculate partial-line DC percent of weight based on inventory categories

Set full-line percent equal to one minus RDC-P percent

Calculate customer service times T2 + T4 for partial-line DC

Accumulate information base for calculating measures of customer service

Alloc DU sales dollars and weight for partial-line DC

Full-line DC processing routine

Calculate customer service times T2 + T4 for full-line DC

Allocate DU sales dollars and weight for full-line DC

Accumulate information base for calculating measures of customer service

Return to D&E normal sales processing routine

Figure 25.--Customer Order Processing

routine that utilized data base variables DCP(7) and PARM. DCP(7) contained the pointer to PARM, a common set of T1 variance parameters that could be utilized by several DU-DC links. This activity block also used data base variables IDUN01 and IDCN01 which are identified in Figure 24.

Activity Block 2 initialized data base variable RDCPC to one. This variable was used to allocate part of a customer's order to a PDC if the majorly assigned DC was an RDC-P. If the DC was a full-line DC, then the entire order was allocated to the DC. The basis for allocation is discussed below.

Block 3 tested to see if the majorly assigned DC was a RDC-P. If it was an RDC-P, then Activity Blocks 4 to 8 were executed. If the DC was a RDC-F or PDC, then the routine skipped to Activity Block 9. Both Blocks 2 and 3 were FORTRAN instructions that used data base variables RDCPC, IDCN01 and DCP(8).

If the majorly assigned DC was an RDC-P, then activity Blocks 4 to 8 identify the activities of splitting the order between the RDC-P and its assigned PDC. The basis for the allocation was to accumulate the weight for the RDC-P, inventory categories as a percent of the total weight for all tracked products and their respectively assigned inventory categories. To accomplish the setting of this partial-line order percentage, PLPC, a FORTRAN activity was developed.

The data base variables that were used were WTCU, PRCT(1) which indicated whether an inventory category of tracked products was an RDC-P's or a full-line DC's, PRCT(2), PRCT(3), ORDGRP(4) which identified the individual product detail summarized for the orders in the order block, IORD, IBLOCK, ITNPC and IDCTEMP which temporarily stored the code of the PDC assignment for the RDC-P. IORD was also the key to the calculation of the PLPC since the percentage only had to be calculated for the first order to be allocated from the order block. PLPC was then constant for all orders in the order block. IDCTEMP was specifically used to allow the use of a common set of FORTRAN statements to update all variables associated with both the RDC-P and PDC partial order shipments.

Activity Block 5 adjusted RDCPC to the percentage of the order to be shipped from the PDC. This block used a basic FORTRAN instruction.

Activity Block 6 calculated the order processing and preparation time (T2) for the RDC-P plus the transit time (T4) from the RDC-P to the selected DU. The calculation of T4 essentially followed the same procedure used to generate T1. The data base variables utilized for this task were IDUNO1, DU(5) which contained the constant transit time for this link, DCP(7), PARM and IDCNO1 for the variable time element. The calculation of T2 used a separate FORTRAN function subprogram that basically followed these steps:

1) Using DCP(7), PARM and IDCNO1 generate a variable time element of 0, 1 or 2;
2) If the DC is within a certain percentage of its capacity constraint, contained in DCCAPC, add an additional day to each order after this capacity limitation was reached.

The T2 routine also used IDCNO, IDUNO1, DCP(8) and DCIS(8) which indicated the size of the DC.

Activity Block 7 accumulated the information that was needed by the MEAS subsystem to calculate the measures of service for the DU and RDC-P link. The data base variables associated with this FORTRAN activity were IDCNO, IBLOCK, DCIS(9), DCIS(10), DCIS(11), DCIS(12), PLPC, ORDGRP(1), OCTDIS(1) which contained the necessary information to determine the amount of sales dollars that were delivered to the DC's customers within a certain number of normal order cycle time (T1 + T2 + T4) days, and OCTDIS(2) which was basically the same as OCTDIS(1) except that customer orders was the basis for accumulation. The concept of the total order cycle time (OCT) is discussed later.

Activity Block 8 was a basic FORTRAN activity that allocated sales information to the DU from the RDC-P. The utilized data base variables were IDUNO1, IBLOCK, ORDGRP(1), ORDGRP(2), DU(14), DU(16), and PLPC. The amount of the order block allocated to the DU was one divided by the number of customer orders, summarized in an order block, times PLPC.

Blocks 9 to 11 were the activities needed to allocate the sales information and generate the necessary customer service times from the full-line DC to the DU. If the majorly assigned DC was either a RDC-F or PDC, then Blocks 4 to 8 would not have been used, RDCPC would have been one and the full customer order would have been allocated in Blocks 9 to 11.

Activity Block 9 calculated the customer service times T2 and T4. T4 was calculated as described above while T2 was essentially calculated the same as above except that an additional test was made to see if a PDC was being processed. If a PDC was being processed, then it was assumed in the modelling of the activities of a PDC that the amount of goods being shipped from the PDC warranted the possibility of pooled or consolidated shipments to the individual DU's. Therefore, a procedure was developed for the scheduled shipment dispatch of a percentage of the customer orders on a pooled basis. If the dollar amount of a customer order was less than a certain amount (SDSM), then a random number, generated from RANF, was used as the basis for determining whether it was one of a specified percentage (CSDP) of the orders that received daily shipment dispatch. The remaining orders less than SDSM plus all orders greater than SDSM went on an approximated, dispatch schedule of Mondays and

Thursdays of simulated time. Besides the above variables, these FORTRAN and assembly routines used IDUNO1, IDCNO, DU(5), DCP(8), PARM, DCP(7), DCCAPC, DCIS(8) and IDCTEMP.

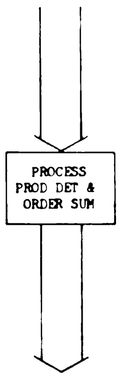
Activity Block 10 allocated the sales information to the DU from the full-line DC. This activity used basic FORTRAN instructions plus data base variables RDCPC, IDUNO1, IBLOCK, ORDGRP(1), ORDGRP(2), DU(13) and DU(15).

Activity Block 11 accumulated the information to calculate the measures of service for the DU and full-line DC links. The activity was FORTRAN based and used data base variables RDCPC, IDCTEMP, IDCNO, IBLOCK, ORDGRP(1), OCTDIS(1), OCTDIS(2), DCIS(9), DCIS(10), DCIS(11) and DCIS(12).

Figure 26 shows the important variables associated with the OPS subsystem routine ORDSUM. This routine's purpose was to process the order block's summarized sales information and product detail at the DC level. Figure 27 shows the block diagram of activities for this routine.

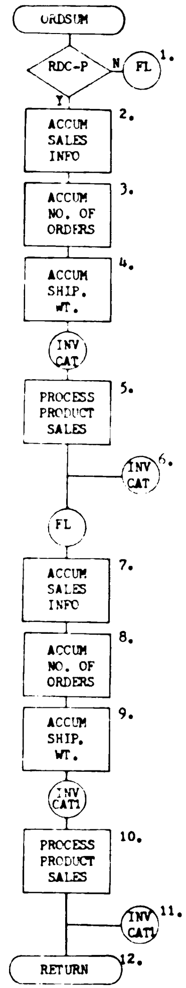
Block 1 was a check to see if the majorly assigned DC was a RDC-P. If it was an RDC-P, then the order block summary had to be allocated between the RDC-P and the PDC. If it was a RDC-F or only a PDC, then the entire order block was accounted for at only one DC via Activity Blocks 7 to 11. The data base variables utilized were IDCNO1 and DCP(8).

Activity Block 2 accounted for the portion of the order block's sales information that was allocated to



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
IBLOCK	Cust order block being processed
ORDGRP	Order group from block
DCP(8)	DC type plus PDC assignment
DCWB	DC weight breaks
ITNPC	Total no. of inventory categories
PRCT(1)	Partial-line inv category indicator
PRCT(2)	Beginning model product code
PRCT(3)	Ending model product code
KEY ENDOGENOUS VARIABLES	
Local	
IDCTEMP	Temporary storage for PDC no.
Common	
RDCPC	Percent of order for full-line DC
PLPC	Percent of order for RDC-P
OUTPUTS	
DCIS(16)	QTD dollar sales
DCIS(17)	QTD weight sales
DCIS(18)	QTD cube sales
DCIS(19)	QTD case sales
DCIS(20)	QTD line sales
DCIS(21)	QTD order sales
WTACUM	DC shipment weight accums
PRDC(4)	Inventory on hand

Figure 26.--Order Summary Processing Variables



Routine to process order block summary at a DC level

1. Check to see if majorly assigned DC is partial line
2. Accumulate sales information at partial-line DC level
3. Accumulate orders for RDC-P
4. Accumulate order block's shipment weight in correct weight category

Process RDC-P's inventory categories

5. Reduce products' inventories-on-hand by products' sales units from order block
6. Loop through all inventory categories for RDC-P

Full-line DC processing routine

7. Accumulate sales information at full-line DC level
8. Accumulate ten orders at full-line DC level
9. Accumulate order block's shipment weight in correct weight category

Process full-line DC inventory categories

10. Reduce products' IOH by products' sales units from order block
11. Loop through all inventory categories for full-line DC
12. Return to D&E normal sales processing routine

Figure 27.--Order Block Summary Processing

the DC based on the partial-line percent, PLPC, calculated in INDORD. This FORTRAN activity used data base variables IDCNO, IBLOCK, ORDGRP(1), ORDGRP(2), ORDGRP(3), PLPC mentioned above plus DCIS(16) through DCIS(20).

Activity Block 3 accumulated the number of orders that were processed from the order block at the RDC-P. The assumption had been made that two orders would be accumulated for each customer's order received at the RDC-P. Therefore, one order was accumulated at the RDC-P plus one was accumulated at the RDC-P's associated PDC. Also the assumption had been made that only one PDC would serve the same DU as a RDC-P. The data base variables associated with this activity, which was programmed in FORTRAN, were IDCNO and DCIS(21).

Block 4 accumulated the order block's shipment weight, which was allocated to the DU's, in the correct customer weight category. The correct weight category was determined by taking the ratio of one over the number of customer orders in the order block multiplied by the total weight in the order block, ORDGRP(2), times PLPC. The total weight in the order block, ORDGRP(2), was then added to the selected category. The other data base variables utilized in this FORTRAN activity were IDCNO, IBLOCK, DCWB and WTACUM.

The next two activity blocks subtracted the order block's summarized product sales detail from the RDC-P's inventories-on-hand for all tracked products and their respective inventory categories. These activities were programmed in FORTRAN and used data base variables IDCNO, IBLOCK, PRCT(1), PRCT(2), PRCT(3), ITNPC, PRDC(4), ORDGRP(4) and PLPC. PRCT(1) allowed the activity to distinguish between RDC-P and PDC inventory categories.

Activity Blocks 7 to 11 basically followed the same procedure as Blocks 2 to 6. Block 7 accumulated the order block's sales information at the full-line DC. This activity used data base variables IDCNO, IBLOCK, ORDGRP(1), ORDGRP(2), ORDGRP(3), IDCTEMP, DCIS(16) through DCIS(20) and RDCPC, the full-line DC percent of the order block's sales information. Block 7 was also a FORTRAN activity.

Block 8 accumulated the number of orders for the order block at the DC level. It was programmed in FORTRAN and used variables IDCNO, IDCTEMP and DCIS(21).

Block 9 accumulated the order block's shipment weight in the correct customer weight category as was done for the RDC-P in Block 4. The FORTRAN activity used variables IDCNO, IDCTEMP, DCWB, RDCPC, and WTACUM.

Activity Blocks 10 and 11 again adjusted the full-line DC's inventories-on-hand, PRDC(4). These inventories were reduced by the sales, ORDGRP(4), in the order block. The other data variables involved in this FORTRAN activity were IDCNO, IDCTEMP, PRCT(1), PRCT(2), PRCT (3), ITNPC, IBLOCK.

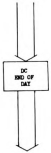
Block 12 signifies a return to the D&E normal sales processing routine that was linked to this OPS routine.

After a DC's simulated sales quota and the individual customer types' quotas had been reached, then certain activities were assumed to be performed at the end of day. These activities primarily centered around the review of the inventory levels and the shipment of product replenishments from the MCC's to the DC being processed.

Figure 29 shows the block diagram of activities associated with the DC end-of-day processing. Figure 28 lists the data base variables that were manipulated and used within this major subprogram. These variables, as was done with the prior OPS subsystem routines are only listed in the text while a brief description of the variables can be gained by referring to Figure 28. A more detailed description can be gained by referring to Appendix 1.

Activity Blocks 1 to 12 in Figure 29 were designed to accumulate certain inventory information plus also determine if any product reorders were to be dispatched from the DC. Block 1 allowed this routine to process only the inventory categories for which this DC was responsible. This was especially critical for RDC-P's. Block 1 was programmed in FORTRAN and used data base variable PRCT(1).

Activity Block 2 performed the updating of certain inventory status variables based on two conditions.



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
NMCCS	Total no. of MCC's
DCMCC(3)	Present MCC prod reorders received
DCMCC(4)	No. of mult prod reorders at MCC
DCMCC(5)	Wt on order at MCC + SDP indicator
DCP1(1)	DC-MCC linked tracked products
DCP1(2)	DC-MCC weight proportion
DCP1(3)	DC-MCC T1 + T4 factors
DCP(8)	DC type + PDC assignment
PARM	Serv time variance functions
ITNPC	Total no. of inventory categories
PRCT(1)	Partial-line inv category indicator
PRCT(2)	Beginning model product code
PRCT(3)	Ending model product code
PRCT(5)	Type of inventory policy code
PRCT(6)	Review period length for inv policy
PRDC(1)	ROP1 + ROP2
PRDC(2)	Replenishment S-level
PRDC(4)	Inventory on hand
PRDC(6)	ROQ + outstanding ROQ indicator
REG(15)	Pound ratio of total product sales
MCC(1)	MCC T2 serv time var pointer
MCC(2)	MCC scheduled SDP policy

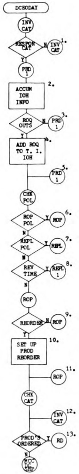
KEY ENDOGENOUS VARIABLES

Local	
T4	Cust OBT time

OUTPUTS	
DCMCC(1)	Total no. of mult-prod reorders
DCMCC(2)	Accum total reorder lead times
DCMCC(3)	Present MCC prod reorders received
DCMCC(4)	No. of mult-prod reorders rec'd at MCC
DCMCC(5)	Wt. on order at MCC + SDP indicator
PRDC(3)	Time-integrated IOH
PRDC(5)	Present stockout days per product
PRDC(6)	ROQ + outstanding ROQ indicator
PRCTDC(2)	QTD single-prod reorders
PRCTDC(3)	Time-integrated stocked-out cases
PRCTDC(4)	Total tracked product case sales
DCMCC1	DC-MCC weight accumulators
TPDEM(1)	QTD domestic, tracked prod case sales

DCSHPAP event attributes:	MCORAR event attributes:
1. Time of arrival	1. Time of arrival
2. Event code	2. Event code
3. Products on order	3. In-solution DC code
4. In-solution DC code	4. Products-on-order
	5. MCC code
	6. Total shipment wt

Figure 28.--DC End-of-Day Processing Variables



End-of-day activities for processing an in-solution DC's inventory levels

Process all inventory categories

1. Process only the inventory categories for which this DC is responsible

Update inventory level status variables

2. Adjust time-integrated stockout and inventory-on-hand variables plus stockout delay if necessary

3. Check to see if a reorder quantity is outstanding for this tracked product

4. If so, add ROQ to time-integrated IOH for total inventory approximation

5. Loop through all tracked products in category

Determine inventory control policy for category

6. Is the category's a reorder point policy?

7. Is the category's a replenishment policy?

8. Hybrid system used. Is it time for inventory review?

Reorder point policy processing

9. Product need reordering--IOH < ROP1

10. Set up single product reorder within total DC-MCC reorders if ROQ not outstanding

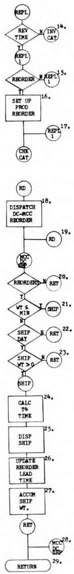
11. Loop through all products in category

Common routine for all inventory categories

12. Loop through all inventory categories

13. Any products reordered--if so, go set up DC-MCC reorder dispatches

Process MCC-DC shipment dispatch links



Replenishment policy processing

14. If category uses replenishment policy, check to see if it is time for product review

Review time for category

15. Product need reordering--IOH < ROP2

16. Set up single product reorder within total DC-MCC reorders if ROQ not outstanding

17. Loop through all products in category

Check to see if all categories processed

Reorder dispatch routine

18. Dispatch DC-MCC reorder using extrapolated weight, weight split proportion, and $t_1 + t_2$ lead times

19. Loop through all MCC's for this DC

End-of-day MCC shipment dispatch routine for DC being processed

20. Reorders outstanding for this MCC-DC link

21. Determine if a minimum shipment dispatch can be made

22. If minimum cannot be made, is this the scheduled shipment dispatch day?

23. Is accumulated shipment weight greater than zero for this link?

Make MCC-DC shipment dispatch

24. Calculate transit time between MCC & DC

25. Dispatch shipment to DC

26. Update reorder lead time accumulator for this MCC-DC link

27. Accumulate shipment weight in category for link

Connector for no action on link

28. Loop through all MCC's for DC being processed

29. Return to D&E normal daily sales processing routine

Figure 29.--DC End-of-Day Activities

Condition 1 was the normal updating of a time-integrated, inventory-on-hand variable, PRDC(3), which at the end of the quarter of simulated time could be divided by the number of workdays in a simulated quarter of the year to get the DC product's average inventory level. Condition 2 was one in which the inventory for a product had stocked out. Two variables were updated for this condition. A time-integrated, stocked-out cases variable, PRCTDC(3), was updated with the number of cases that were presently stocked out. This variable was used by the MEAS subsystem to determine a customer service penalty time for inventory stockouts that was added to the normal order cycle time to obtain an average total order cycle time. The other variable, PRDC(5), was updated every day in which a product was stocked out in order to be able to calculate the average and standard deviation of the product stockout delays given a stockout had occurred. This routine was programmed in FORTRAN and also used data base variables IDCNO and PRDC(4).

Blocks 3 and 4 were used to make an adjustment to the time-integrated inventory-on-hand, PRDC(3), for any outstanding DC product reorders. If there were any outstanding DC product reorders, indicated in variable PRDC(6), then the reorder quantity (ROQ), in variable PRDC(6), was added to PRDC(3) in order to be able to approximate the total average inventory committed to this DC for this product. If there were no outstanding ROQ's, then the next tracked product was processed. These FORTRAN activities also used data base variable IDCNO.

Block 5 was used to process all the tracked products in an inventory category. Processing the tracked products by category was necessary because certain data base, inventory variables were only accumulated and processed on a categorical basis rather than a product basis. This block used data base variables PRCT(2) and PRCT(3).

After the inventory status information had been accumulated for the DC, the routine then made a check of all the products in a category for any product reorders. In designing the mathematical model, three basic inventory management policies had been designed. Block 6 checked for the first type of policy which was a daily reorder point system. Data base variable PRCT(5) contained the inventory category's policy indicator. It had been assumed that all the products in a category used the same type of policy since there was the possibility of sample products being used to develop inventory category information.

Block 7 checked to see if the inventory category was an optional replenishment type policy based on a periodic review. Block 8 checked for the time for reviewing inventory levels given that the inventory policy was a hybrid or a combination daily reorder point and optional replenishment inventory policy. These were the basic policies built into the model to allow the flexibility of using various types of policies. Blocks 5 to 8 were programmed in FORTRAN and, besides using data base variable PRCT(5), PRCT(6) was used.

Given that the products in a category used a daily reorder point or hybrid system, Block 9 checked the product for a possible reorder. The product's inventory-on-hand (IOH), in variable PRDC(4), was checked against its reorder point, ROP1 of PRDC(1). If no reorder was necessary, then the next tracked product for the category was processed. If a product reorder was necessary, then activity Block 10 set up the reorder. Block 9 used FORTRAN instructions plus data base variable IDCNO.

Activity Block 10 used a FORTRAN subprogram to set up the product reorder. This subprogram set data base variable PRDC(6) by subtracting the IOH, in variable PRDC(4), from PRDC(2) which was set by an inventory management routine in M&C. PRCTDC(2) was adjusted for another single-product reorder plus PRCTDC(4) and TPDEM(1) were adjusted for the reordered quantity to approximate the case sales for this product. In addition, MCORAR event attributes 4 and 6 were set for the possible multiple-product reorder. This last activity used data base variables PRDC(6) plus WTCU. Also data base variable IDCNO was used in all of the activities in Block 10.

Block 11 was used to process all the tracked products in an inventory category. It was a FORTRAN activity that used data base variables PRCT(2) and PRCT(3). Block 12 was used to process all inventory categories ITNPC.

After all inventory categories had been processed, Block 13 checked the MCORAR attribute 4 to see if any single-product reorders had been necessary. If so, then a reorder had to be dispatched via Block 18 and 19. If not, then the routine to test for MCC-DC shipment dispatches was processed in Blocks 20 to 28.

Activity Blocks 14 to 17 were used to set up any product reorders using the basic, optional replenishment system. Block 14 checked to see if it was time to review the inventory levels. If it was not time, then the next inventory category was processed. PRCT(6) was used to check this time of inventory review.

If it was time to review inventory levels, then activity Block 15 checked the IOH in variable PRDC(4) against ROP2 in variable PRDC(1). EXTENDED FORTRAN instructions were necessary to shift out the ROP2 from PRDC(1) for checking purposes. If a reorder was not necessary, then the next tracked product was checked for this category. If a product reorder was necessary, then Block 16 followed essentially the same procedure as described for Block 10. All the same data base variables were adjusted as in Block 10. Block 17 was the basis for processing all tracked products in an inventory category. Block 17 used data base variables PRCT(2) and PRCT(3).

Given that single-product reorders had been developed in the prior activities, multiple-product reorders were dispatched to the supplying MCC's for the DC being processed (IDCNO). In order to accomplish this multiple-product reorder dispatch, a FORTRAN subprogram was developed that followed these basic steps:

1) Add the tracked products' weight to REG(4) and then extrapolate the tracked product weight for the firm's total product line--used data base variable REG(15) for extrapolating MCORAR attribute 6;

2) Assign the products to their supplying MCC's using data base variables DCP1(1) and IDCNO1 plus MCORAR attribute 4;

3) Split the total weight of the reorder between the MCC's based on the MCC-DC weight split proportions in variable DCP1(2);

4) Generate the reorder lead time elements T1, the reorder lead time of transmission, using data base variables DCP1(3), PARM and IDCNO1, plus the MCC order processing and preparation time, T2, using data base variables MCC(1), and PARM--do this for each of the DC's supplying MCC's;

5) Dispatch a multiple-product reorder event to each of the supplying MCC's that has a weight proportion greater than zero plus

adjust DCMCC(2) for the generated reorder lead time elements.

In generating the reorder lead time elements T1 and T2, the same basic routine that was used in generating customer service time elements T1 and T4 was used. The only differences were that the Monte Carlo procedure returned, for instance, a 4, 6 or 8 instead of a 0, 1 or 2 plus this variable element was not added to any type of constant service time.

Block 19 completed the reorder dispatch subprogram by indicating that a multiple-product reorder was dispatched for each supplying MCC--used data base variable NMCCS. The overall routine was programmed in FORTRAN, but it used some of the special features of EXTENDED FORTRAN available on the CDC 6500.

Blocks 20 to 28 accomplished the investigation of the outstanding MCC reorders for the DC being processed (IDCNO) to see if any shipments should be dispatched to the DC. The procedure was to determine if there were any reorders at the MCC that could be dispatched. Block 20 checked this using data base variable DCMCC(4). If there were no outstanding reorders, then the processing went on to the next supplying MCC for this DC--indicated by Block 28 which used variable NMCCS.

If reorders were outstanding, a check was made to see if there was enough weight accumulated, DCMCC(5), greater than the minimum shipment weight for either a truckload or carload, MCCWB, to dispatch a shipment. This checking was done in Block 21.

If the shipment weight was not large enough to initiate a minimum shipment dispatch, then the day of simulated time was tested to see if a scheduled shipment dispatch should be made. A shipment was dispatched if none had been made over the last "shipment dispatch" review period. The scheduled shipment was necessary to ensure that at least one shipment was made within a minimum number of days. This FORTRAN routine used data base variable MCC(2).

If this was the scheduled shipment dispatch day, Block 23 made sure there was weight to be shipped. This block used data base variable DCMCC(5).

Activity Blocks 24 and 25 initiated the MCC-DC shipment dispatch. Block 24 calculated the transit time for the MCC-DC link. This generation used the same basic procedure used for generating reorder lead time element T1 plus data base variables DCP1(3), PARM, and T4.

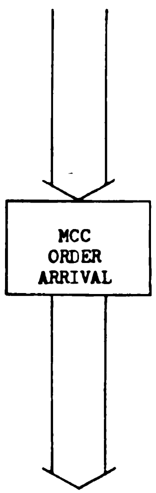
Activity Block 25 was the FORTRAN routine that dispatched the shipment from the MCC to the DC. In developing the MCC shipment dispatch, DCSHPAR event attributes 1 through 4 were set using the calculated T4, from the last Activity Block, plus IDCNO and DCMCC(3). In addition, the total number of multiple-product reorders, DCMCC(1), was adjusted with DCMCC(4).

Activity Block 26 updated the reorder lead time accumulator, DCMCC(2), using data base variables IDCNO, the calculated T4, and DCMCC(4). For each multiple-product reorder that was outstanding, T4 was added to DCMCC(2). DCMCC(4) was then reset to zero in this FORTRAN activity.

Activity Block 27 accumulated the dispatched shipment weight in the appropriate weight category of the alternative weight categories, DCMCC1. This basic FORTRAN activity used data base variables IDCNO, IDCNO1, DCMCC(5), the dispatched weight, DCP(8), the DC type which enabled the routine to determine which of the weight categories, MCCWB, was used for a certain type of DC.

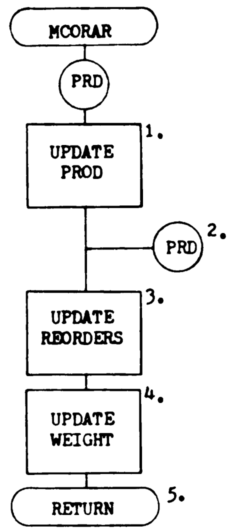
Block 28 shows that all the supplying MCC's were checked to see if any shipment dispatches were to be made. After all the MCC's had been checked, Block 29 indicates a return to the D&E normal sales processing routine that called this OPS subprogram after all customer sales had been processed for the simulated day of operation.

The next major component in the OPS subsystem is the processing of a MCC order arrival. This is a variable-time event activity that used the data base variables listed in Figure 30. The logical flow of activities are presented in Figure 31. Activity Blocks 1 and 2 of Figure 31 show the EXTENDED FORTRAN activity of setting



NAME	DESCRIPTION
INPUTS	
MCORAR event attributes:	
1. Time of arrival	
2. Event code	
3. In-solution DC code	
4. Products on reorder	
5. MCC code	
6. Total shipment wt	
ITNSP	Total no. of tracked products
OUTPUTS	
DCMCC(3)	Present MCC received products on reorder
DCMCC(4)	No. of mult-prod reorders rec'd at MCC
DCMCC(5)	Wt on order at MCC + SDP indicator

Figure 30.--MCC Order Arrival Processing Variables



"Variable-time" routine to process a DC to MCC order arrival

Process product on reorder

Add product to next shipment for this link

Loop through all products in reorder

Add one more outstanding multiple-product reorder for this link

Update the total shipment weight on reorder

Return to GASP "EXECUTIVE"

Figure 31.--MCC Order Arrival Processing

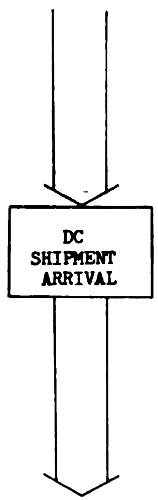
the indicators showing the products that were added to the total list of products that were awaiting dispatch to the DC. These Blocks used data base variables DCMCC(3) and ITNSP plus MCORAR event attributes 3, 4 and 5.

Activity Block 3 updated the total number of multiple-product reorders that had been received for the DC-MCC link being processed. This basic FORTRAN activity used data base variables DCMCC(4) plus MCORAR event attributes 3 and 5.

Activity Block 4 updated the total weight to be dispatched on the next MCC-DC shipment. This FORTRAN activity used data base variable DCMCC(5) plus MCORAR event attributes 3, 5 and 6. After updating the total weight to be shipped, Block 5 indicates a return to the GASP "EXECUTIVE" routine to process the next event in the queue.

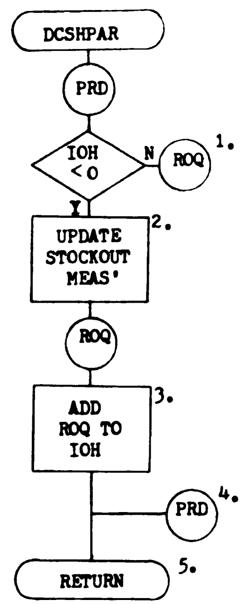
The last major component of the OPS subsystem is again a variable-time event activity used to process the arrival of a shipment to replenish product inventories at a DC. This activity, called DCSHPAR, uses the variables listed in Figure 32 to update the inventory status variables. Figure 33 shows the logical activities to accomplish this updating of the status variables.

Activity Block 1 checked the products on the shipment, specified in DCSHPAR event attribute 3, to see if their present, DC inventories-on-hand were stocked out.



NAME	DESCRIPTION
INPUTS	
DCSHPAR event attributes:	
1. Time of arrival	
2. Event code	
3. Products on reorder	
4. In-solution DC code	
ITNSP	Total no. of tracked products
PRDC(4)	Inventory on hand
PRDC(6)	ROQ + outstanding ROQ indicator
OUTPUTS	
PRDC(4)	Inventory on hand
PRDC(5)	Present stockout days per product
PRDC(6)	ROQ + outstanding ROQ indicator
PRCTDC(1)	QTD no. of prod stockouts
DCIS(13)	Percent case units backordered
DCIS(14)	Avg stockout delay accumulators
DCIS(15)	Std dev stockout delay accumulators

Figure 32.--DC Shipment Arrival Processing Variables



"Variable-time" routine to process a MCC to DC shipment arrival

Process all products received

See if inventory-on-hand is negative

If so, update inventory stockout measures

Process reorder quantity

Add product's reorder quantity to inventory-on-hand, reset ROQ indicator

Loop through all products in shipment arrival

Return to GASP "EXECUTIVE"

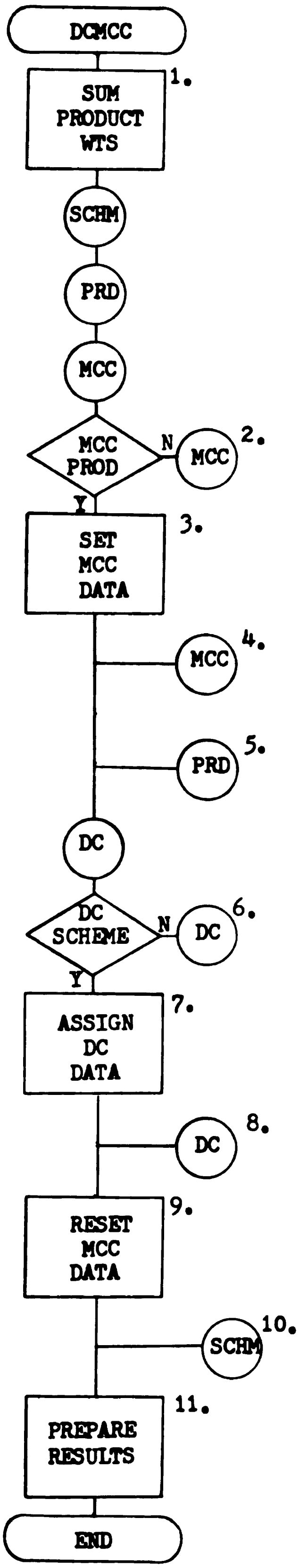
Figure 33.--DC Shipment Arrival Processing

If a product was stocked out, then certain stockout measures, PRCTDC(1), DCIS(13), DCIS(14), DCIS(15), were updated in Block 2. These FORTRAN activities also used data base variables PRDC(4) and PRDC(5) plus DCSHPAR event attribute 4.

Activity Blocks 3 and 4 updated the inventories-on-hand for all products in the received shipment. The indicator of an outstanding product reorder was also set off. These activities used data base variables PRDC(4) and PRDC(6) plus DCSHPAR event attributes 3 and 4. After updating all product inventories, the routine returned to the GASP "EXECUTIVE" to process the next imminent event.

Supporting Data System

The only computer supporting activity for the OPS subsystem was used to establish some basic DC-MCC product link information and punch the information in a form that could be read exogenously into the operating system. Figure 34 outlines the activities that were used in this EXTENDED FORTRAN activity that specifically developed the exogenous data base variables DCP1(1) and DCP1(2). DCP1(1) contained the packed computer words that indicated the products that were to be supplied over the alternative MCC-DC links. DCP1(2) contained the proportions of the total replenishment weight to be shipped to the DC from each of its supplying MCC's. These proportions were



Set up DC-MCC product link and replenishment weight proportions

Sum total weight for all tracked products

Process each MCC ranking scheme

Check each tracked product

Process each in-solution MCC

Check to see if this ranked MCC has product

Set product indicator and add weight proportion for MCC selected

Loop through all MCC's

Loop through all tracked products

Check each potential DC to see if it uses present scheme

See if DC uses present ranking scheme

Assign data to DC

Loop through all potential DC's

Reset product indicators and MCC weight proportion accumulators

Loop through all MCC ranking schemes

Prepare DC-MCC output results

End of analysis

Figure 34.--DC-MCC Link Information Generation

extremely critical when sampled, tracked products were used instead of the total product line for inventory control purposes.

The input information into the program was:

1) A control card that contained the number of potential DC locations, the number of supplying MCC's, the number of tracked products, and the number of priority ranking schemes for the potential DC's;
2) One card each for the alternative ranking schemes that indicated the priority with which DC's desired to be supplied from the alternative MCC's;
3) Tracked product information, with one card per product, that contained product identification information, total domestic weight sales in a prior year, plus an indicator for each MCC as to whether the MCC produced the product or not;
4) One card for each potential DC that indicated the ranking scheme it was to use.

The details of the program are explained in Figure 34. Block 1 developed the weight proportions for all tracked products. Blocks 2 to 4 accumulated the weight proportions and set the product indicators for each of the MCC's of highest rank in the present priority scheme and for all tracked products.

After all tracked products were processed, the potential DC's were checked to see if they used the present priority ranking scheme. If they used it, then the packed, product indicator words plus the weight proportions were assigned to those DC's. Blocks 9 and 10 indicate the resetting of the product indicator words and weight proportion accumulators plus the processing of all priority ranking schemes. Finally in Block 11 the results were punched for all potential DC's.

CHAPTER V

THE MEASUREMENT SUBSYSTEM

Inputs, Outputs and Data Base Requirements

The MEASUREMENT (MEAS) subsystem had the task of developing the criteria for evaluating alternative distribution plans that had been simulated. The criteria of evaluation were developed over the planning horizon and focused on measures of costs, sales and service. The DC cost measures were based on the five primary physical distribution activities of transportation, materials handling or unitization, communications, facility commitment and inventory control. No costs were accumulated at the DU or MCC stages. The measures of service focused on customer service plus the service aspects concerning the shipment of inventories between the DC and MCC stages of the physical distribution system. Sales measures centered on the amount of sales that had been accumulated for each DU, in-solution DC and MCC-DC link.

In line with the above statements, the OPERATING SYSTEM of the MEAS subsystem incorporated eight major computer subprograms:

1) SERV--the routine to calculate the measures of customer and DC-MCC reorder service;
2) EXTRAP--the routine which extrapolated the tracked products' average inventory characteristics for the firm's total product line;
3) OBTCST--the routine to calculate the outbound transportation cost from the in-solution DC's to their assigned DU's;
4) IBTCST--the routine to calculate the inbound transportation cost from the supplying MCC's to each of the in-solution DC's;
5) TPCST--the routine to calculate the cost of preparing customer orders for DC shipment dispatch;
6) COMCST--the routine to calculate the cost of DU to DC order transmission activities, for which the DC was responsible, plus the activities of processing the order up until the time that the order was physically picked and prepared for shipment dispatch;
7) FACST--the allocated, investment cost for land, building and equipment at the in-solution DC site;
8) INVCST--the cost of holding or carrying inventories at the DC stage, the carrying cost for pipeline inventories for the MCC and DC links plus, possibly, the cost of

handling and preparing the MCC dispatched shipments to each in-solution DC.

These measures of cost and service were calculated quarterly using the base data developed in the OPS subsystem. The measures were inputted to the MONITOR and CONTROL subsystem which used part of the variables for dynamic feedback responses and which outputted all the measures to the REPORT GENERATOR system. The measures of sales were left intact, as received from the OPS subsystem, and merely passed them along with the measures of cost and service to M&C.

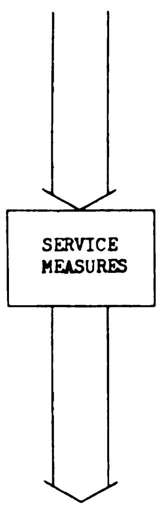
In LREPS the assumption had been made to output model information on a quarterly basis. Therefore, the MEAS activities were only utilized at the end of every quarter of simulated time. The rationale for the quarterly outputting of information was similar to the rationale for developing end-of-period accounting reports that not only show historical information, such as a profit and loss statement, but also snapshot, status information such as the information shown in a balance sheet.

All of the activities in the MEAS OPERATING SYSTEM used basic FORTRAN instructions. Therefore, no particular mention will be made of the utilized programming language(s) in the detail description of each of the above routines. However, a system flowchart of the variables used in the routine, the logical block diagram plus any programming problems or peculiar programming techniques will still be included in the detailed routine discussion.

The SUPPORTING DATA SYSTEM for the MEAS subsystem had three computer activities. The first routine generated a matrix of calculated highway distances from each of the potential DC locations to a sample set of destination points for each DC. This matrix of highway distances from the potential DC's served as part of the inputs into the second supporting, computer activity of generating regression equations. The regression equation coefficients were the basis for determining the outbound transportation rates from the in-solution DC's to each of their respectively assigned DU's. The other inputs into this second computer activity were actual, weighted average freight rates. These average rates were based upon the percentages of total weight that were shipped in the selected order weight classes. The third supporting computer activity generated these percentages.

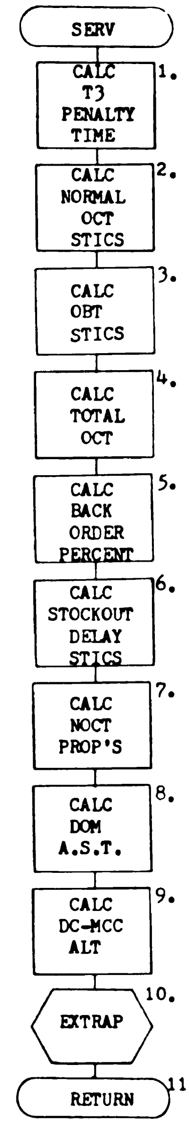
Operating System

The first major component of the MEAS subsystem is the routine to calculate the measures of service (SERV). Figure 35 lists the variables associated with this routine, while Figure 36 identifies the sequence of activities with which these service measures were developed for each in-solution DC.



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
ITNPC	Total no. of inventory categories
PRCTDC(4)	Total cases sold for tracked products
DCIS(21)	QTD order sales
PRCTDC(1)	QTD stockouts
NMCCS	No. of MCC's in-solution
DCMCC(1)	Total no. of mult-prod reorders
DAYCON	Cust TOCT decision parameter
DCIS(16)	QTD dollar sales
OUTPUTS	
PRCTDC(3)	Inv cat T3 penalty time
DCIS(1)	DC cust T3 penalty time
DCIS(9)	Avg normal OCT
DCIS(10)	Std dev normal OCT
DCIS(11)	Avg OBT
DCIS(12)	Std dev OBT
DCIS(7)	Avg total OCT
DCIS(13)	Percent cases backordered
DCIS(14)	Avg stockout delay
DCIS(15)	Std dev stockout delay
DCMCC(2)	DC-MCC avg reorder lead time
OCTDIS(1)	NOCT sales dollar proportions
OCTDIS(2)	NOCT order proportions
REG(13)	NOCT sales dollars
DOMAST	Domestic avg total OCT

Figure 35.--Calculation of Service Measures' Variables



Calculate customer and DC-MCC service measures

1. Calculate customer service time element T3
2. Calculate mean and standard deviation of customer normal order cycle time
3. Calculate mean and standard deviation of customer outbound transportation time
4. Calculate average total order cycle time
5. Calculate percentage of case units backordered
6. Calculate mean and standard deviation of product stockout delays
7. Calculate normal order cycle time proportions
8. Calculate domestic average service time
9. Calculate average lead time for each DC-MCC link
10. Link to routine to extrapolate DC inventory characteristics
11. Return to M&C end-of-quarter routine

Figure 36.--Calculate Service Measures

Activity Block 1 calculated a customer service penalty time, T3, that resulted when a DC stocked out of tracked products during the past quarter's activities. T3 was added to the averages of the other customer service time elements, T1, T2 and T4, to arrive at a total average order cycle time (TOCT).

According to the math model specifications, T3 was calculated for each inventory category and then a weighted average, based on total, tracked product case sales, was developed for the DC. Data base variables DCIS(1), PRCTDC(3), PRCTDC(4), and ITNPC were used in this activity.

Activity Block 2 calculated the mean and standard deviation of the customer normal order cycle time (NOCT). The NOCT had been defined as the sum of the averages of customer service time elements T1, T2 and T4. This activity used data base variables DCIS(21) plus DCIS(9) for the mean and DCIS(10) for the standard deviation of the NOCT.

The mean and standard deviation of customer outbound transit time, T4, was calculated in Activity Block 3. DCIS(11) and DCIS(12) were used for the mean and standard deviation of T4 while DCIS(21) contained the QTD order sales. Both Activity Blocks 2 and 3 used a computer library function, SQRT, for computing the standard deviation.

Activity Block 4 combined the mean NOCT, in data base variable DCIS(9), with the penalty time, T3, to develop the average total order cycle time, DCIS(7).

Block 5 developed another measure of inventory stockouts. This measure was the percentage of tracked product case sales, PRCTDC(4), that were backordered for all inventory categories, ITNPC. The measure was stored in data base variable DCIS(13).

Activity Block 6 calculated the mean and standard deviation of the delay with which the tracked products were delivered to customers because of product stockouts. This activity used data base variables DCIS(14) and DCIS(15) for the mean and standard deviation of the delay plus the total number of product stockouts, PRCTDC(1), for all inventory categories, ITNPC. This routine also used the SQRT function.

Activity Block 7 calculated the proportions of the DC sales dollars that were delivered to its customers within a NOCT of three, five, seven, nine and eleven days. Similar proportions were calculated for DC sales orders. The data base variables used in these activities were OCTDIS(1) for the DC sales dollar proportions, OCTDIS(2) for the DC customer order proportions, DCIS(16), DCIS(21) and IREGNO.

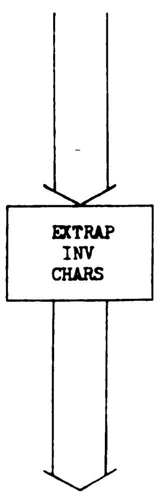
Activity Block 8 accumulated the DC's increment to a measure of the domestic total average order cycle time, DOMAST. The average was a weighted average based on customer orders, DCIS(21). The DC's total average OCT was retrieved from DCIS(7).

Activity Block 9 calculated the total average reorder lead times for each DC-MCC link associated with the in-solution DC being processed. The reorder lead time was composed of the DC to MCC reorder transmission time, T1, the reorder processing and preparation time, T2, the time that an order waited to be shipped from the MCC, T3, and the MCC to DC shipment transit time, T4. The data base variables used in this activity were DCMCC(1), DCMCC(2) and NMCCS.

Activity Block 12 was a link to the MEAS routine, EXTRAP, that developed additional measures of inventory characteristics. Block 12 signifies a return to the M&C end-of-quarter (EOQ) routine that called this subprogram.

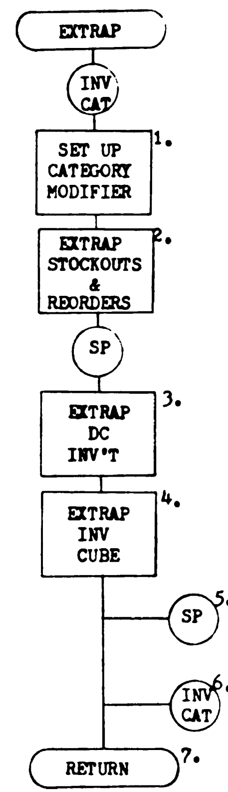
The next major component of the MEAS subsystem is the extrapolation of average inventory characteristics resulting from the activities associated with the processing of tracked products. This routine generalized or extrapolated the average characteristics, which it calculated, to the total product line. If the total product line was used, then there was no extrapolation necessary. The only necessary thing was the calculation of average characteristics. The utilized data base variables are shown in Figure 37.

Activity Block 1 of Figure 38 calculated the modifier which was used to extrapolate the inventory characteristics for a specific inventory category. This



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
PRCT(2)	Beginning model product code
PRCT(3)	Ending model product code
PRCT(4)	Total no. of category products
PRDC(3)	Time-integrated IOH
CGSCS	Cost of goods sold per case
KEY ENDOGENOUS VARIABLES	
Local	
CM	Category modifier
OUTPUTS	
DCCOST(6)	DC extrapolated average investment
PRCTDC(1)	QTD stockouts
PRCTDC(2)	QTD single-product reorders
PRDC(3)	Time-integrated IOH

Figure 37.--Inventory Characteristics Extrapolation Variables



Extrapolate DC inventory characteristics

Process all inventory categories

Calculate the modifier to extrapolate this inventory category's characteristics

Extrapolate stockouts and single-product reorders

Process all tracked products in category

Extrapolate product's average DC investment

Extrapolate product's average cubic inventory

Loop through tracked products

Loop through inventory categories

Return to MEAS service measures' routine

Figure 38.--Inventory Extrapolation Routine

category modifier, CM, was calculated by dividing the total number of products, PRCT(4), that an inventory category was representing by the number of sample tracked products, PRCT(3) minus PRCT(2) plus one, in the category. The use of the modifier accomplished the same thing as if, for instance, the number of stockouts for the tracked products was divided by the number of tracked products to get the average stockouts per product. This average would then be multiplied by the total number of products.

Activity Block 2 extrapolated the number of stockouts, PRCTDC(1), and single-product reorders, PRCTDC(2), for a DC's inventory category. The basis of extrapolation was the multiplication of the tracked product stockouts and reorders by the CM.

Activity Block 3 extrapolated the tracked products' average investment. This was calculated by multiplying each of the tracked product's time-integrated inventory in case units, PRDC(3), by the cost of goods sold per case unit, CGSCS, and CM. The results for all tracked products were stored in DCCOST(6). DCCOST(6) then contained the extrapolated, time-integrated average investment in inventories based on the products' cost of goods sold. This figure, DCCOST(6), will be later multiplied by the daily inventory carrying charge, DICC, to get the cost of holding inventory at a DC.

Activity Block 4 followed basically the same procedure as in Block 3 except that the average case units, in data base variable PRDC(3), was converted to an extrapolated, average cubic IOH using data base variables, CM and CUBCS.

Activity Blocks 5 and 6 identify the activities of processing all tracked products in an inventory category, PRCT(3) minus PRCT(2) plus one, for all inventory categories, ITNPC. Block 7 indicates a return to the MEAS service (SERV) routine that called this routine.

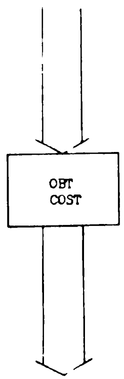
After processing the measures of service and doing the necessary extrapolation for both service and costing purposes, the next components are the MEAS cost routines.

The first cost routine is graphically presented in Figure 40. The data base variables are presented in Figure 39. The purpose of this routine is to calculate the cost of outbound transportation from the in-solution DC, IDCNO, to its assigned DU's.

Activity Block 1 initialized data base variable, DCCOST(1), to zero. This variable will be the accumulator of the total DC outbound transportation cost for all DU's.

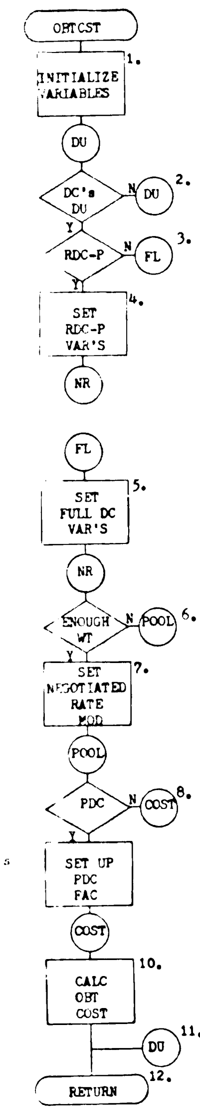
Activity Block 2 used data base variables, DU(8) and IDUNO1, to see if the DU was assigned to the DC being processed. If not, then the next DU was checked for processing.

Activity Block 3 checked to see if the DC was a RDC-P. If it was, then a rate correction factor was set equal to something greater than one in Activity Block 4. The rate correction factor was used to modify the outbound freight rates for the higher costs of shipping



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
DU(8)	DU's DC assignment
IDUNO1	Demand unit selected for processing
DCP(8)	DC type + PDC assignment
DU(11)	Full-line DC highway distance
DU(12)	RDC-P highway distance
DU(15)	Full-line DC QTD lb. sales
DU(16)	RDC-P QTD lb. sales
DCIS(17)	QTD weight sales
IREGNO	No. of region being processed
OBTNRM(1)	Regional "a" negotiated rate modifier
OBTNRM(2)	Regional "b" negotiated rate modifier
DCP(3)	Constant "a" coefficient
DCP(4)	Variable "b" coefficient
DCP(5)	Real rate increase "a" modifier
DCP(6)	Real rate increase "b" modifier
IYEAR	Year of simulated time
NDUS	Total no. of DU's being processed
KEY ENDOGENOUS VARIABLES	
Local	
RCF	DC-type rate correction factor
DIST	DC-DU distance
WT	DC-DU wt
POOLFC	PDC pooled shipment rate mod factor
OUTPUTS	
DCCOST(1)	DC outbound transportation cost

Figure 39.—Outbound Transportation Cost Calculation Variables



Calculate DC outbound transportation cost

Initialize outbound transportation cost variables

Process all DU's for DC being processed

If DU is not in DC's area, get another DU

See if DC is a RDC-P

Set rate correction factor, highway distance and QTD weight sales for DU & RDC-P

Go to negotiated rate test

Full-line DC variable set routine

Set rate correction factor, highway distance, and QTD weight sales for DU & full-line DC

Negotiated transportation rate modification test routine

Enough weight processed by DC to warrant negotiated transportation rates

Set regional negotiated rate modification factors

Routine to see if DC is a PDC for pooled shipment rate modification

Is DC a PDC

Set up PDC pooled shipment rate modification factor

Outbound transportation cost routine for DC-DU link

Calculate outbound transportation cost

Loop through all DU's in DC area

Return to MAC end-of-quarter activity routine

Figure 40.—Outbound Transportation Cost Routine

small partial orders from the RDC-P. Activity Block 4 also retrieved the RDC-P to DU shipped weight and highway distance. These activities used data base variables, DCP(8), IDCNO1, IDUNO1, DU(12), DU(16), RDF, DIST and WT.

If the DC was a full-line DC, then Activity Block 5 set the rate correction factor, RCF, to one plus retrieved the weight, DU(15), and the highway distance, DU(11), for the DU being processed, IDUNO1. The retrieved weight and distance were stored in DIST and WT.

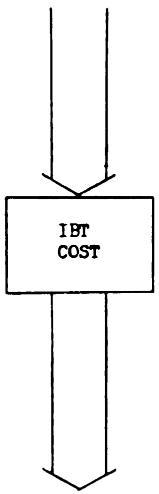
Activity Blocks 6 and 7 checked to see if the amount of weight processed over the last quarter might have warranted the use of negotiated freight rates. The assumption had been made that if the weight, DCIS(17), was large enough, then the DC would have negotiated for cheaper transportation rates for supplying its customers. The modifiers were developed by region. Therefore, the negotiated rate modifiers, OBTNRM(1) and OBTNRM(2), were picked up for the region being processed, IREGNO, if enough weight had been processed over the last quarter.

Activity Blocks 8 and 9 did similar activities as Blocks 6 and 7. If the DC was a PDC, then it was assumed that the amount of goods being shipped out of the PDC warranted pooled or consolidated shipments to the DU's. These pooled shipments, also referred to in the OPS subsystem in reference to the calculation of the customer service time element T2, were assumed by management to be dispatched in short enough time in order not to hurt customer service seriously. The data base variable, POOLFC, held the PDC rate modifier. This rate modifier was less than one. Activity Block 8 also used data base variables DCP(8) and IDCNO1 in determining the type of DC.

Activity Block 10 calculated the outbound transportation rate per pound using regression equations based on the highway distance, DIST, to the DU from the DC. The regression equations constant "a" coefficient, DCP(3), and variable "b" coefficient, DCP(4), based on distance developed the basic rate. This rate was then modified with rate modifiers OBTNRM(1), OBTNRM(2), RCF and POOLFC. In addition to the above modifications, the beginning base year rate, that was the direct output from the regression equations, was also modified by an annual increase in the rate based on the year of simulated time, IYEAR. These "real" rate increases were stored in data base variables DCP(5) and DCP(6) for the potential DC, IDCNO1, being processed. Finally the rate was multiplied by the total weight, WT, shipped from the DC to the DU and the result added to DCCOST(1).

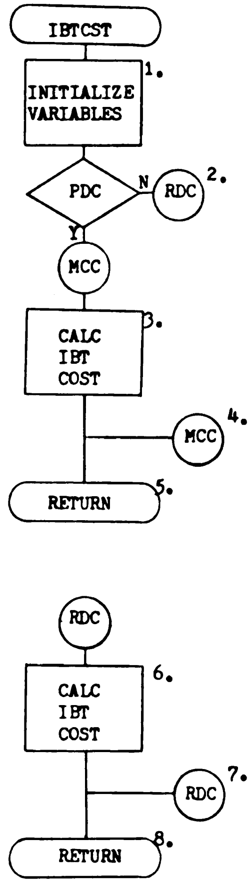
After all the DU's had been processed, indicated by data base variable NDUS, then the routine returned back to the calling M&C end-of-quarter (EOQ) routine.

Figure 41 shows data base variables associated with the routine for calculating the inbound transportation cost from the MCC's to the DC being processed.



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
DCP(8)	DC type + PDC assignment
DCP2	DC-MCC freight rates
DCMCC1	Accumulated DC-MCC wt
KEY ENDOGENOUS VARIABLES	
Local	
IVOLP	No. of PDC weight categories
IVOLR	No. of RDC weight categories
OUTPUTS	
DCCOST(2)	DC inbound transportation cost

Figure 41.--Inbound Transportation Cost Calculation Variables



Calculate MCC-DC inbound transportation cost

Initialize DC inbound transportation cost variable

Check to see if DC is a PDC

PDC--process all MCC links

Calculate inbound transportation costs for PDC weight categories

Loop through all MCC's linked to this PDC

Return to M&C end-of-quarter activity

RDC--process all MCC links

Calculate inbound transportation costs for RDC weight categories

Loop through all MCC's linked to this RDC

Return to M&C end-of-quarter activity

Figure 42.--Inbound Transportation Cost Routine

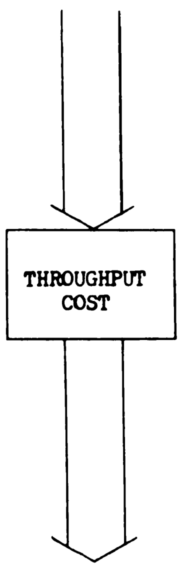
Activity Block 1 in Figure 42 initialized data base variable, DCCOST(2), which was used to accumulate the total cost for moving the weight for the MCC-DC links. After initializing this variable, Activity Block 2 checked to see if a PDC was being processed. This checking used data base variables DCP(8) and IDCNO1. If a PDC was being processed, then Activity Blocks 3, 4 and 5 were processed. If not, then Blocks 6, 7 and 8 were processed.

The reason for separate blocks for the PDC's and RDC's was that the PDC generally shipped in higher weight categories when they dispatched shipments. Therefore, Block 3 used higher weight categories for accumulating weight and calculating costs. Blocks 3 and 4 used data base variables DCP2, DCMCC1, IVOLP, IDCNO1 and NMCCS.

Activity Blocks 6 and 7 used data base variables DCP2, DCMCC1, IDCNO1, IVOLR and NMCCS. After processing all the MCC-DC links, this routine returned to the M&C EOQ calling routine.

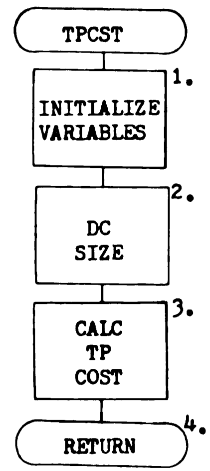
Figure 44 shows the major activities associated with the next major component of the MEAS subsystem. This component calculated the cost of preparing a customer order for shipment. Activity Block 1 cleared variable DCCOST(3), shown in Figure 43, for accumulating the DC throughput cost.

Block 2 determined the size of DC, using data base variables IDCNO1 and DCIS(8). Given the DC size, Activity



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
DCIS(8)	DC size indicator
DCIS(17)	QTD wt sales
THRUPC	Throughput cost by DC size
OUTPUTS	
DCCOST(3)	DC throughput costs

Figure 43.--Throughput Cost Calculation Variables



Calculate DC throughput costs

1. Initialize DC throughput cost variables

2. Determine DC size

3. Calculate DC throughput costs

4. Return to M&C end-of-quarter activity routine

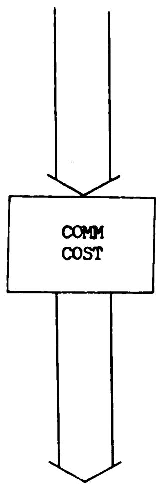
Figure 44.--Throughput Cost Calculation Routine

Block 3 calculated the total throughput cost, DCCOST(3), for the past quarter's activities using data base variables DCIS(17) and THRUPC. Block 4 was a return to M&C EOQ.

Figure 45 lists the variables associated with the cost of transmitting and processing an order up to the point when it was physically prepared for shipment. These communications costs, which are accumulated at the DC stage, are allocated to the DC on the basis of domestic, regional and DC fixed and variable cost factors. The reason for the three cost levels was to allow for the flexibility of processing the order on the basis of decentralized, regional and/or domestic systems.

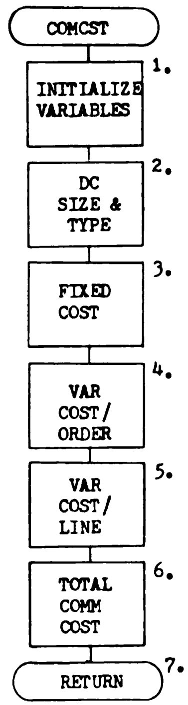
Activity Block 1 initialized the DC variable that accumulated the total communications cost, DCCOST(4). This block also set to zero three other data base variables. These three variables were FC, the total of all fixed cost elements, VCO, the total variable cost per order, and VCL, the total variable cost per line item, for all three geographical bases.

Activity Block 2 determined the DC size and type. This activity used data base variable IDCNO1, DCP(8) and DCIS(8). Activity Blocks 3 to 5 calculated the total FC, VCO and VCL. To develop these totals, data base variable IREGNO, COMMC, COMMCR, REG(17), COMMCD and NUMDCS were used.



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
DCP(8)	DC type + PDC assignment
COMMC	DC communication cost factor
COMMCR	Regional communication cost factor
COMMCD	Domestic communication cost factor
DCIS(20)	QTD line sales
DCIS(21)	QTD order sales
DCIS(8)	DC size indicator
IREGNO	No. of region being processed
NUMDCS	No. of in-solution DC's domestically
REG(17)	No. of in-solution DC's per region
KEY ENDOGENOUS VARIABLES	
Local	
FC	Fixed cost for all geo units
VCO	Var cost/order for all geo units
VCL	Var cost/line for all geo units
OUTPUTS	
DCCOST(4)	DC communication's cost

Figure 45.--Communications Cost Calculation Variables



Calculate DC communication's cost

Initialize DC communication's cost variables

Determine DC size and type

Calculate fixed communication's cost for DC, region and domestic geographic units

Calculate variable communication's cost per order for all geographic units

Calculate variable communication's cost per line for all geographic units

Calculate total DC communications cost

Return to M&C end-of-quarter activity routine

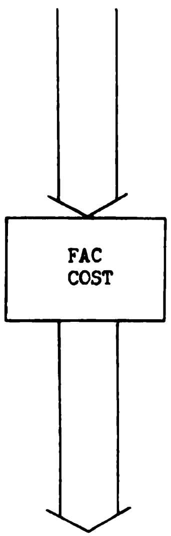
Figure 46.--Communications Cost Calculation Routine

Activity Block 6 calculated the total communication's cost, DCCOST(4), using data base variables FC, VCO, VCL, DCIS(21) and DCIS(20). Block 7 identifies a return back to the M&C EOQ routine.

The calculation of an allocated facility cost per quarter is graphically presented in Figure 48. This component's purpose was to prepare an allocated, capital investment for each quarter in land, building and equipment. Activity Block 1 cleared the DC variable, DCCOST(5), that was to contain the allocated investment. Block 2 identified the DC size and type using the data base variables DCP(8), IDCN01 and DCIS(8). The amount of investment was assumed to vary with the size and type of DC.

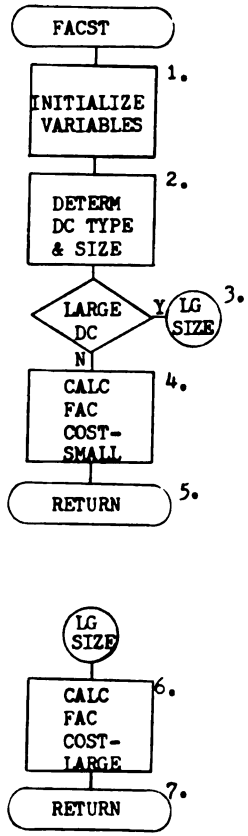
A further assumption was also made concerning the proportion of the total investment that was made in equipment and in the land and building between large and small DC's. This last assumption was the basis for Activity Blocks 3, 4 and 6. Block 3 distinguished between the size of DC's using data base variable DCIS(8). Given that a DC was classified as being in a small category, then the ratio of equipment to land and building investments was low. The rationale was that the smaller DC's were most likely to have less automated equipment.

Activity Block 4, therefore, calculated the allocated facility cost, DCCOST(5), using the small ratio and data base variables INVEST and CLF, a cost-of-living



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
DCP(8)	DC type + PDC assignment
DCIS(8)	DC size indicator
INVEST	DC avg investment dollars
CLF	DC cost of living factor
OUTPUTS	
DCCOST(5)	DC facility cost

Figure 47.--Facility Cost Calculation Variables



Calculate DC allocated facility cost

Initialize DC facility cost variables

Determine DC type and size

Check to see if DC is in large size category

Calculate facility cost using small DC equipment to building ratio

Return to M&C end-of-quarter activity routine

Large size DC routine

Calculate facility cost using large DC equipment to building ratio

Return to M&C end-of-quarter activity routine

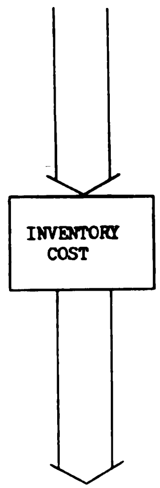
Figure 48.--Facility Cost Calculation Routine

factor for the potential DC location. Activity Block 6 used the same data base variables except the large ratio of equipment to land and building investments was used. The ratios were built directly into the routine. Blocks 5 and 7 identify a return to the M&C EOQ routine.

The calculation of the cost of carrying and handling product inventories is represented by Figure 50. The data base variables for this component are listed in Figure 49. Given the extrapolated, DC average investment, calculated in the MEAS EXTRAP routine, Activity Block 1 calculated the cost of carrying or holding inventories, DCCOST(6), using the daily inventory carrying charge, DICC.

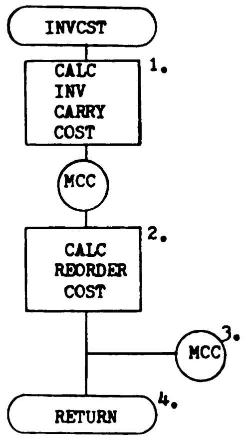
Activity Blocks 2 and 3 calculated the cost of preparing multiple-product reorders for shipment dispatch at each of the supplying MCC's for the DC being processed. These activities used data base variables DCMCC(1), MCC(3), NMCCS and DCCOST(6). DCCOST(6) was used to add these reorder costs to the already accumulated inventory carrying charges.

The cost of processing the reorder from the DC to the MCC was not included in this routine because it had already been included in the DC communication's cost. The rationale for this exclusion was that, with the increased use of computers, the placement of product reorders has almost become a by-product of processing customer orders. In addition, if the decision might be to exclude the <u>MCC</u> order preparation costs at the DC



NAME	DESCRIPTION
INPUTS	
IDCNO	In-solution DC being processed
DICC	Inventory carrying charge
NMCCS	No. of MCC's in-solution
DCMCC(1)	No. of mult-prod reorders
MCC(3)	MCC reorder cost
OUTPUT	
DCCOST(6)	DC inventory cost

Figure 49.--Inventory Cost Calculation Variables



Calculate cost of handling and carrying inventory at a DC level

Calculate the cost of carrying inventory

Process reorder cost for all DC-MCC links

Calculate reorder cost for multiple-product reorders

Loop through all DC-MCC links

Return to M&C end-of-quarter routine

Figure 50.--Inventory Cost Calculation Routine

stage of the model, then MCC(3) can be set equal to zero. Block 4 identifies a return back to the M&C EOQ routine.

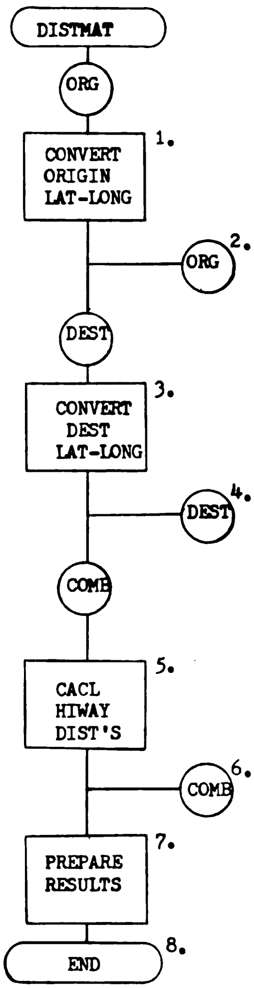
Supporting Data System

As mentioned in the introductory section of this chapter, there are only three supporting computer activities for the MEAS subsystem. Figure 51 identifies the logical activities associated with the FORTRAN activity that had an effect on data base variables DCP(3), the outbound transportation rate constant "a" coefficient, and DCP(4), the outbound transportation rate variable "b" coefficient.

The purpose of this program was to calculate a matrix of highway distances for the potential DC's to a sample of destination points for each potential DC. The program's inputs included the city name, latitude and longitude for each of the potential DC's and for their sample destination points. The routine to calculate the highway distances was the same one used in the D&E "Distance Check" routine that developed the conversion factor from spherical to road distances.

The output of the run was a listing of the calculated distances with the rows of the matrix being all sample destination points and the columns being the potential DC locations. All elements of the matrix were calculated and printed.

The output from this program was then keypunched along with a weighted average freight rate for each



Calculate matrix of highway distances

Routines to convert potential DC latitude, longitude

Convert potential DC latitude, longitude to rectangular X & Y coordinate basis

Loop through all potential DC's

Routines to convert sample destination latitude, longitude

Convert sample destination latitude, longitude to rectangular X & Y coordinate basis

Loop through all sample destination points

Routine to calculate highway distances

Calculate highway distance for a combined origin-destination

Loop through all combinations

Prepare listing of results

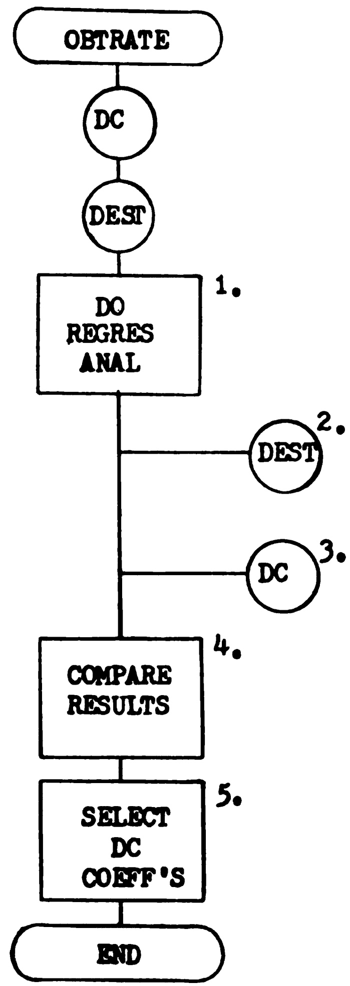
End of analysis

Figure 51.--Distance Matrix Routine

desired sample destination-potential DC link. This card input served as the basic input into a system library, univariate regression analysis program. The independent variable was distance and the dependent variable was the weighted average freight rate.

Straight line regression analysis was run on these inputs as shown in Figure 52. Blocks 4 and 5 were manual activities that compared the generated "a" and "b" coefficients for significance and similarity. After the comparisons, a final set of "a" and "b" coefficients were selected for each potential DC location. Block 5 was necessary to allow the possibility of using common sets of coefficients for several potential DC's instead of an individual set for each potential DC. The output of this supporting data system activity was the preparation of the LREPS exogenous inputs, DCP(3) and DCP(4).

In order to determine the weighted average freight rates used in the prior computer program, the third supporting computer activity developed the percentages of total weight shipped in specified order weight classes. These weight class percentages were calculated in a FORTRAN program from the accumulated weight sales of the individual customer orders that had been summarized into order blocks and written on a tape file in the D&E's "Order File Generator." These percentages were then used to calculate the weighted average freight rates from the potential DC's to their randomly selected DU's.



Program to develop potential DC a & b coefficients

Process each potential DC location

Process sample of destination points for DC

Do regression analysis for potential DC and its sample destination points

Loop through all sample destination points for DC

Loop through all potential DC's

Compare potential DC a & b coefficient results for similarity

Select final set of potential DC a & b coefficients

End of analysis

Figure 52.--DC-DU OBT, Freight Rate Regression Analysis

No block diagram is shown for this program since it was a very simple summarization and division program. This program also affected LREPS data base variables DCP(3) and DCP(4). Although this last program was simple, it had a very serious effect on the development of the potential DC's "a" and "b" coefficients. By changing the specified order weight classes or the composition of the individual customer orders summarized in the "Order File Generator," the "a" and "b" coefficients could be significantly affected.

CHAPTER VI

THE MONITOR & CONTROL SUBSYSTEM

Inputs, Outputs and Data Base Requirements

The MONITOR & CONTROL (M&C) subsystem had the responsibility of monitoring the activities of the operating system plus controlling the feedback of information that was used in making future decisions. The monitoring activities included the supervision of the sequence of events and activities over simulated time, the scheduling of the fixed-time events, and the gatekeeping activities associated with the inflow and outflow of information linked to the operating system. The controlling activities included reviewing and developing the feedback responses plus updating scheduled changes into the physical distribution system configuration.

The inputs into this subsystem came not only exogenously through the gateway but also from the MEAS subsystem. The MEAS subsystem passed to the M&C subsystem the criteria for evaluating the simulated, distribution plan that was in effect over the last quarter of simulated time.

The activities of the M&C subsystem happened on both a daily basis, when considering the operation of the D&C and OPS subsystems, and quarterly when considering MEAS and M&C itself.

The OPERATING SYSTEM is divided into two main sections. The first section includes the supervisory activities which are grouped under the name of the model MONITOR. The MONITOR controlled the order of and, in some cases, performed the activities of the operating system. The MONITOR is further subdivided into three sections. The first section controlled the order with which activities were executed. This subsection of the MONITOR is called the EXECUTIVE. The EXECUTIVE had the responsibility of picking the next imminent event from a file that was chronological and first in-first out (FIFO) in order, and calling the event activities that were to be executed at that discrete point in simulated time. The role of the EXECUTIVE in this model was handled by the GASP simulation language subprograms. The GASP routines used in LREPS are listed in the OPERATING SYSTEM section of this chapter. These GASP routines are not discussed in detail.

The LREPS events, that were utilized by the GASP EXECUTIVE in controlling the activities of the operating system, are generally explained via a system flowchart of LREPS data base variables, a logical block diagram of activities, plus any peculiar programming techniques. The programming languages will not be discussed since all the events were programmed using basic FORTRAN instructions. The event activities to be discussed are:

1) DAILY--the event to initiate the normal sales processing activities for the day;
2) MONTH--the event to initiate the processing of beginning-of-month activities;
3) QUAR--the event to initiate the processing of end and beginning-of-quarter activities associated with the operating system's information input and output plus the control of the feedback responses;
4) HYR--the event that basically initiated the same activities as QUAR but did some additional half-year activities;
5) YR--the event that was basically the same as HYR but did additional yearly activities;
6) BOCYC--the event that initialized the LREPS operating system for program execution;
7) EOCYC--the event that completed the activities of a simulation cycle plus generated the necessary information to terminate the LREPS operating system;
8) MCORAR--the variable-time event that initiated the OPS subsystem processing of a MCC order arrival;
9) DCSHPAR--the variable-time event that initiated the OPS subsystem processing of a DC shipment arrival.

The last two events are not discussed in this chapter since they were already discussed in the OPS subsystem. In addition to the above subprogram events, the user-written MAIN program and EVENTS subprogram required for GASP program execution are also discussed in this first subsection of the model MONITOR.

The second subsection of the MONITOR is the section dealing with the gateway of information into and out of the operating system. In this section three quarterly activity subprograms are discussed:

1) EOQ--the activity routine that supervised the development of the model outputs plus initiated the actual outputting of information from the operating system;

2) EXOG--the activity routine responsible for reading in the operating system's exogenous inputs;

3) BOQ--the activity routine responsible for the initialization of endogenous data base variables at the DU, DC and MCC stages of the model for the coming quarter's activities.

The last subsection of the OPERATING SYSTEM'S MONITOR is the activities associated with the scheduling of the fixed-time events. Given that the model had been defined as a hybrid system that had both variable-time and fixed-time events, this activity routine, called SCHED, scheduled the fixed-time events.

The second major component of the OPERATING SYSTEM is the model CONTROLLER. The CONTROLLER generated the feedback responses that affected the future actions of the operating system. The CONTROLLER, which was composed of FORTRAN, EXTENDED FORTRAN and assembly routines, had these specific system activities:

1) REVIEW--the group of subprograms that reviewed and developed the endogenous feedback responses and which were divided into these quarterly activities:
   a) DCSMOD--the activity of calculating the DC and regional sales forecast modification factors;
   b) RWRC--the activity of calculating a new regional total to tracked product weight ratio for MCC shipment weight extrapolation;
   c) INVMGT--the calculation of new inventory control variables used by the OPS subsystem;
   d) LOCATE--the facility location algorithm for determining and scheduling any DC facility system additions or deletions;
   e) EXPAND--the scheduling of a DC facility expansion given a DC was reaching its capacity.
2) UPDATE--the following group of subprograms that activated any exogenously or endogenously

scheduled activities for changing the simulated PD system:

a) PUTDCN--the quarterly activity of making any scheduled DC facility additions;

b) DELDC--the quarterly activity of making any scheduled DC facility deletions;

c) MODIFY--the quarterly activity of making any scheduled DC facility expansions;

d) DCDU--the quarterly activity of generating the information needed by the other LREPS subsystems for the DU to in-solution DC links.

The SUPPORTING DATA SYSTEM has three computer activities for the M&C subsystem. These three routines are discussed in reference to the logical activities, the programming language(s) used, the information and the specific LREPS exogenous data base variables related to the program, plus any peculiar programming problems and techniques.

The SUPPORTING DATA SYSTEM programs are:

1) Data Preparation Run--the preparation of the operating system's exogenous input tape;

2) DC to DU Packing Routine--a program for the packing of the feasible DC's that could serve each of the DU's;

3) DC X and Y Coordinates--the calculation of the potential DC converted, rectangular X and Y coordinates.

Operating System

As was mentioned in the first section of the chapter, the M&C OPERATING SYSTEM has two main subsections. The first subsection is called the MONITOR which supervised the activities of the operating system. The second subsection is called the CONTROLLER. The CONTROLLER reviewed and developed the feedback responses of the operating system plus updated any changes to the PD system that were necessary.

Monitor

The MONITOR is further subdivided into (1) the EXECUTIVE, (2) the "gatekeeper" of information into and out of the LREPS operating system plus (3) its "fixed-time scheduler." Each of these subsections are now discussed.

Executive.--The EXECUTIVE controlled the order with which the activities of the operating system were executed. The role of the EXECUTIVE was handled by the GASP EXECUTIVE plus the user-written subroutine EVENTS. In addition, the MAIN program routine for effecting computer program execution is included as part of the EXECUTIVE since it was indirectly involved in the order in which activities of the operating system were executed.

The MAIN routine had the responsibilities of (1) initializing the unit numbers of the input and output files, (2) buffering in the first group of customer orders and (3) providing the link to the GASP routines and the user-written EVENTS routine to begin program execution. The end of computer program execution occurred in this MAIN routine after one or more simulation cycles through a planning horizon could have been accomplished. The GASP feature of being able to stack a series of simulation cycles was used to be able to run more than one cycle during any one execution of the LREPS operating system.

The GASP routines used in the simulation are listed below:

GASP	Main GASP routine.
DATAIN	Routine to read in initial GASP data cards.
SET	Routine to initialize filing arrays, update pointer system, and maintain statistics on the number of file entries.
FILEM	Filed entries in file JQ of arrays NSET and QSET.
FINDN	Found event in array NSET with an attribute equal to a specific value.
RMOVE	Removed entries from file JQ of arrays NSET and QSET.
PRNTQ	Computed and printed time-integrated statistics on the number of entires in a file.
SUMRY	Basic output routine of GASP which printed its statistical data.

MONTR Routine that provided the capability to selectively monitor events for debugging and program tracing.

ERROR Called when an error was detected in any GASP subroutine except PRNTQ, SUMRY and MONTR, all of which printed their own error messages.

RNSET Routine that generated "seed" for random number generator.

DRAND Routine that generated a uniformly distributed random variate in interval 0 to 1.

LOCAT For GASP IIA, this routine converted a one-dimensional array, such as NSET, to a two-dimensional representation.

OTPUT Routine that allowed LREPS to write out additional, printed results of simulation activities over a past period.

RNORM Routine that generated a normally distributed random variate.

XMAX Routine that selected the maximum of two numbers.

The above GASP routines were implemented on the CDC 6500. These routines are not the complete set of routines available with the GASP simulation language. Only those basic routines required to operate the LREPS operating system were used. Other statistical summary routines were not used. A detailed discussion of each of the utilized GASP routines is given in Simulation with GASP II.[1]

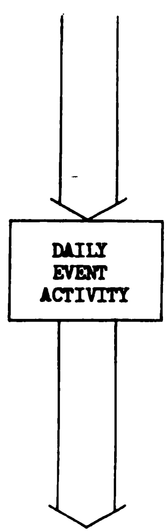
Along with the GASP routines, the EXECUTIVE includes the user-written subroutines EVENTS, DAILY, MONTH, QUAR, HYR, YR, BOCYC, EOCYC, MCORAR and DCSHPAR. These discrete-time events allowed LREPS to start and end the activities which took place in the computer model's operating system.

The EVENTS subprogram was called from GASP. After being called, EVENTS called the next imminent event based on the event code passed to it from GASP. After the appropriate event's activities were processed, EVENTS then returned to GASP. No system flowchart and block diagrams are included for this EVENTS routine since it was merely a set of subprogram linkages.

The first event activity to be discussed is the DAILY fixed-time event. Figure 53 identifies the data base variables associated with this routine. In addition to the LREPS data base variables, the variable TNOW that GASP used for keeping track of simulated time was also input into this routine.

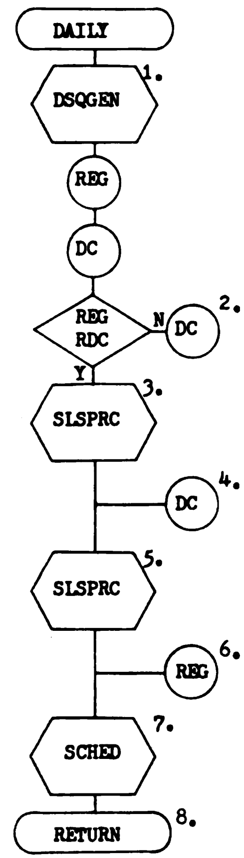
Activity Block 1 in the logical block diagram, Figure 54, for DAILY shows a link to the D&E routine that generated the daily domestic sales dollar forecast. No LREPS data base variables are associated with this activity block.

Activity Block 2 is associated with the activity of processing the RDC's in a region before the PDC was processed. This initial processing of the region's RDC's before its PDC was necessary in order to make sure that all customer orders for both the PDC and any RDC-P's were processed before doing the PDC's end-of-day activities. The data base variables that were used to see if the RDC was in this PDC's region were DCP(8) and IDCN01 which was set using DCIS(2).



NAME	DESCRIPTION
INPUTS	
NREGS	No. of in-solution regions
NUMDCS	No. of in-solution DC's
DCIS(2)	Potential DC in-solution
DCP(8)	DC type + PDC assignment
TNOW	GASP day of simulated time
OUTPUTS	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
IREGNO	Region being processed

Figure 53.--Daily Event Processing Variables



Routine to initiate processing of fixed-time daily activities

Go to D&E routine to generate daily domestic forecast

Process one region at a time

Process DC's in region--RDC's first, then PDC

Check to see if RDC is in this region

Go to D&E routine to process the RDC's normal daily activities

Loop through all in-solution RDC's

Go to D&E routine to process the PDC's normal daily activities

Loop through all regions being processed

Go to M&C fixed-time scheduler to generate next fixed-time event

Return to M&C calling routine

Figure 54.--Daily Event Activity Routine

After selecting a RDC that was in the region being processed, Activity Block 3 is a link to the D&E routine, SLSPRC, that initiated the processing of the normal daily activities associated with an in-solution DC. The variables IDCNO and IDCNO1 were passed to SLSPRC. Activity Block 4 signifies the checking of all RDC's that were in-solution. Data base variables NUMDCS and NREGS were used for this checking procedure.

After all the RDC's had been processed, the PDC for the region was then updated for its own normal daily activities. The D&E routine SLSPRC was again called to do this processing in Activity Block 5. The data base variables passed to the D&E routine were IDCNO and IDCNO1 after it had been set with IREGNO. IDCNO1 was validly set using IREGNO since the assumption had been made that a region had only one PDC and, consequently, the region number was also used as the PDC number.

Activity Block 6 used data base variable NREGS to process all in-solution regions. It should also be noted that LREPS was designed to process less than the total number of defined market regions. This was the reason for the variable NREGS.

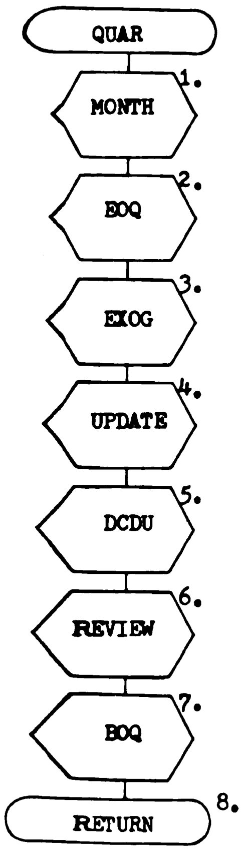
Activity Block 7 is a link to the MONITOR's scheduler of fixed-time events. This was the only location in the LREPS operating system where SCHED was called since DAILY always happened at each discrete point of simulated time. The GASP variable TNOW was passed to SCHED from DAILY.

Block 8 signifies a return back to the routine that called DAILY. DAILY could have been called from either the GASP EXECUTIVE or from one of the other fixed-time event activities.

The MONTH event activity is not discussed in reference to a block diagram or data base system flowchart. The monthly routine performed two major activities. The first activity is a link to the DAILY routine to process the customer sales for the simulated end-of-month day. The other activity includes the addition of a monthly sales forecast increment, XINC, to TDSF, the total annual domestic forecast. This last activity will be explained in conjunction with the YR event activity.

The QUAR event activity is only discussed in reference to the block diagram in Figure 55 since it was merely a set of linkages to other routines in the M&C. Activity Block 1 is the linking to the event activity MONTH to process the normal sales activities for the last day of the quarter. After processing these sales, the EOQ routine was called to develop the remaining measures of output plus output the results of LREPS for the past quarter.

Activity Block 3 then read in any exogenous inputs into the LREPS common data base. After inputting the exogenous changes, Activity Block 4 updated the PD system for any scheduled DC changes plus any changes that resulted from reading the exogenous input data. After



Routine to process quarterly activities

Link to routine to process the activities for the last month of the quarter

Link to routine to process end-of-quarter activities

Link to routine to read LREPS exogenous inputs

Link to routines to make DC facility changes

Link to UPDATE routine to develop DC to DU link information based primarily on exogenous inputs

Link to routines to do PD system review activities based primarily on endogenous variables

Link to routine to process beginning-of-quarter initialization activities

Return to GASP EXECUTIVE or other calling routine

Figure 55.—Quarterly Event Activity

making possible changes to the PD system, Activity Block 5 called the UPDATE subprogram DCDU to develop DC to DU link information that resulted from primarily changes in the exogenous data base variables but also from changes in the DC's that were now in solution. Activity Block 6 then passed through all the M&C routines that controlled and developed the feedback responses of the model.

Given the stages and the links between stages that were now in solution, Activity Block 7 called the routine to initialize the necessary endogenous variables to process the coming quarter's activities. Activity Block 8 returned control back to the GASP EXECUTIVE, if QUAR was called directly, or to the calling routine if it was called from another event activity.

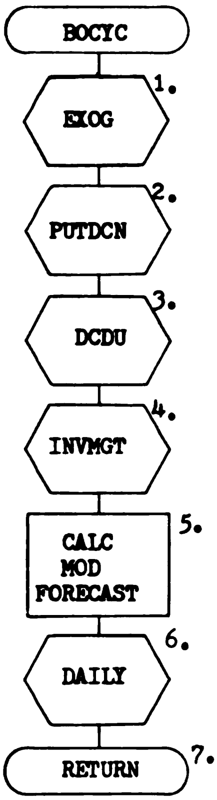
The HYR event activity was placed in LREPS for future model flexibility. Therefore, HYR only called QUAR and will not be discussed further.

The event activity for the annual processing of activities for the beginning and end of year is called YR. This routine linked to the QUAR event to process the normal end- and beginning-of-quarter activities and then performed three tasks. Task one was to increment data base variable IYEAR by one. Task two calculated a new monthly sales forecast increment, XINC. XINC was calculated as a monthly average of the difference between last year's and the coming year's annual domestic sales dollar forecast, TDSF. The last task was to modify TDSF with the negative product of five and one-half times XINC. In order to develop a linear trend in domestic forecasted sales that increased in constant monthly increments of XINC, these last two tasks were performed.

The BOCYC event activity is explained in reference to a block diagram but no data base system flowchart. This event was exogenously set, along with the EOCYC event, through the GASP input cards. The BOCYC event was used to initialize the execution of a cycle through the planning horizon which was limited in length to the time limit set in the EOCYC event.

Activity Block 1 of Figure 56 initialized the LREPS common data base for program execution by a link to "EXOG." Activity Block 2 used some of the previously read exogenous data, particularly data base variable DCP(12), to put into solution the starting DC facility configuration. This DC facility initialization was performed by the UPDATE subprogram, PUTDCN.

Activity Block 3 then developed the necessary DC to DU link information while the next activity block calculated the values for the beginning inventory levels and inventory control variables for each of the starting DC's. Activity Block 5 performed similar activities as those in the YR event routine for calculating the monthly trend factors for the first year's annual domestic sales forecast. Finally, Activity Block 6 was called to process the first day's sales activities plus begin scheduling



Beginning of cycle event activity

1. **Link to routine to read LREPS exogenous inputs**

2. **Link to routine to make DC facility initialization**

3. **Link to routine to develop DC to DU link information**

4. **Link to routine to initialize DC inventories plus calculate inventory parameters**

5. **Calculate modified annual domestic forecast**

6. **Link to routine to process first day's activities**

7. **Return to GASP EXECUTIVE**

Figure 56.--Beginning of Cycle Event Activity

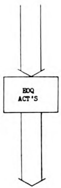
endogenously the fixed-time events. Block 7 returned control to the GASP EXECUTIVE.

As indicated above, the event activity EOCYC ended the execution of the LREPS operating system for a cycle. This exogenously set event was set one day after the last simulated year of activity. No system flowchart or block diagram are developed since the routine only set the GASP variable, MSTOP, equal to a negative one in order to stop cycle execution and then returned back to the GASP EXECUTIVE. As was mentioned earlier, GASP handled the stacking of simulation cycles through its input data cards although the stacking of cycles could have been done in this routine.

The only other event activities associated with this section of the MONITOR are MCORAR and DCSHPAR. These routines were discussed in the OPS subsystem. After their program execution, they returned to the GASP EXECUTIVE.

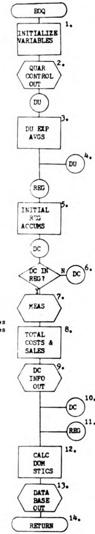
Input/Output Gatekeeper.--Figure 57 identifies the data base variables and Figure 58 the logical activities that were necessary for the development of the final model outputs plus those activities that physically outputted the information from the LREPS operating system for the Report Generator (RPG) system. These activities were associated with what is called the end-of-quarter (EOQ) activity routine.

Activity Block 1 initialized the variables needed for a quarterly control record that contained information



NAME	DESCRIPTION
INPUTS	
NDUS	No. of DU's being processed
DU(13)	QTD dollar sales--full-line DC
DU(14)	QTD dollar sales--RDC-P
DU(15)	QTD weight sales--full-line DC
DU(16)	QTD weight sales--RDC-P
DCP(8)	DC type & PDC assignment
NUMDCS	No. of in-solution DC's
NREGS	No. of in-solution regions
NMCCS	No. of in-solution MCC's
ITNSP	Total no. of tracked products
ITNPC	Total no. of inventory categories
MCCWB	MCC weight breaks
DCWB	DC weight breaks
OUTPUTS	
DU(9)	Exp avg dollar sales
DU(10)	Exp avg weight sales
NRUN	GASP variable
NAME(1-6)	GASP variables
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
REG(16)	Regional total PD cost for quarter
REGSAC	Regional quarterly dollar and wt sales
DOMSAL	Domestic quarterly dollar and wt sales
DOMAST	Domestic avg total OCT
DCIS(1-21)	In-solution DC data
DCCOST(1-6)	In-solution DC cost data
DCMCC(1)	Total no. of mult-prod reorders
DCMCC(2)	DC-MCC average reorder lead time
PRDC(3)	Average cubic inventory-on-hand
PRCTDC(1)	QTD product stockouts
PRCTDC(2)	QTD single-product reorders
OCTDIS(1)	NOCT sales dollars proportions
OCTDIS(2)	NOCT sales order proportions
WTACUM	DC shipment weight accums
DCMCC1	DC-MCC weight accumulators

Figure 57.--EOQ Input/Output Analysis



Routine to prepare and output end-of-quarter results

1. Initialize variables plus RPG report tape control record
2. Link to RPG subsystem via output of the quarterly control record of information

Process each DU

3. Prepare exponentially averaged DU sales dollars and weight
4. Loop through all DU's in-process

Process data by region

5. Reset regional sales accumulators

Develop each DC's measures of cost and service

6. Process only region's assigned DC's
7. Link to MEAS routines to calculate measures of service and cost
8. Sum PD total cost for DC plus update regional and domestic accumulators
9. Link to RPG subsystem via output of in-solution DC detail data
10. Loop through all in-solution DC's
11. Loop through all in-solution regions
12. Calculate domestic summary statistics
13. Link to RPG subsystem via output of LREPS common data base
14. Return to M&C QUAR event activity

Figure 58.--End-of-Quarter Activity Routine

identifying this simulation cycle, NRUN, plus additional identifying, alphabetic information, NAME(1) to NAME(6). The variables used in the control record were from the GASP data base which had been initialized with punched cards. Activity Block 2 wrote this control record on the RPG magnetic tape referred to in Chapter 2. This last activity used an EXTENDED FORTRAN instruction to buffer out this information.

Activity Blocks 3 and 4 looped through all the demand units to develop exponentially smoothed DU dollar and weight sales. The quarter-to-date sales, the sum of DU(13) plus DU(14) and the sum of DU(15) plus DU(16), were averaged with the past averages to arrive at new averages, DU(9) and DU(10). The alpha factor or smoothing constant was set internal to these basic FORTRAN instructions. The averages were developed for all in-solution DU's, NDUS.

Activity Block 5 used basic FORTRAN to reset to zero the regional and domestic sales accumulators used in succeeding activities. Data base variables REGSAC, DOMSAL and REG(16) were reset.

Activity Block 6 used data base variables DCP(8), IREGNO and IDCNO to see if the present DC was in the region being processed. If the DC was in this region, Activity Block 7 shows the link to the MEAS subsystem routines that calculated the measures of service and cost. Data base variables IDCNO, IDCNO1 and IREGNO were passed to the subprograms to identify the DC and region being processed.

Activity Block 8 then developed the total cost per DC, DCIS(6), and per region, REG(16), using data base variables DCCOST(1) to DCCOST(6) that were calculated in the preceding MEAS activities. In addition, the sales dollars and weight by region, REGSAC, were accumulated using data base variables DCIS(16) and DCIS(17).

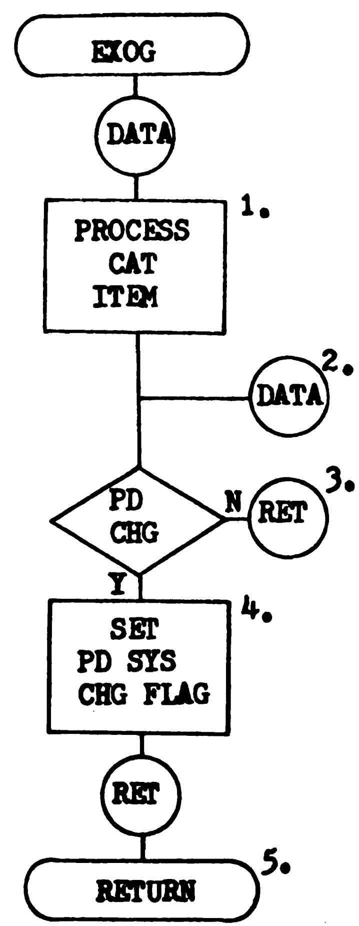
Activity Block 9 identifies the EXTENDED FORTRAN subprogram that outputted important DC information in a separate file for this quarter. The DC data base variables that were written on the RPG tape were DCIS(1) to DCIS(21), DCMCC(1) and DCMCC(2) using NMCCS, PRDC(3) using ITNSP, PRCTDC(1) and PRCTDC(2) using ITNPC, DCCOST(1) to DCCOST(6), OCTDIS(1) and OCTDIS(2), WTACUM using DCWB, DCMCC1 using NMCCS and MCCWB. Activity Block 10 used data base variable NUMDCS to loop through all in-solution DC's, while Activity Block 11 used data base variable NREGS to loop through all in-solution regions.

Activity Block 12 then developed the domestic measures of sales, DOMSAL, and customer service, DOMAST. These measures were developed using data base variables DCIS(16), DCIS(17), DCIS(7), DCIS(21) and NUMDCS. DOMAST was a weighted average total order cycle time based on DC customer orders.

Finally, Activity Block 13 is again a means of linking the LREPS operating system to the RPG system. This activity used EXTENDED FORTRAN instructions to write the entire LREPS common data base on the RPG tape for this quarter. All data base variables except those peculiar to any one of the major subsystems were written. With this last output operation, there were now three separate files of information on the RPG tape for a quarter. All three files of information would hopefully suffice any data analysis that was to be done in the RPG system. Block 14 identifies a return to the M&C QUAR routine.

Figure 59 identifies the logical activities associated with the M&C routine responsible for inputting the LREPS exogenous inputs into the operating system's data base. In addition, the routine could also set data base variables classified as endogenous in Appendix 1. The procedure with which the variables were set was to identify the sequential position of each LREPS variable within the common data base. This assigning of sequential numbers was called "cataloging the data base." As an example, if there were a total number of variables in the data base equal to ten thousand, then each of the variables was given a number from one to ten thousand depending upon its sequential position.

Given the cataloged data base, then the EXOG routine basically read a "catalog" of information for each variable or group of sequential variables that were to be set. A catalog contained two tape records. The first record contained not only the location or reference number of the first data base variable to be set, but also the



Routine to read in LREPS exogenous inputs

Process each catalog of information

Input catalog of LREPS data

Loop through all catalogs for quarter

Check to see if any exogenous PD system changes

Set PD system change indicator on

No exogenous PD system changes

Return to M&C QUAR event activity

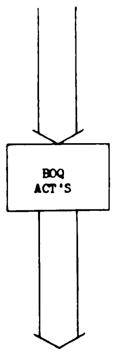
Figure 59.--LREPS Exogenous Input Routine

number of sequential items to be set after this first variable. The second record of the catalog contained the data that was to be read in. The catalogs were grouped by the quarter in which they were to be read. Given a five-year planning horizon, there would then be twenty quarterly files of catalogs. The first quarter's file initialized the total common data base.

Activity Blocks 1 and 2 identify the EXTENDED FORTRAN activities of reading a quarter's catalog of exogenous inputs. Activity Blocks 3 and 4 are associated with the testing of data base variables IUDF(2) to IUDF(5) for possible exogenous changes to the PD system being simulated. If there were exogenous changes, then data base variable PDSCHG was set in order to signal other M&C routines that information associated with PD system changes should be developed. After these activities had been completed, Block 5 of Figure 59 identifies a return to the M&C QUAR routine.

The last activity associated with the inflow and outflow of information from the LREPS operating system is the beginning-of-quarter (BOQ) activity of clearing or reinitializing certain endogenous variables in the LREPS common data base for the next quarter's activity. This routine was programmed in basic FORTRAN and initialized the data base variables listed in the output section of Figure 60.

Figure 61 shows the activities associated with reinitializing the variables for the different stages and



NAME	DESCRIPTION
INPUTS	
NDUS	No. of DU's being processed
NMCCS	No. of in-solution MCC's
NREGS	No. of in-solution regions
ITNSP	Total no. of tracked products
ITNPC	Total no. of inventory categories
OUTPUTS	
DU(13)	QTD dollar sales--full-line DC
DU(14)	QTD dollar sales--RDC-P
DU(15)	QTD weight sales--full-line DC
DU(16)	QTD weight sales--RDC-P
DCIS(1)	In-solution DC stage data
DCIS(6)	In-solution DC stage data
DCIS(7)	In-solution DC stage data
DCIS(9) to	
DCIS(21)	In-solution DC stage data
OCTDIS(1)	MOCT sales dollar proportions
OCTDIS(2)	MOCT sales order proportions
WTACUM	DC weight accumulations
DCMCC1	DC-MCC weight accumulations
DCMCC(1)	No. of mult-prod reorders
REG(4)	QTD tracked product weight sales
REG(13)	Regional actual service level
REG(16)	Regional total PD cost
REGSAC	Regional sales accumulators
PRDC(3)	DC time-integrated IOH
TPDEM(1)	QTD total domestic demand
PRCTDC(1) to	
PRCTDC(4)	DC inventory category variables
DOMAST	Domestic avg TOCT
PDTCST	Domestic total PD cost
DOMSAL	Domestic sales accumulators

Figure 60.—Beginning-of-Quarter Activity Variables

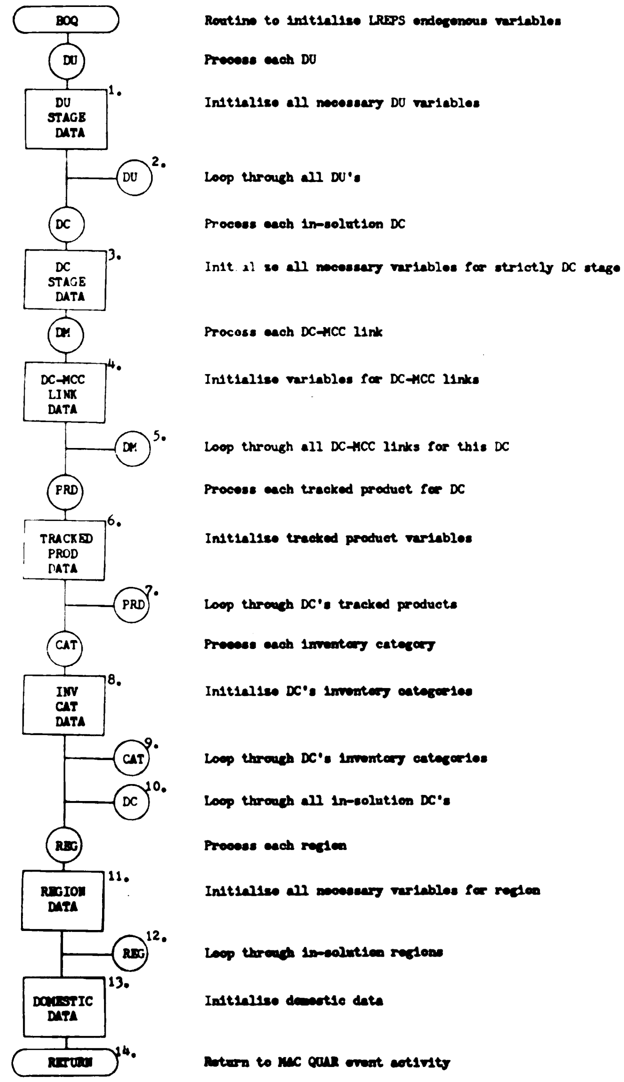


Figure 61.—Beginning-of-Quarter Activity Routine

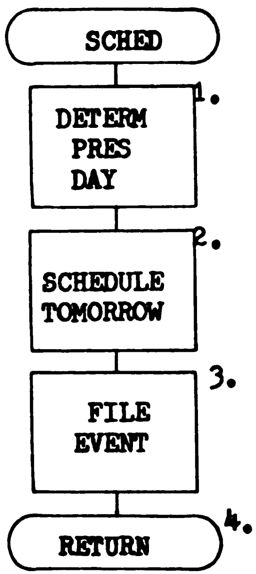
links in the PD system. Activity Blocks 1 and 2 initialized DU data base variables DU(13) to DU(16) using NDUS. Activity Block 3 reset DCIS(1), DCIS(6), DCIS(7), DCIS(9) to DCIS(21), OCTDIS(1), OCTDIS(2) and WTACUM to zero for each of the in-solution DC's. Activity Blocks 4 and 5 set DC to MCC link data base variable DCMCC(1) to one plus reinitialized data base variable DCMCC1 to zero for each of the MCC links, NMCCS.

Activity Blocks 6 and 7 reset PRDC(3) to zero for all tracked products, ITNSP. The DC inventory category variables PRCTDC(1) to PRCTDC(4) were reset in Activity Blocks 8 and 9 for all inventory categories, ITNPC. Block 10 shows that the above activities were done for each of the in-solution DC's, NUMDCS.

Activity Blocks 11 and 12 reset the regional data base variables REG(4), REG(13), REG(16) and REGSAC to zero using variables NREGS. The domestic variables TPDEM(1) for all tracked products, ITNSP, DOMAST, PDTCST, and DOMSAL were reset to zero in Activity Block 13. Finally, Block 14 identifies the return to the M&C QUAR routine.

Fixed-Time Scheduler.--The last component in the operating system's MONITOR is the fixed-time scheduler, SCHED, of events. As was previously mentioned in the first part of this chapter, there are nine event activities, each with their own event code. Seven of these event activities, all but BOCYC and EOCYC, were scheduled endogenously within the LREPS operating system. Given the assumption that LREPS would only process the workdays of a simulated year, two hundred and fifty-two days, SCHED, by means of NSCH, an exogenously set array of event codes and days of occurrence, scheduled the next endogenous fixed-time event for tomorrow. For instance, if today was day twenty of simulated time and there were twenty-one days in a month, then the event activity MONTH would be scheduled for tomorrow. If the day was sixty-two, then QUAR was scheduled. Figure 62 shows the basic activities associated with scheduling DAILY, MONTH, QUAR, HYR or YR. Block 1 determined the present day using the GASP variable TNOW. Block 2 used TNOW plus NSCH to schedule tomorrow. If "today" was not in the array NSCH, then the DAILY fixed-time event was scheduled. Activity Block 3 stored the fixed-time event, using the GASP subprogram FILEM, in the filing arrays NSET and QSET. The fixed-time event's attributes were the time of event processing, and its respective event code.

Because tomorrow was always scheduled every day, at any one time the maximum number of fixed-time events in the GASP filing arrays NSET and QSET was two. An endogenous fixed-time event plus the end-of-cycle event that was set exogenously were the two events. After scheduling the next fixed-time event, SCHED returned to the M&C DAILY routine that always called it.



Routine to schedule next fixed-time event

Determine the present workday in simulated time

Schedule appropriate fixed-time event for tomorrow

File event in the event filing array using GASP routine FILEM

Return to M&C DAILY event activity

Figure 62.--Fixed-Time Scheduler Routine

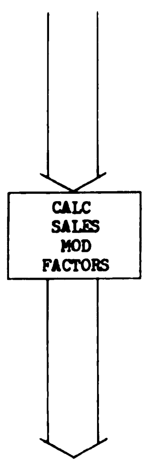
Controller

The operating system's controller is divided into two major sections. The first section reviewed and developed the feedback responses. This section is called REVIEW. The second section changed the PD system for any exogenously or endogenously scheduled PC changes. This section is called UPDATE.

Review.--The first subprogram in the REVIEW section calculated the DC and regional, exponentially averaged factors for modifying forecasted sales. These factors were based upon actual and desired levels of customer service. The customer service levels were defined in terms of average total order cycle time.

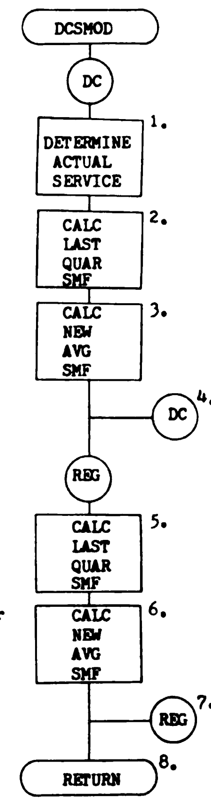
Figure 63 identifies the data base variables associated with this subprogram called DCSMOD. Figure 64 identifies the logical activities in calculating the sales modification factors DCIS(4) for each in-solution DC and REG(14) for each in-solution region.

Activity Block 1 determined the proportion of dollar sales, DCACTS, that was delivered within the specified customer service time that management desired for the nation, DAYCON. The proportion was interpolated from the proportions that had been accumulated in the normal order cycle time array, OCTDIS(1). Since the management parameter, DAYCON, was based on the TOCT, it was reduced by the back order penalty time T3 in data base variable DCIS(1). Then the correct set of proportions from



NAME	DESCRIPTION
INPUTS	
DAYCON	Cust total OCT decision parameter
OCTDIS(1)	NOCT sales dollar proportions
DCIS(1)	DC cust T3 penalty time
DCIS(5)	DC cumulative weighted index
NUMDCS	No. of in-solution DC's
NREGS	No. of in-solution regions
REG(5)	Desired level of service for region
REG(8)	Regional cumulative wt index
DCP(8)	DC type + PDC assignment
KEY ENDOGENOUS VARIABLES	
Local	
DCACTS	DC actual service
Common	
IDCNO	In-solution DC being processed
IREGNO	No. of region being processed
OUTPUTS	
DCIS(4)	DC sales modification factor
REG(13)	Last quarter's reg sales modification factor
REG(14)	E.A. regional sales modification factor

Figure 63.—Sales Modification Factor Calculation Variables



Return to calculate DC and regional factors

Process each DC

Determine actual level of service measure to be used for SMF

Calculate SMF for last quarter

Calculate new exponentially averaged DC SMF

Loop through all in-solution DC's

Process each region

Determine actual level of service measure and calculate SMF for last quarter

Calculate new exponentially averaged regional SMF

Loop through all in-solution regions

Return to M&C QUAR event activity

Figure 64.—Sales Modification Factor Calculation Routine

OCTDIS(1) could be selected, using DAYCON less DCIS(1), within which the interpolation of the final proportion occurred.

Activity Block 2 divided the DC's service proportion DCACTS, by the regional desired level of service, REG(5). Block 3 then exponentially averaged last quarter's sales modification factor, DCACTS, with the old factor, DCIS(4), to arrive at a new SMF for the DC. This new SMF was stored in DCIS(4). Block 4 indicates that the above procedure was done for all in-solution DC's, NUMDCS.

After the DC SMF's had been calculated, then the new regional SMF's, REG(14), were calculated. Block 5 indicates that the region's SMF, REG(13), for the last quarter, was first calculated. REG(13) was a weighted average of the DC's sales modification factors, DCIS(4), based on the DC cumulative weighted indices, DCIS(5), and the weighted index for the region, REG(8).

Block 6 shows the activities associated with the calculation of the new exponentially averaged regional SMF, REG(14). For both the new exponentially averaged DC and regional SMF's, the alpha factors were built directly into the basic FORTRAN instructions.

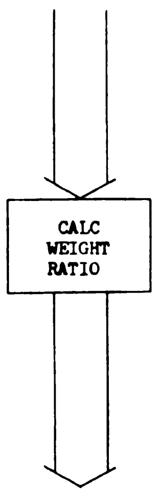
The next routine to be discussed is RWRC. The purpose of this routine was to calculate a new regional ratio of total product line to tracked product line weight sales. This ratio was used in the OPS subsystem to determine the total replenishment weight shipped from a MCC to a DC. Figure 65 shows the data base variables involved in the calculation of these ratios. Figure 66 is the block diagram of logical activities for this basic FORTRAN subprogram.

Activity Block 1 reset to zero an accumulator, SUMWT, for the region's total customer weight sales. Block 2 then checked the next DC, IDCNO, to see if it was in the region, IREGNO, being processed. Data base variables DCP(8), DCIS(2) and IDCNO1 were used in this checking process.

Given that the DC was in the region being processed, then its QTD weight sales, DCIS(17), were accumulated in SUMWT. Block 4 indicates that all in-solution DC's, NUMDCS, were checked for each region.

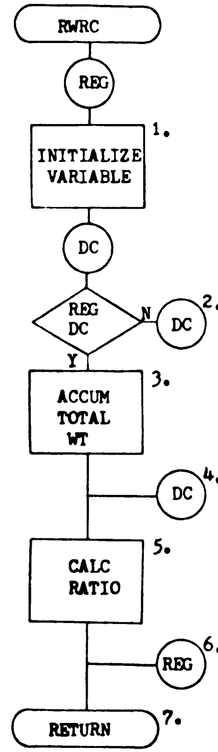
After the region's total weight sales had been accumulated, then the total tracked product weight sales, REG(4), that had been accumulated in the OPS subsystem, was divided into SUMWT. This quotient, the new regional weight ratio, replaced the old ratio in data base variable REG(15). Block 6 indicates that these ratios were calculated for all in-solution regions, NREGS. Block 7 shows a return to the M&C quarterly event activity.

The next major activity was concerned with the calculation of the variables used in the inventory control operations of the OPS subsystem. These variables were calculated according to the management specifications as to the type of inventory control policy plus the policy's



NAME	DESCRIPTION
INPUTS	
NUMDCS	No. of in-solution DC's
NREGS	No. of in-solution regions
REG(4)	QTD tracked product weight sales
DCIS(2)	Potential DC pointer
DCIS(17)	QTD DC total weight sales
DCP(8)	DC type + PDC assignment
KEY ENDOGENOUS VARIABLES	
Local	
SUMWT	Total QTD regional wt sales
Common	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC code being processed
IREGNO	No. of region being processed
OUTPUTS	
REG(15)	Regional weight sales ratio

Figure 65.--Regional Weight Ratio Calculation Variables



Routine to calculate new regional weight ratio

Process each region

Initialize variable to accumulate DC total sales

Process each in-solution DC

Check to see if DC in region

Accumulate DC's weight to region's total weight

Loop through all in-solution DC's

Calculate weight ratio for region

Loop through all in-solution regions

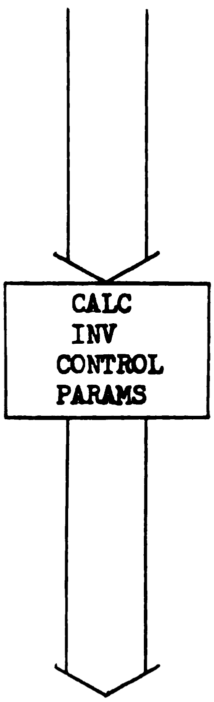
Return to M&C QUAR event activity

Figure 66.--Regional Weight Ratio Calculation Routine

respective parameters. Since all of a company's products might not have been tracked, the management specifications were set by inventory category and not by each of the model's tracked products.

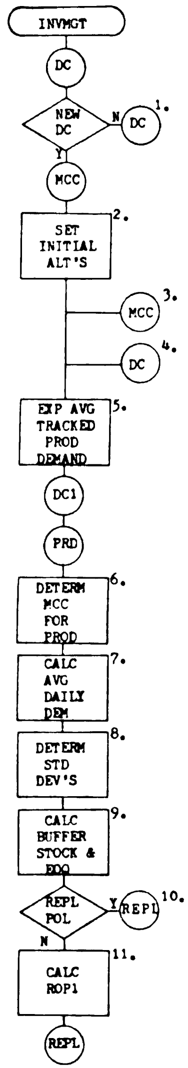
The data base variables used in this subprogram, which used both assembly and FORTRAN routines, are listed in Figure 67. The logical activities are identified in Figure 68. This subprogram was called at both the beginning of cycle to set the initial values and quarterly to update the values of the control parameters from the results of past sales activity. This recalculation of the inventory control parameters was the basis for calling this section of LREPS a dynamic inventory control model.

Examining the activities of this routine in detail, Activity Block 1 checked data base variable PRDC(2) to see if it contained a negative one that would have been set in the UPDATE routine PUTDCN if a new DC was coming into solution. If it did contain a negative one, then Activity Blocks 2 and 3 calculated the initial average reorder lead times between this DC, IDCNO, and each of its supplying MCC's. Data base variables DCIS(2), IDCNO1 set from DCIS(2), DCP1(3), PARM, MCC(1) plus a constant T3 were used in setting this initial average lead time, DCMCC(2). The initial lead time was calculated as the sum of the averages of the reorder lead time elements T1, T2, T3 and T4 for each of the supplying DC-MCC links, NMCCS. Block 4 indicates that each of the in-solution DC's was checked to see if it was a new DC.



NAME	DESCRIPTION
INPUTS	
DCIS(2)	Potential DC pointer
DCIS(5)	DC cumulative weighted index
NUMDCS	No. of in-solution DC's
NMCCS	No. of in-solution MCC's
DCP1(1)	DC-MCC tracked product supply links
DCP1(3)	DC-MCC T1 & T4 factors
ITNSP	Total no. of tracked products
ITNPC	Total no. of inventory categories
TPDEM(1)	QTD tracked product case sales
PRCT(2-7)	Inventory category variables
DICC	Daily inventory carrying charge
PARM	Service time variance functions
MCC(1)	MCC T2 order prep time factors
NWKDYS	No. of workdays per simulated year
CGSCS	Tracked product cost of goods sold
KEY ENDOGENOUS VARIABLES	
Local	
DEM	DC tracked product avg daily demand
BUF	DC tracked product buffer stock
EOQ	DC tracked product EOQ
SLT	DC std dev lead time
SRP	DC std dev review period length
SPD	DC std dev avg daily demand
ORDCST	Tracked product ordering cost
Common	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
OUTPUTS	
PRDC(1)	DC tracked product ROP's
PRDC(2)	DC tracked product S-level
PRDC(4)	DC tracked product initial inventory level
DCMCC(2)	DC-MCC avg reorder lead time
TPDEM(2)	Exp avg tracked product case sales

Figure 67.--Inventory Management Variables



Routine to calculate inventory control parameters

Process each in-solution DC

1. Check to see if new DC brought into solution

If so, calculate DC-MCC initial average reorder lead times

2. Calculate initial average lead time

3. Loop through all MCC links for this DC

4. Loop through all in-solution DC's

5. Exponentially average tracked products' domestic demands

Process each in-solution DC

Process all tracked products for DC

6. Determine the MCC supplying the product to the DC

7. Calculate the DC's average daily demand for this tracked product

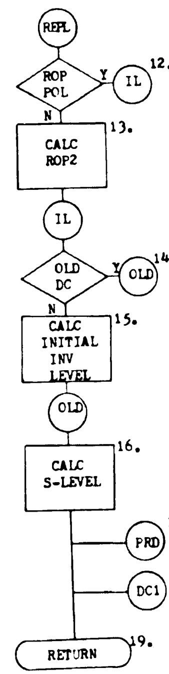
8. Determine standard deviations of lead time, review period and average daily demand

9. Calculate buffer stock and EOQ for tracked product

10. Check to see if tracked product uses replenishment policy

11. Calculate ROP1 for tracked products for use with either reorder point or hybrid policy

Routine to see if product uses ROP2



Routine to see if product uses ROP2

12. Check to see if tracked product uses reorder point policy

13. Calculate ROP2 for tracked products for use with either replenishment or hybrid policy

Routine to calculate inventory levels

14. Check to see if initial inventory level needed

15. Calculate initial inventory level for this new DC coming into solution

Old DC; bypassed setting of IOH

16. Calculate tracked product's S-level

17. Loop through all tracked products

18. Loop through all in-solution DC's

19. Return to M&C GASP EXECUTIVE activity

Figure 68.--Inventory Management Routine

In order to predict the level of quarterly sales in the next quarter of simulated time, Activity Block 5 calculated an exponentially averaged total product demand, TPDEM(2), for each of the tracked products, ITNSP. The past quarter's sales activity, TPDEM(1), was averaged with the old tracked product's prediction, in data base variable TPDEM(2), to determine the new prediction. Again as in other places where exponential averaging had been used, the alpha factors were built directly into the FORTRAN instructions.

After calculating the above predictions for the entire domestic market area, the remaining activity blocks calculated the specific tracked product inventory control variables for each in-solution DC. Activity Block 6 first determined the MCC supplying the tracked product being processed. This activity used EXTENDED FORTRAN to shift out the MCC number from DCP(1) using the potential DC code, IDCN01.

After determining the MCC supplying the product, the next activity calculated the average daily demand for the product from the DC being processed. The average daily demand, DEM, was the product of the new domestic prediction, TPDEM(2), the DC's cumulative weighted index, DCIS(5), plus the reciprocal of the number of workdays in a simulated quarter of activity which was NWKDYS divided by four.

After the average daily demand was calculated, three standard deviations were calculated in Activity Block 8. SLT, the standard deviation of reorder lead time, was calculated as the square root of the average lead time, DCMCC(2), assuming the lead times were Poisson distributed. SRP, the standard deviation of the review period length, was calculated as the quotient of the average review period length, PRCT(6), divided by the square root of twelve. SRP's calculation assumed a uniform distribution. SPD, the standard deviation of daily demand, was calculated as the square root of the average daily demand, DEM. This calculation also assumed the demands were Poisson distributed. These activities used basic FORTRAN instructions plus the computer system library function SQRT.

After calculating these basic values, Activity Block 9 calculated the tracked product's buffer or safety stock, BUF, plus its economic order quantity, EOQ. BUF was the product of data base variable PRCT(7) and the sum of SLT times SPD and DEM times SRP. EOQ used the standard EOQ formula plus data base variables TPDEM(2), ORDCST, DICC, and CGSCS.

After calculating the buffer stock and economic order quantity, Activity Blocks 10 to 13 calculated the reorder points used in the alternative inventory control policies that were built into LREPS. Block 11 calculated ROP1 if the tracked product used a daily reorder point or hybrid system. Data base variables PRCT(2), PRCT(3), ITNPC and PRCT(5) were used to determine the product's type of policy. If the product used a daily reorder point or a hybrid system, then ROP1, data base variable PRDC(1), was calculated as the sum of BUF plus the product of DCMCC(2) and DEM. At this point in the program, ROP2 was also set equal to ROP1.

If the tracked product used either a replenishment or hybrid system, then Activity Block 13 calculated ROP2 as the sum of BUF, the product of DEM times DCMCC(2) and DEM times one half of PRCT(6). The calculated ROP2 was then packed, using EXTENDED FORTRAN instructions, into PRDC(1) along with ROP1.

As was similarly done in Activity Block 1, data base variable PRDC(2) was checked in Activity Block 14 for a negative one to see if the DC being processed was a new DC. If it was a new DC, then the DC's initial inventory levels had to be determined. These initial inventory levels, PRDC(4), were set in Activity Block 15 as the sum of EOQ plus BUF.

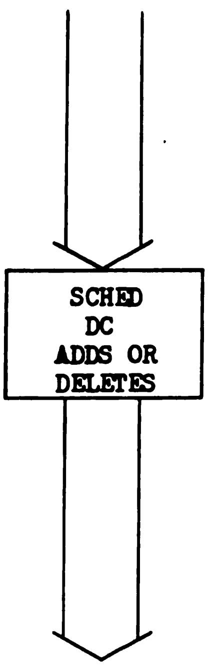
Activity Block 16 then calculated the standard S-level, data base variable PRDC(2), used for all LREPS' inventory policies. The S-level was the sum of EOQ and ROP2. The above equation was also correct for the daily reorder point system since ROP2 was set equal to ROP1 for that policy.

Finally, Activity Blocks 17 and 18 show that the above activities were done for each tracked product, ITNSP, and all in-solution DC's, NUMDCS. Activity Block 19 identifies the return back to the calling GASP EXECUTIVE activity.

The above inventory management routine is a module that is hopefully general enough to handle the policies of many companies. If the routine is not, however, satisfactory for a certain company's operations, then the module can be replaced with the specific policies that the company desires. This substitution of a different module was successfully accomplished in a test case for LREPS.

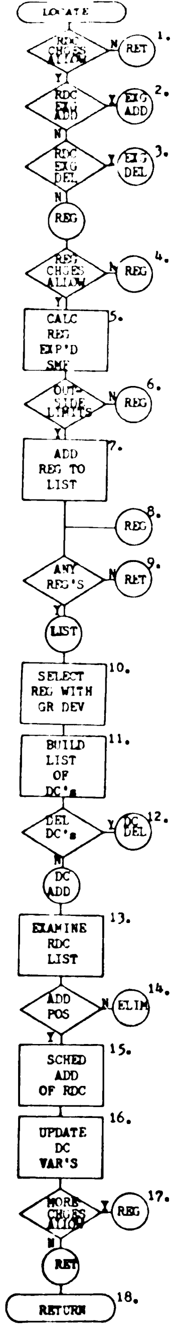
The facility location algorithm, called LOCATE, is probably the most sophisticated user-written routine in LREPS. This subprogram was responsible for the scheduling of the addition or deletion of DC's in the PD system configuration. A basic assumption was that LOCATE would only add or delete facilities but not add <u>and</u> delete DC's in the same quarter. Depending upon the constraint values, the review of the DC system configuration was done, at the most, quarterly.

Figure 69 shows the data base variables involved in this routine, while Figure 70 shows the logical activities. This routine was composed of a number of subprograms. Assembly, basic FORTRAN and EXTENDED FORTRAN instructions were used throughout the entire routine.



NAME	DESCRIPTION
INPUTS	
NATCON	Domestic constraints on-off flag
INVSTS	Total inv for in-solution DC's
MXINVS	Max total inv. for in-solution DC's
INVSPS	Total inv for in-process DC's
MXIVPS	Max total inv for in-process DC's
NMINPS	Dom no. of in-process add DC's
NMBDLS	Dom no. of in-process delete DC's
MXIPAS	Max no. of in-process DC's
MAXDCS	Max no. of allowable DC's
NUMDCS	No. of in-solution DC's
IUDF(1)	PD facility changes flag
REGCON	Regional constraints' on-off flag
REG(1)	Max no. of DC's allowed
REG(2)	Max no. of DC's that can be added
REG(3)	Max no. of DC's that can be deleted
REG(6)	Max in-solution DC investment
REG(7)	Max in-process DC investment
REG(9)	No. of in-process add DC's
REG(10)	No. of in-process delete DC's
REG(11)	Total in-solution DC investment
REG(12)	Total in-process DC investment
REG(17)	No. of in-solution DC's
REG(14)	Exp avg SMF
NREGS	No. of in-solution regions
NDUS	No. of in-solution DU's
DCASGN	List of feasible DC's per DU
DCCAPC	DC dollar size capacities
TNOW	GASP day of simulated time
TMINPR	Delay time to add DC
DCIS(6)	In-solution DC total PD cost
DCIS(17)	In-solution DC QTD weight sales
NDCADD	Max no. DC's put in-process/quarter
KEY ENDOGENOUS VARIABLES	
<u>Local</u>	
RLCSMF	Expected regional SMF
RLIST	List of reg SMF's plus later eligible DC list
DCSUM	Eligible DC's sum of DU sales
NDCIN	No. of DC's in-process this quarter
<u>Common</u>	
IDUNO	No. of DU being processed
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
IREGNO	No. of region being processed
OUTPUTS	
DCP(12)	Time of DC change + type change
REG(9)	No. of in-process add DC's
REG(10)	No. of in-process delete DC's
REG(12)	Total in-process DC investment
INVSPS	Total dom in-process DC investment
NMINPS	Dom no. of inprocess add DC's
NMBDLS	Dom no. of in-process delete DC's
NDCIN	Dom no. of in-process DC's added this quarter

Figure 69.--Facility Location Algorithm Variables



Routine to schedule RDC additions or deletions

1. Check domestic constraints to see if RDC additions or deletions can be done this quarter

2. Check to see if RDC's are to be added on a fixed basis

3. Check to see if RDC is to be deleted on fixed basis

Routine to select region(s) for possible changes

4. Check regional constraints to see if RDC changes can be made

5. Calculate the expected regional SMF for locate routine

6. Is regional SMF outside limits for either addition or deletion?

7. Add region to list with its calculated, expected SMF plus absolute deviation from limit

8. Loop through all in-solution regions

9. Check to see if any regions were put in list for possible changes

Process list of regions

10. Select region with greatest absolute deviation from upper or lower limit

11. Build list of potential DC's in region to be examined

12. See if in-solution RDC's are to be possibly deleted

Attempt to add a RDC in region

13. Examine list of eligible RDC's for possible addition

14. Does the best RDC fit the criteria for addition?

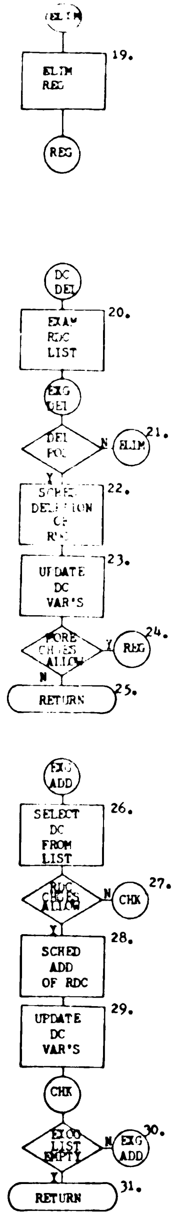
15. Schedule addition of RDC

16. Update domestic and regional variables for RDC in-process of addition

17. Check domestic constraints to see if more additions can be scheduled

Connector for return

18. Return to M&C QUAR event activity



Best RDC for region did not fit criteria

19. Eliminate region from consideration

Go back to routine to select regions for possible changes

Attempt to delete a RDC from list

20. Examine list of eligible in-solution RDC's for possible deletion

Routine for possible deletion of specific RDC

21. Does the selected RDC fit the criterion for deletion?

22. Schedule deletion of RDC

23. Update domestic and regional variables for RDC in process of deletion

24. Check domestic constraints to see if more deletions can be made

25. Return to M&C QUAR event activity

Routine to schedule exogenously RDC's into process of being added

26. Select next DC from list

27. Check domestic and regional constraints to see if more additions can be scheduled

28. Schedule addition of RDC

29. Update domestic and regional variables for RDC in process of addition

Process all DC's in exogenous list

30. Check to see if list is empty

31. Return to M&C QUAR event activity

Figure 70.--Facility Location Algorithm

Examining the routine in detail, Activity Block 1 checked the domestic constraints to see if any changes in the DC configuration could take place. By setting the data base variable NATCON off, no DC system change was allowed and the whole routine was skipped. If NATCON was "on," then the various in-solution and in-process values and constraints were checked to see if changes could take place. The total PD system variables that were checked were MAXDCS, NMINPS and NUMDCS. The checked in-process variables were INVSTS and MXINVS, INVSPS and MXIVPS, NMINPS, NMBDLS and MXIPAS and NDCADD and NDCIN. These variables were checked in a basic FORTRAN subprogram and, if any of the constraints were not satisfied, then LOCATE was skipped.

If DC changes were allowed domestically, then the next two activity blocks checked data base variable IUDF(1) to see if any exogenous DC changes had been specified. If IUDF(1) was positive, then the DC's packed in IUDF(1) were to be set into process that quarter as long as the constraints for addition were not violated. If IUDF(1) was negative, then the absolute value of IUDF(1) was the DC to be set in the process of deletion. This method of deletion only allowed one possible DC to be deleted per quarter. Both of these activities were associated with what was determined as the addition or deletion of DC's on a fixed basis instead of letting the basic algorithm add or delete DC's on a so-called free basis. Also in any one quarter only exogenous or endogenous changes were allowed and not both.

If no exogenous changes were desired, then LOCATE began the series of steps to determine if DC's were to be added or deleted on an endogenous or free basis.

Activity Blocks 4 through 11 are the activities used to determine which of the market regions was in the most critical need of change. Block 4 checked the regional constraints on DC changes to see if anything could be done in this region. Here again the basic FORTRAN subprogram that checked the domestic constraints was called to check the constraints for the region being processed, IREGNO. Data base variables REGCON, REG(1), REG(9) and REG(17), REG(2), REG(3), REG(9) and REG(10), REG(6) and REG(11), and, finally, REG(7) and REG(12) were checked against each other. If any constraints were violated, the region was skipped for consideration.

Given the region could be considered, then Activity Block 5 calculated an expected, regional sales modification factor (SMF) similar to that used by the DC's to modify forecasted sales for the quality of service being given to its customers. This SMF, RLCSMF, was different from that of the DC's in that it was calculated not only on the basis of actual service given in the last quarter but also the service that might be given by the present DC's in the process of being added. Data base variables

REG(9), REG(17) and REG(14) were used in the formula for calculating RLCSMF.

After calculating the region's expected SMF, it was compared to limits that were set internal to this routine. If RLCSMF was greater than the lower limit and less than the upper limit, then there was no action necessary in this region. If RLCSMF was less than the lower limit, then this region would possibly be considered for a DC addition. If RLCSMF was greater than the upper limit, then this region would be considered for a possible DC deletion.

Activity Block 7 added this region to the list of regions, RLIST, to be considered plus the deviation from the limit of which it was outside. Block 8 indicates that all in-solution regions, NREGS, were checked. Activity Block 9 checked to see if any regions were put in the list, RLIST. If no regions were put in the list, LOCATE returned back to the M&C QUAR event activity.

Given there were region(s) in the list, then Activity Block 10 selected the region, IREGNO, with the greatest deviation from its respective limit. A region was now selected for a possible DC change. In addition, IREGNO was set negative if a deletion or set positive if an addition was possible. Activity Blocks 5 to 10 were performed in a basic FORTRAN subprogram that was called from LOCATE.

Activity Block 11 identifies the activities associated with building a new list, RLIST, of all the potential

DC's for the region to be examined in detail. Data base variables DCP(8), IDCNO1 and the total number of potential DC's were checked to see if they should be added to the region's list.

Activity Block 12 then checked IREGNO to see if it was positive or negative. If negative, LOCATE went to a subprogram that contained EXTENDED FORTRAN instructions to see if a DC could be deleted. If IREGNO was positive, then LOCATE went to a subprogram that also contained EXTENDED FORTRAN instructions to see if a DC could be added.

Given that IREGNO was positive, then Activity Blocks 13 through 16 are concerned with the possible addition of a DC. Activity Block 13 looped through the list of potential DC's, RLIST, eliminating first the DC's that were already in solution, were in solution and later deleted, or were in the process of addition. Data base variable DCP(12) was checked in this first step. The second step accumulated the exponentially averaged sales dollars, DU(9), for the DU's with which each of the eligible or remaining DC's best served. Data base variables NDUS, IDUNO and DCASGN were used in this process. After all the DU's had been processed, the DC with the lowest accumulation of sales dollars, DCSUM, was eliminated from the eligible list. This elimination continued until two DC's were left. Then the DC with the higher sales was selected as a possible candidate for addition.

Activity Block 14 checked the accumulated sales, DCSUM, for the possible candidate to see if it was greater than the capacity constraints for the smallest size DC, DCCAPC. If it was greater, then Activity Block 15 scheduled the DC into the process of addition. Data base variable DCP(12) was set for the new in-process DC using variables TNOW and TMINPR.

Activity Block 16 updated the various DC in-process variables, NMINPS and REG(9) and INVSPS and REG(12) using data base variables INVEST, DCSUM, DCCAPC, IDCN01 and IREGNO. These last two activity blocks were performed in a FORTRAN subprogram called from LOCATE.

After the DC had been scheduled into process and the necessary variables updated, then Activity Block 17 represents the activities of again checking the national constraints, checked in Activity Block 1, to see if more DC's could be added. If so, the whole process, Activity Blocks 4 through 16, was repeated. If not, then LOCATE returned back to the M&C QUAR activity.

Block 19 is associated with the activity of eliminating a region from consideration. If after checking all of its eligible DC's and none of them were large enough in potential sales volume, then it was eliminated. REGCON was set off for this region.

Activity Blocks 20 to 25 are associated with those activities of possibly scheduling into process the deletion of a DC. These blocks were a series of FORTRAN subprograms that were used for both the endogenous and exogenous possible deletions of DC's.

Activity Block 20 is strictly associated with the possible endogenous deletion of a DC. This block used data base variables IDCNO1, IREGNO, DCP(8) and RLIST to determine the eligible DC's that could possibly be deleted. In building the eligible list only RDC's presently in solution and not being deleted were considered.

After a preliminary list of eligible DC's was built, their last quarter's cost per pound of sales was calculated. Data base variables IDCNO, DCIS(6) and DCIS(17) were used to calculate this cost measure which was stored in RLIST. The DC with the highest cost measure was then selected for possible deletion.

Activity Block 21 then checked DCP(12) to see if the DC had been in solution for at least two years. If not, then this region was eliminated from consideration. If the DC had been in solution long enough, then Activity Block 22 set data base variable DCP(12) equal to the negative sum of data base variables TNOW plus TMINPR.

Activity Block 23 called a basic FORTRAN subprogram that updated the necessary in-process DC variables NMBDLS and REG(10) using IREGNO. After that activity, the data base variables discussed in relation to Activity Block 1 were checked to see if more deletions could be made domestically. If so, return to LOCATE Activity Blocks 4 to 16. Otherwise, LOCATE returned to the M&C QUAR activity.

Activity Blocks 21 through 24 are also used for the exogenous or fixed deletion of a DC. If the process were on a fixed basis, RLIST contained the DC specified in IUDF(1). Also the constraints were set so that only one DC could be scheduled for deletion.

Activity Blocks 26 through 30 are associated with the exogenous scheduling of DC's into the process of being added to the PD system.

Activity Block 26 set data base variable IDCNO1 equal to the next DC to be unpacked from IUDF(1). The procedure was such that up to ten DC's could possibly be added in any one quarter. Therefore, the DC's were drawn from this possible list of ten DC's and added into process as long as the domestic and regional constraints were not violated.

Activity Block 27 checked the constraint variables NATCON, NDCIN and NDCADD to see if they were satisfied. If not, the subprogram branched to Activity Block 30. Otherwise, Activity Blocks 28 and 29 scheduled the DC into process and updated the necessary in-process variables. Data base variable DCP(12) was set with TNOW plus TMINPR to schedule the DC into process. NMINPS, INVSPS, REG(9), REG(12) and NDCIN were updated for the new in-process DC.

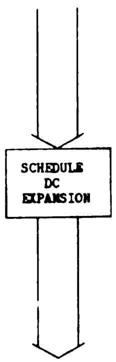
Finally, Activity Block 30 checked to see if the exogenous list in IUDF(1) was empty. If it was empty, LOCATE returned to the M&C QUAR activity. If not, the next DC was selected from the list for possible addition in Activity Blocks 28 and 29.

The last subprogram associated with the CONTROLLER's REVIEW section is called EXPAND. This FORTRAN subprogram basically checked all in-solution DC's to see if they were in need of expansion. If they were, then they were put into the process of expansion to the next size DC. Figure 71 shows this subprogram's data base variables.

Activity Block 1 of Figure 72 checked data base variables NATCON, INVSTS and MXINVS, INVSPS and MXIVPS, NMINPS, NMBDLS and MXIPAS and NMINPS, NUMDCS and MAXDCS to see if any expansions could take place. If not, then the activities in EXPAND were skipped.

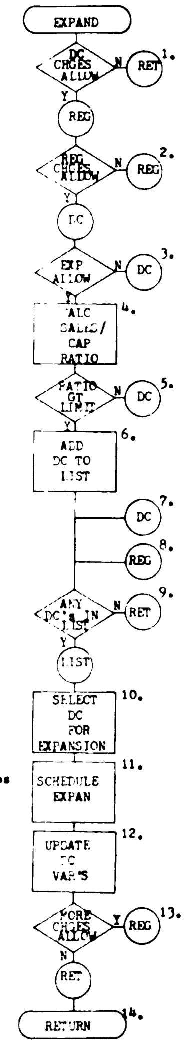
Given that the domestic constraints were not violated, then Activity Block 2 checked the regional constraints for the region being processed, IREGNO. Data base variables REGCON, REG(11) and REG(6), REG(12) and REG(7), and REG(2) and REG(9) were checked.

If expansions could take place in this region, then all in-solution DC's, NUMDCS, were first checked to see if they were in the region being processed and then to see if they needed to be expanded. Activity Block 3 checked data base variables DCP(8) for the correct region and DCP(12) to see if it was not already in the process of being expanded. If either of these conditions were unsatisfied, then the DC, data base variable IDCN01, was eliminated from further processing.



NAME	DESCRIPTION
INPUTS	
INVSTS	Domestic in-solution DC investment
MXINVS	Max dom in-solution DC investment
MXIVPS	Max dom in-process DC investment
MXIPAS	Max dom no. of in-process DC's
INVSPS	Dom in-process DC investment
NATCON	Domestic constraints on-off flag
REGCON	Regional constraints on-off flag
NMINPS	Dom no. of in-process add DC's
NMBDLS	Dom no. of in-process delete DC's
REG(11)	Regional in-solution DC investment
REG(12)	Regional in-process DC investment
REG(6)	Max reg in-solution DC investment
REG(7)	Max reg in-process DC add investment
REG(9)	No. of in-process add DC's--reg
REG(2)	Max no. of in-process add DC's
DCP(12)	Time of DC change + type change
DCIS(2)	Potential DC in-solution
DCIS(8)	DC dollar size indicator
DCIS(16)	DC QTD dollar sales
DCCAPC	DC dollar size capacity
NUMDCS	No. of in-solution DC's
NREGS	No. of in-solution regions
TNOW	GASP day of simulated time
THINPR	Delay time to expand DC
INVEST	Avg inv't by DC size + type
DCP(8)	DC type + PDC assignment
MAXDCS	Maximum DC's in system
KEY ENDOGENOUS VARIABLES	
Local	
SRATIO	DC sales to capacity ratio
RLIST	List of DC's for possible expansion
Common	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
IREGNO	No. of region being processed
OUTPUTS	
DCP(10)	Exp avg quar dollar sales
DCP(12)	Time of DC change + type change
INVSPS	Domestic in-process investment
NMINPS	Domestic no. of in-process DC's
REG(9)	No. of in-process add DC's--reg
REG(12)	Regional in-process investment

Figure 71.--DC Facility Expansion Scheduling Variables



Routine to make possible DC expansions

1. Check domestic constraints to see if DC expansion(s) can be done this quarter

Process each region for possible changes

2. Check regional constraints to see if DC expansion(s) can be done this quarter--no DC's added

Process each DC in region

3. Check to see if DC, in region being processed, is allowed to be expanded

4. Calculate sales to capacity ratio using quarterly data

5. Is the sales to capacity ratio greater than the specified limit?

6. If so, add DC to list of possible expansions

7. Loop through all in-solution DC's

8. Loop through all in-solution regions

9. Check to see if any DC's were put in list of possible expansions

Process list of DC's

10. Select DC with largest ratio

11. Schedule the DC's expansion

12. Update the domestic and regional DC in-process variables

13. Check domestic constraints to see if more expansions can be done?

Connector for return

14. Return to M&C QUAR event activity

Figure 72.--DC Expansion Scheduling Routine

Activity Block 4 calculated a sales dollar to capacity ratio that was later compared to a specified percentage of the present DC maximum size in order to determine if the DC was nearing its upper capacity restriction. Data base variables IDCNO, DCCAPC, DCIS(8) and DCIS(16) were used to calculate the sales to capacity ratio, SRATIO.

This ratio, SRATIO, was compared to the specified percentage of DCCAPC in Activity Block 5. The percentage was built into the routine. If the DC was above its limit, then it was added to a list, RLIST, of possible DC's to be scheduled for expansion. Block 6 added the DC to the list, while Activity Block 7 indicates that all in-solution DC's were checked for possible expansion. Block 8 indicates that all regions, NREGS, were checked before any DC's were actually scheduled for expansion.

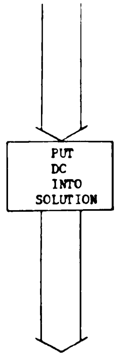
After examining all in-solution DC's for possible expansion, Activity Block 9 was a check to see if any DC's were in RLIST. If not, the rest of the activities in EXPAND were skipped. If DC's were in RLIST, then Activity Block 10 identifies the activities of selecting the DC with the largest sales to capacity ratio. IDCNO1 was set equal to this DC and Activity Block 11 then scheduled this DC into the process of expansion. Data base variables IDCNO1, DCIS(2), TNOW, TMINPR and a constant of nine million were used to set DCP(12) for the time of expansion. Nine million plus the time of expansion was used to differentiate a DC being expanded from a DC that was being added into or deleted from solution.

Activity Block 12 then updated the domestic and regional in-process data base variables INVSPS, NMINPS, REG(12), REG(9) using data base variables INVEST, DCIS(8), DCP(8), IDCNO, IDCNO1 and IREGNO. The difference between the investments for the DC's present and scheduled sizes was added to the in-process investment variables.

Activity Block 13 is similar to the first activity block of EXPAND in which the domestic constraints were checked to see if any expansions could take place. If more expansions were possible, then the program looped back to build RLIST again and possibly expand another DC. If no more expansions were allowed, then Activity Block 14 indicates a return to the M&C QUAR event activity.

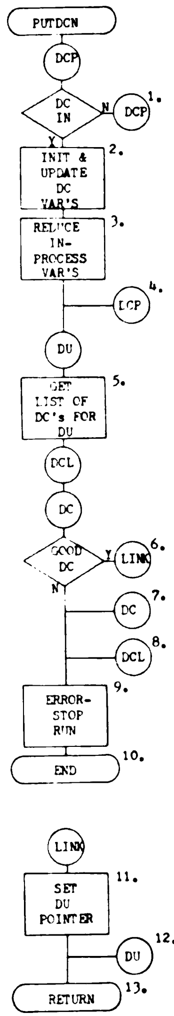
Update.--The next set of routines are in the UPDATE section of M&C's CONTROLLER. These routines activated any scheduled changes to the PD system which was to be tested over the next simulated quarter of activity.

The first subprogram centers on the activities that were involved in putting DC's into solution. Various in-process and in-solution DC variables were updated plus the new DC's status variables were initialized for the coming quarter of activity. Figure 73 shows the data base variables that were involved in this DC update procedure while Figure 74 identifies the logical activities.



NAME	DESCRIPTION
INPUTS	
TNOW	GASP day of simulated time
NDUS	Total no. of DU's
DCASGN	List of feasible DC's to DU's
ITNSP	Total no. of tracked products
DCCAPC	DC size capacities
DCP(6)	DC type + PDC assignment
DCP(10)	DC exp avg dollar sales
DCP(12)	Time of DC change + type change
INVEST	Avg inv't by DC size + type
REG(14)	Regional exp avg SMF
KEY ENDOGENOUS VARIABLES	
Common	
IDUNO	No. of DU being processed
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
IREGNO	No. of region being processed
OUTPUTS	
NUMDCS	No. of in-solution DC's
DCIS(1) to DCIS(21)	In-solution DC variables
DU(8)	In-solution DC pointer
REG(9)	Reg no. of in-process DC's--add
REG(11)	Regional in-solution investment
REG(12)	Regional in-process investment
REG(17)	Reg no. of in-solution DC's
NMINPS	Dom no. of in-process DC's
INVSTS	Domestic in-solution investment
INVSPS	Domestic in-process investment
PRDC(2)	DC tracked product S-level
OCTDIS(1)	NOCT sales dollar proportions
PDSCHG	PD system change indicator

Figure 73.--Put DC Into Solution Variables



Routine to add a DC into solution

Check each potential DC location to see if it was to be added

Was this DC to come into-solution now?

Initialize and update in-solution DC variables for incoming DC

Reduce number of DC's in process of being added--regionally and domestically

Loop through all potential DC locations

Process each DU

Get ranked list of feasible DC's that can serve this DU

Process each DC in DU's list until in-solution DC found

Process each in-solution DC

Check to see if this in-solution DC matches DU's potential DC

Loop through all in-solution DC's

Loop through all potential DC's in DU's list

Print error indicating DU could not be linked to in-solution DC & stop run

End of computer program execution

Routine to link DU to in-solution DC

Set DU pointer for row of in-solution DC array serving it

Loop through all DU's

Return to M&C QUAR event activity

Figure 74.--Put DC Into Solution Activities

Activity Block 1 checked data base variables DCP(12) with TNOW for each potential DC, IDCN01, to see if it was to come into solution for the coming quarter. If it was to come into solution, then Activity Block 2 updated data base variables NUMDCS, REG(17), INVSTS, REG(11) using data base variables DCP(8), IDCN01, IREGNO, DCIS(8), IDCNO, DCCAPC, INVEST and DCP(10). Block 2 also initialized data base variables DCIS(1) through DCIS(21). Specifically, DCIS(4) was set equal to the exponentially averaged regional sales modification factor, REG(14). DCIS(2) was set equal to IDCN01, the row of the potential DC array, DCP. DCIS(8) was set equal to the DC's expected size, while the rest of the DCIS variables were set to zero. Also initialized were data base variables OCTDIS(1), set equal to one, and PRDC(2), which was set equal to a negative one. Because PRDC(2) was so initialized, the INVMGT routine would then calculate the DC's initial inventory control variables. Data base variable PDSCHG was also set so that additional DC to DU information could be generated.

Activity Block 3 then reduced the necessary in-process variables for the domestic and regional areas of consideration. Data base variables NMINPS, INVSPS, REG(12) and REG(9) were reduced. Block 4 indicates that each of the potential DC's, a maximum of thirty-five, were processed.

The remaining activity blocks are designed to link the DU's to the best in-solution DC that could serve it.

The basis for picking the best DC was the heuristically selected list of potential DC's, DCASGN, that could serve each of the DU's.

Activity Block 5 obtained the ranked list of feasible DC's that could serve the DU being processed, IDUNO1. Each of the DC's in the DU's list, starting at the best and going to the worst DC, was checked against the list of in-solution DC's until a match occurred. This checking was done in Activity Block 6. The matched DC, IDCNO, was then given this DU for servicing. Activity Block 11 set data base variable DU(8) with IDCNO which established the row of the in-solution DC array, DCIS, to which this DU was linked. Earlier it was noted that DCIS(2) contained the row of the potential DC array, DCP, to which the row of the in-solution DC array, DCIS, was linked. Activity Blocks 7 and 8 indicate that all in-solution DC's, NUMDCS, were checked against the DC's in the DU's list until a match occurred. Block 9 was an error condition in which no in-solution DC matched the DU's list. The last DC in the list was always supposed to be a PDC which could never be deleted from solution.

Activity Block 12 indicates that each DU was linked to its best DC and then the routine returned to the M&C QUAR event activity.

The next major component of UPDATE is the deletion of DC's from the PD system configuration. The LREPS data base variables used in this FORTRAN subprogram are listed in Figure 75. The subprogram's logical activities are shown in Figure 76.

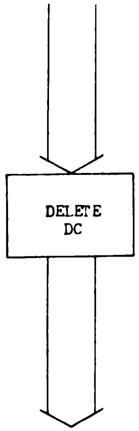
Activity Block 1 checked data base variable DCP(12) to see if it was negative. If it was negative, then the absolute value of DCP(12) was checked against TNOW to see if it was time for the DC to be deleted. Data base variables, DCIS(2), IDCNO and IDCNO1 were also used in this checking procedure.

If a DC was to be deleted, then Activity Blocks 2 through 9 represent the activities that were necessary to delete the DC.

Activity Block 2 first reduced the in-solution DC variables NUMDCS, REG(17) and INVSTS and REG(11) using data base variables INVEST, DCP(8), IREGNO, DCIS(8), IDCNO and IDCNO1. This block then set PDSCHG so that new DC to DU link information would be generated. Activity Block 3 then reduced the in-process variables NMBDLS and REG(10) using IREGNO.

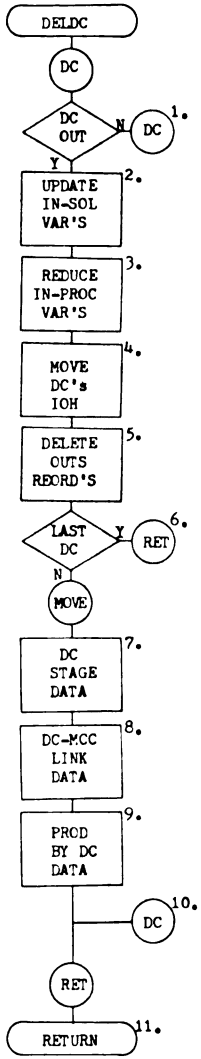
Activity Block 4 shifted the deleted DC's remaining inventories-on-hand and any outstanding stockout commitments, PRDC(4), back to the DC's assigned PDC. All tracked products, ITNSP, were processed for the deleted DC, IDCNO.

In Activity Block 5, the GASP subprograms RMOVE and FINDN were used to delete any outstanding DC to MCC dispatched orders plus any MCC to DC dispatched shipments for the DC being deleted. These outstanding events were destroyed and no further processing was done with them.



NAME	DESCRIPTION
INPUTS	
TNOW	GASP day of simulated time
DCP(8)	DC type + PDC assignment
DCP(12)	Time of DC change + type change
INVEST	Avg inv't by DC size + type
ITNSP	Total no. of tracked products
NMCCS	No. of in-solution MCC's
NREGS	No. of in-solution regions
KEY ENDOGENOUS VARIABLES	
Common	
IDCNO	No. of in-solution DC being processed
IDCNO1	Potential DC being processed
IREGNO	No. of region being processed
OUTPUTS	
NUMDCS	No. of in-solution DC's
DCIS(1) to DCIS(21)	In-solution DC variables
REG(10)	Reg no. of in-process delete DC's
REG(11)	Reg total in-solution investment
REG(17)	Reg no. of in-solution DC's
NMBDLS	Domestic no. of in-process delete DC's
INVSTS	Domestic in-solution investment
DCMCC(1) to DCMCC(5)	DC-MCC link variables
PRDC(1) to PRDC(6)	DC tracked product variables
PDSCHG	PD system change indicator

Figure 75.--Delete DC From Solution Variables



Routine to delete a DC from solution

Check each in-solution RDC

1. Check to see if DC to go out

2. Update in-solution variables for deletion of DC

3. Reduce in-process variables for actual deletion

4. Move deleted DC's inventory-on-hand(IOH) to its assigned PDC's IOH

5. Delete any outstanding DC-MCC reorders for deleted DC

6. Check to see if it is last in-solution DC being processed

Move last in-solution DC's data to deleted DC's row

7. Move in-solution DC stage data

8. Move DC-MCC link data--change DC code in variable time events

9. Move tracked product information by DC data

10. Loop through all in-solution RDC's

Return connector

11. Return to M&C QUAR event activity

Figure 76.--Delete DC From Solution

The next series of activity blocks has to do with the order in which in-solution DC's were retained in the DCIS array. The DCIS array had been designed to allow a certain maximum number of DC's to be in solution at any one time. If the maximum number was not in solution, then the DC's that were in solution were put in the first rows of the array so that the skipping of unoccupied rows would not have to be done. Therefore, if a DC was deleted from an inner row, then the last DC was shifted up to take the deleted DC's row.

Activity Block 6 checked to see if it just happened that the DC in the last occupied row, NUMDCS plus one, was deleted. If so, then no shifting was necessary.

If the deleted DC was not in the last occupied row, then Activity Block 7 moved the DCIS information using data base variables IDCNO, the DC being deleted, and, NUMDCS plus one, which identified the last occupied row.

Activity Block 8 moved the information for the DC-MCC links to the new row plus the GASP subroutine FINDN was used to locate the outstanding events for the DC, NUMDCS plus one, being moved. All the events for the moved DC were given its new in-solution DC code, IDCNO. In Activity Block 9 the tracked product information for the DC was moved. The tracked product information was in data base variables PRDC(1) to PRDC(6) and had to be moved for all tracked products, ITNSP.

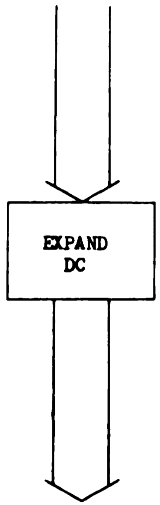
Finally, Activity Block 10 indicates that all in-solution RDC's were checked for possible deletion.

No PDC's were checked since they could not be deleted. Block 11 then shows a return to the M&C QUAR event activity.

When it came time for a DC to be expanded to a larger size, subprogram MODIFY performed the necessary activities for the expansion. Figure 77 lists the data base variables involved in the expansion and Figure 78 shows the logical activities. Activity Block 1 of Figure 78 checked data base variable DCP(12) for the possible expansion. If DCP(12) less nine million was equal to TNOW, then it was time for the expansion. Data base variables IDCNO, DCIS(2) and IDCNO1 were also used in this checking process.

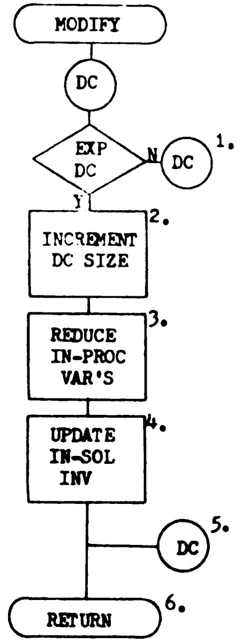
If the DC was to be expanded, then Activity Block 2 incremented DCIS(8), the DC size indicator, by one. Activity Block 3 reduced the in-process variables NMINPS, INVSPS, REG(9) and REG(12). Data base variables INVEST, DCP(8), IDCNO1, IDCNO, DCIS(8) were used in this reduction process. As in the subprogram that scheduled the DC expansion, only the difference between the investment figures for the old and new DC sizes was used as the in-process investment.

This difference was then added to the in-solution variables INVSTS and REG(11) in Activity Block 4. Activity Block 5 indicates that all in-solution DC's were checked for possible expansion. After checking all DC's, the routine returned to the M&C QUAR event activity.



NAME	DESCRIPTION
INPUTS	
TNOW	GASP day of simulated time
DCP(8)	DC type + PDC assignment
DCP(12)	Time of DC change + type change
INVEST	Avg inv't by DC size + type
NUMDCS	No. of in-solution DC's
KEY ENDOGENOUS VARIABLES	
Common	
IDCNO	In-solution DC being processed
IDCNO1	Potential DC being processed
IREGNO	No. of region being processed
OUTPUTS	
DCIS(8)	DC dollar size indicator
NMINPS	Domestic no. of in-process add DC's
INVSTS	Domestic in-solution DC investment
INVSPS	Domestic in-process DC investment
REG(11)	Regional in-solution DC investment
REG(12)	Regional in-process DC investment
REG(9)	Reg no. of in-process add DC's

Figure 77.--Expand In-Solution DC Capacity Variables



Routine to expand size of in-solution DC's

Process each in-solution DC

Check to see if DC to be expanded

Increment DC size indicator by one

Reduce regional and domestic in-process variables

Update regional and domestic in-solution investments

Loop through all in-solution DC's

Return to M&C QUAR event activity

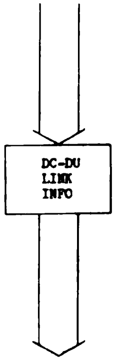
Figure 78.--Expand DC Size Routine

The last subprogram in the UPDATE section of the M&C's CONTROLLER is concerned with the development of DC to DU link information needed for operating purposes. Figure 79 lists the data base variables used in this routine while Figure 80 shows the logical activities needed to generate the required information.

Activity Block 1 checked data base variable PDSCHG for any PD system changes. If there had been changes, then Activity Block 2 used a basic FORTRAN subprogram to sort the DU's in-solution DC assignments, DU(8), into ascending order. All DU's, NDUS, were sorted with the original row for the DC's DU's stored in DU(6). The result of this sorting process was that all DU's for each in-solution DC were now grouped together. The reason for sorting the DU's on their DC assignment was to develop the basis for the Monte Carlo selection of a DC's DU's in the D&E sales processing routine.

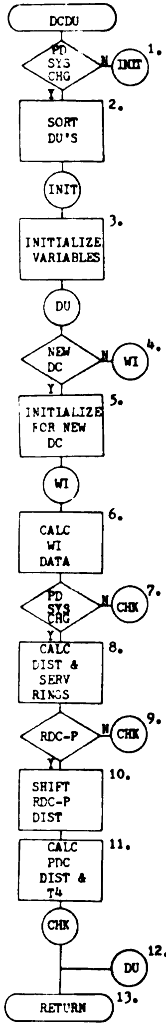
After grouping the DU's by DC, Activity Block 5 initialized data base variables WIACC, DCIS(3) and DCIS(5) for the first in-solution DC, IDCNO. Data base variables DU(8) and IDCNO were then checked for a change of DC in Activity Block 4. If there was a change, then data base variables WIACC, DCIS(3) and DCIS(5) were again set to zero for the new DC to be processed.

Activity Block 6 accumulated the cumulative weighted index by DU, DC and region using exogenous data base variable DU(4) which was changed annually. The accumulation,



NAME	DESCRIPTION
INPUTS	
PDSCHG	PD system change indicator
NDUS	No. of DU's being processed
NUMDCS	No. of in-solution DC's
DU(4)	DU weighted index
IDUNO1	DU selected for processing
DU(2)	DU x-coordinate
DU(3)	DU y-coordinate
DCIS(2)	In-solution DC pointer
DCP(1)	Potential DC x-coordinate
DCP(2)	Potential DC y-coordinate
DCP(9)	Mileage ring set indicators for elements T1 + T4
DCP(8)	DC type + PDC assignment
RING	Alt cust mileage ring sets
KEY ENDOGENOUS VARIABLES	
Local	
WIACC	Accumulated weighted indices
Common	
IDCNO1	Potential DC being processed
IDCNO	In-solution DC being processed
OUTPUTS	
DU(5)	DU service ring numbers
DU(6)	Sorted DU code
DU(7)	Cumulative weighted index
DU(8)	In-solution DC pointer
DU(11)	Road distance from full-line DC
DU(12)	Road distance from RDC-P
DCIS(3)	No. of DU's per DC
DCIS(5)	Sum of DU weighted indices
REG(8)	Regional cumulative weighted index

Figure 79.--DC to DU Link Information



Routine to generate DC to DU link information

1. Check to see if any PD system changes have occurred for coming quarter

2. Sort DU's in order to group them by DC

Beginning of routine to generate DC-DU info

3. Initialise variables used to accumulate DU & DC data

Process each DU

4. Check to see if all DU's for present DC processed

5. Change of DC, initialise variables for accumulating information for new DC

Generate weighted index information

6. Calculate DU, DC and regional weighted index data

7. Check to see if PD system change

8. If PD system change, calculate road distance & service rings (T1 & T4) for majorly assigned DC to DU

9. Check to see if majorly assigned DC was RDC-P

10. Move majorly assigned DC distance to RDC-P distance variable

11. Calculate distance and service ring number T4 for PDC to DU

Connector to bypass calculation of service rings and distances

12. Loop through all DU's

13. Return to MAC QUAR event activity

Figure 80.--DC to DU Link Information Generation

WIACC, was stored in DU(7) for the DU being processed, IDUNO1. The DC and regional accumulations were updated in data base variables DCIS(5) and REG(8). Also data base variable DCIS(3) was incremented by one for another assigned DU.

Activity Block 7 again checked PDSCHG to see if it was set on. If PDSCHG was on, then Activity Block 8 calculated the road distance from the majorly assigned DC, IDCNO, to the DU being processed. This road distance was calculated using the same basic FORTRAN subprogram used in the D&E "Distance Check" program. Data base variables DU(2), DU(3), DCP(1) and DCP(2) were used to calculate the road distance which was stored in DU(11).

After calculating the road distance, the constant or average service time elements T1 and T4 from the DC to the DU were determined using data base variables DCP(9), DCIS(2), IDCNO1 and RING. This activity used EXTENDED FORTRAN instructions to shift out the ring set indicator from DCP(9). Using the ring set numbers and the calculated road distance, the constant service time elements were determined and packed into data base variable DU(5).

Activity Block 9 checked to see if the majorly assigned DC was a RDC-P. If it was a RDC-P, then Activity Block 10 shifted the distance calculated in Block 8 from DU(11) to DU(12). Then Block 11 calculated the road distance from the RDC-P's assigned PDC to the DU being processed. Again data base variables DU(11) and DU(5) were set using variables DCP(8), IDCNO1, DU(2), DU(3), DCP(1), DCP(2), DCP(9) and RING. Only service time element T4 from the PDC to the DU was determined since the order was always assumed to be transmitted to the RDC-P. Therefore, customer service time element T1 was the same for both orders split between the PDC and the RDC-P.

Activity Block 12 indicates that all DU's, NDUS, were processed before the routine returned to the M&C QUAR event activity.

Supporting Data System

The supporting data system for the M&C subsystem includes three computer activities. The first computer activity is the preparation of the LREPS exogenous (EXOG) input tape. This program utilized EXTENDED FORTRAN instructions to process the exogenous information from all four LREPS subsystems. Except for the D&E subsystem's Demand Unit (DU) tape, the inputs into this program were punched cards. A small part of the punched card information was from computer activities in the LREPS supporting data systems. The high majority of this punched card information was from non-computer activities.

As was briefly discussed in the OPERATING SYSTEM of this chapter, the EXOG subprogram in the M&C MONITOR processed a tape file of quarterly exogenous inputs that were cataloged into segments of the LREPS common data base.

The card input into the tape preparation run was, therefore, organized to contain the following information:

1) A variable designating the quarter in which the catalog of information was to be read in;
2) Identifying information as to which supporting data system activity prepared the data and in which simulation cycle it was used;
3) The catalog's beginning reference number within the LREPS common data base;
4) The number of additional sequential data base variables read in after the beginning reference number;
5) The mode of the data being read in for the specified catalog of information--integer, real, octal or tape binary (last mode for DU tape);
6) The "data" cards themselves.

The exogenous card file was kept in sequence based upon a sort key composed of the quarter number, the supporting system identification number, the card type (a control card--card type one--telling the number of catalogs per quarter, an individual catalog card--card type two, or a data card--card type three) and finally a sequential card number within card type. The data cards were read under common formats for each mode of data.

The program converted the input data to the "EXOG" magnetic tape in quarterly files. The exogenous information was also printed if the cycle number was different from the base-run cycle number. This last statement refers to the use of a base run--simulation cycle number one--of exogenous data inputs that were needed in order to operate LREPS. Simulation cycles after the base run were given new cycle numbers and only the exogenous data that was peculiar to that cycle was printed. The base cycle data was, therefore, only printed for the base run.

No block diagram is given for this program since it essentially merged a card deck of information into the DU tape to prepare the EXOG tape. In essence the program served two purposes. The first purpose was to prepare the EXOG tape. The other purpose was to provide a meaningful way for organizing the LREPS operating system data inputs for easy reference and cycle documentation. At any time the base run could be created by collecting and processing the cycle one data.

The second computer activity in the M&C supporting data system is an EXTENDED FORTRAN activity for converting a card deck of information that was in standard binary coded decimal form into an octal representation. The BCD deck contained one card for each DU. The information in each card was composed of the DU identifying code plus the codes of all potential DC's that could feasibly serve the DU. The potential DC codes were ranked in decreasing order of priority for serving the DU. The last potential DC code for a DU was that of a PDC since a PDC could never be deleted from the physical distribution system configuration.

The program converted the BCD deck to an octal deck. The newly punched deck also contained identifying information used in conjunction with the EXOG tape preparation run. The first set of punched cards contained packed computer words that contained the number of DC's that could feasibly serve each of the DU's. The second set of packed computer words contained the list of DC's for each DU. This packed computer word deck was stored in the LREPS data base array, DCASGN.

The last M&C computer activity will also be discussed without a block diagram of activities. This basic FORTRAN program converted the potential DC's latitudes and longitudes to rectangular X and Y coordinates. These coordinates were read into the LREPS common data base variables DCP(1) and DCP(2) through the EXOG tape preparation run. The same routines that were used in the D&E "Distance Check" program were used in the conversion process.

CHAPTER VI--FOOTNOTES

[1] Kiviat and Pritsker, pp. 1-21.

CHAPTER VII

THE REPORT GENERATOR SYSTEM

Report Generator Inputs

The inputs into the Report Generator (RPG) system included two basic classes of information. The first class of information was the LREPS RPG tape output generated directly from the operating system. The other class of information was the supplementary or descriptive information needed to clarify items in the managerial reports. This supplementary information contained such items as the city and state names of potential DC and MCC locations plus tracked product names.

As was briefly discussed in Chapter VI, the first class of information contained three files of information for each quarter of simulated activity. The three files of information were a quarterly control record, a file of in-solution DC data plus a complete dump of the LREPS common data base.

In addition, the RPG tape was designed to hold more than one simulation cycle of information. This feature enabled reports to be developed based on multi-cycle analysis. However, no multi-cycle reports were prepared on the computer. The present computer programs analyzed the quarterly files of information for a single simulation cycle.

The basic philosophy in developing the RPG tape plus supplementary information was to provide a complete data base of raw or low-level information with which to analyze the simulation results. The desire was not to rerun the operating system to do additional output analysis but only to develop additional computer programs that would analyze the stored results of the operating system. This output analysis took much less time using separate computer programs instead of rewriting the output section of the LREPS operating system and then rerunning it.

The RPG system also allowed the flexibility of developing individual reports for different users of the model without rewriting programs in the LREPS operating system.

Finally, the RPG system allowed us to execute the total LREPS system on a faster basis than if the operating system were bound or slowed up by a printer.

DC-Based Reports

The first set of RPG reports presented the results that were summarized and accumulated for each DC that was in solution during a simulation cycle. The reports were prepared in a book form.

The philosophy in developing these reports was to provide the basis for later report sophistication plus present the information going from highly summarized, aggregative data to detailed data concerning each DC that was ever in solution in a simulation cycle. The information was also printed in such a way as not to overload the manager with too much information on any one page.

The book of information was prepared on 8-1/2 by 11 inch paper which contained:

1) A cover page containing descriptive information about the simulator, the date it was prepared, the simulation cycle number plus alphabetic information describing the purpose of the cycle;
2) A table of contents outlining the following information in the report;
3) Following the table of contents, a set of reports that very broadly summarized quarterly activities--this set of reports contained one page for each quarter of simulated activity and each page included information about the in-solution DC's sales, total distribution cost, sales less distribution cost, average total order cycle time plus its average cubic inventory on hand during the quarter;

4) A cycle summary of quarterly information which was accumulated as the quarterly summaries had been printed and which presented the DC's that had ever been in solution during the cycle plus their total sales, distribution costs, sales less distribution costs, their weighted average total order cycle time based on orders plus the simple average of their average cubic inventory on hand;

5) After the cycle summary was printed, a series of DC detail reports--these reports contained individual, quarterly information categorized as follows for each DC:

   a) Total sales in dollars, weight, cube, cases, lines and orders--in reference to the LREPS data base, these variables were DCIS(16) to DCIS(21);

   b) Total cost for DC to DU outbound transportation, MCC to DC inbound transportation, DC throughput, DU, DC and MCC communications, DC fixed facility allocation, DC inventory carrying and handling, and all costs in total--LREPS data base variables used in printing the report are DCCOST(1) to DCCOST(6) plus DCIS(6);

c) Customer service time statistics which included the mean and standard deviation of both the normal order cycle time and the outbound transportation time, the inventory stockout penalty time, T3, plus the average total order cycle time--LREPS data base variables used in printing the report are DCIS(9) to DCIS(12), DCIS(1) and DCIS(7);

d) DC information to identify certain aspects of the DC--this information included DCIS(3), the number of demand units that this DC served, DCIS(4), the sales forecast modification factor, DCIS(5), the sum of the assigned demand units' weighted indices, and finally, DCIS(8), the facility size indicator;

e) DC information describing the condition of the DC's product inventories over the last quarter and which referred to LREPS data base variables DCIS(13), the percentage of the tracked products' total case unit sales that were back-ordered, DCIS(14) and DCIS(15), the mean and standard deviation of the stockout delay in getting a tracked product to a customer given that it stockout out, PRDC(3), the extrapolated, average total cubic inventory-on-hand for all tracked products, PRCTDC(1) and PRCTDC(2), the extrapolated, total number of product stockouts and single-product reorders against the MCC;

f) DC-MCC reorder information concerning the number of multiple-product reorders, DCMCC(1), and the average reorder lead time, DCMCC(2), from the DC to each of its supplying MCC's;

g) The normal order cycle time, sales dollar proportions, OCTDIS(1), and customer order proportions, OCTDIS(2), for the DU's served from this DC (A report was printed for each set of proportions);

h) The accumulations, WTACUM, of the amount of weight shipped in the various DC weight categories, DCWB, to the DU's;

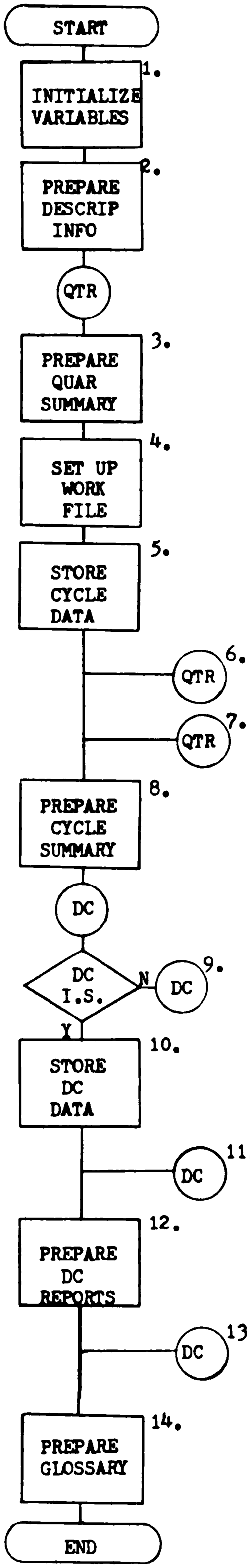
i) The accumulations, DCMCC1, of the amount of weight shipped from the DC's supplying MCC's in the various shipment weight categories, MCCWB (A report was printed for each supplying MCC and the defined weight categories).

Appendix 2 gives a sample printout of the basic report formats for the entire DC-based reports.

Given the above output, Figure 81 shows the logical activities associated with the preparation of these different reports that were printed in an assembly and EXTENDED FORTRAN program. Activity Block 1 indentifies the initialization process of reading card information concerning (1) the potential DC and MCC city names, (2) the report's cover page and (3) the report's table of contents. Activity Block 1 also read the control record for the first quarter from the RPG tape for cycle identification information. Block 2 then printed the report's cover page and table of contents.

Activity Blocks 3 to 7 then describe the activities of preparing the quarterly summary reports for all in-solution DC's, of the building of a cyclic work file of each DC's detailed quarterly information and of the accumulation of the DC's summarized, cyclic data. Associated with the above activities, only the quarterly "DC" files of information in the LREPS RPG tape were processed. The other RPG quarterly control and data base files were skipped.

After the quarterly summary reports were completed, Activity Block 8 represents the activities of printing the cycle summary report. The basis for the preparation of this report was simple arithmetic. Thus, this report could be the focal point of the use of more sophisticated and detailed financial and mathematical analysis.



Start processing RPG DC-based reports

Initialize variables, arrays, potential DC names, MCC names, cycle identification information

Prepare initial, descriptive information about report

Process each in-solution DC per quarter

Prepare line of DC information for quarter

Put DC on work file and set indicator that DC was in-solution for quarter

Accumulate cycle summary data

Loop through all in-solution DC's for quarter

Loop through all quarters in RPG tape file

Prepare cycle summary information

Process each in-solution DC

Check to see if DC was ever in-solution during cycle

Read DC's work file and store data for each quarter that it was in-solution

Loop through all quarters that DC was in-solution

Prepare DC detail reports

Loop through all in-solution DC's

Prepare glossary of mnemonics and their interpretations

End of report generation

Figure 81.--RPG DC-Based Reports' Activities

This cycle summary report was printed after the quarterly summaries because of the savings in computer time processing and storage. The page, however, was placed immediately after the table of contents and before the more detailed quarterly summaries.

Activity Blocks 9 to 13 identify the activities of preparing the detailed DC reports. These reports were only prepared for DC's that had ever been in solution during the entire simulation cycle.

Finally, Activity Block 14 represents the activity of printing a glossary of mnemonics and their interpretations that were used in preparing the report and column headings.

LREPS Variable List

This basic FORTRAN activity was used for validating and checking the values of LREPS data base variables at specified quarters of simulated time. The listing or dump of these variables included the catalog position or reference number of the desired variable in the data base, the number of elements that were associated with this variable plus a listing of the data.

The input into this program included:

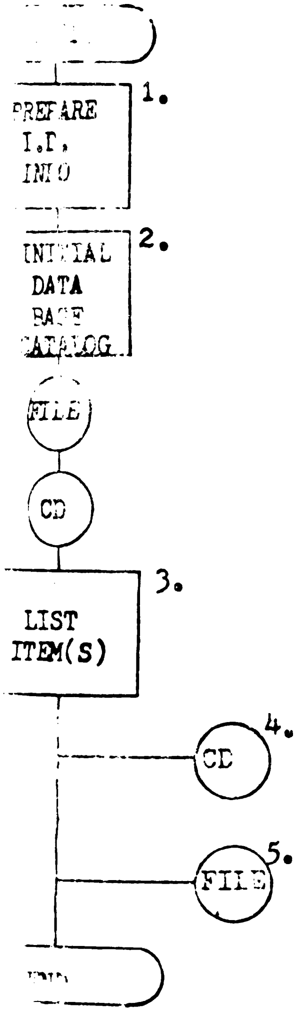
1) A card was read which was used for computer run identification--this card was printed across the top of each page;
2) A deck of cards containing information for the LREPS "cataloged" data base--the

information included each variable's beginning sequential position or catalog reference number in the data base which could be subtracted from the following variable's reference number to determine the total number of elements associated with the variable plus the cataloged variable's mode of data was included in the card deck;

3) The LREPS common data base files on the RPG tape--all other files on the RPG tape for the cycle being checked were skipped;

4) A deck of "instruction" cards that identified the LREPS data base variables that were to be dumped and provided a control digit, called the "file skip control," which indicated whether more variables were to be dumped from the present quarterly LREPS data base file (zero used to indicate this), or indicated whether additional variables were to be dumped in the Xth file succeeding the present file for this cycle.

Figure 82 shows the block diagram of activities to dump the LREPS data base variables. Activity Blocks 1 and 2 initialized the report heading plus the basic data base information needed to dump the variables. Blocks 3,



Start processing variables to be listed

Prepare identifying information as report heading

Initialize information for catalog of data base variables

Process each data base file of information

Process each instruction card

List item(s) using correct mode

Loop through all instruction cards per file

Loop through all data base files

End of analysis

Figure 82.--RPG Variable List Routine

4 and 5 dumped the data from the LREPS data base files according to the deck of instruction cards.

CHAPTER VIII

TOTAL COMPUTER MODEL

Operating System Linkages

Figure 83 illustrates the linkages between the major subprograms of the LREPS operating system and the model's major subsystems. The subprograms are divided into GASP and non-GASP routines. The non-GASP routines represent the user-written routines that were developed by the LREPS project team. These non-GASP routines are further subdivided on the basis of event activity. For each event activity, the lines below the event subprogram block indicate the routines to which it was linked. The linkages between the subprogram blocks flow downward, to the right and to the left. The downward orientation signifies that a routine or routines below a certain subprogram block to which they are connected was called by the upper routine during program execution.

Figure 83 also serves as an illustration of the connection of the major subsystems in the operating system. The two subprograms of the D&E subsystem are identified in

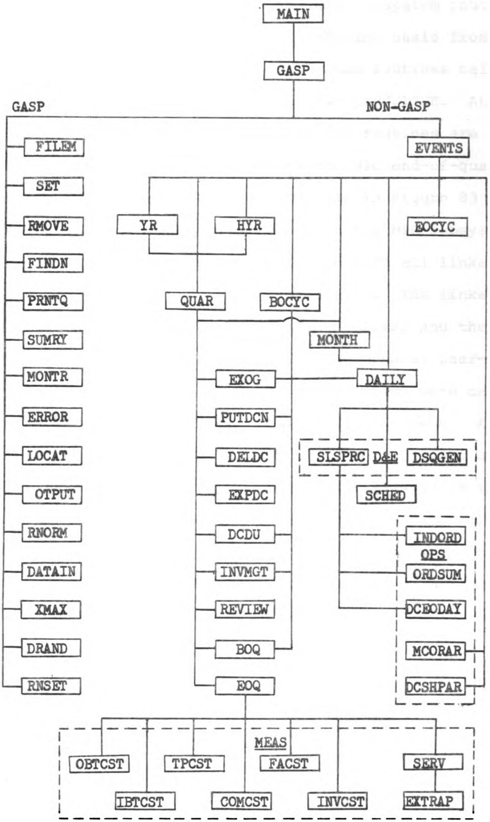


Figure 83.--LREPS Operating System Linkages

the box outlined with bold dashed lines and labeled "D&E." Below the D&E subprograms are the OPS subsystem routines some of which were called on a fixed-time basis from the D&E subprogram SLSPRC and the other two routines called on a variable-time basis from the M&C EXECUTIVE. At the bottom of Figure 83 the MEAS subsystem routines are identified plus their linkages to the M&C end-of-quarter subprogram. The remaining subprograms in Figure 83 are associated with the major sections of the M&C subsystem.

Figure 83 does not entirely identify all linkages among the operating system's subprograms. The linkages between some of the user-written subprograms and the GASP subprograms FILEM, which was called by several user-written routines, and FINDN and RMOVE, which were called by the M&C subprogram DELDC, have not been shown. Also the linkages among the GASP subprograms themselves are not illustrated. The main purpose of Figure 83 is to identify the entire set of operating system subprograms, the GASP and non-GASP routines, the linkages among the user-written subprograms plus the linkages among the LREPS D&E, OPS, MEAS and M&C subsystems.

Activating Alternative LREPS Versions

In developing a computerized operational model of LREPS, the approach was to serially operationalize the entire D&E, OPS, MEAS and M&C LREPS subsystems. As these subsystems were operationalized, they were merged to form the alternative versions of LREPS. After the four subsystems were operationalized and merged to form the first total LREPS computer model, new versions of LREPS included sophisticated versions of existent subprograms.

In order to operationalize a specific version or level of LREPS, an input-output computer system flowchart of the LREPS operating system was needed. Figure 84 shows the flow of information into and out of the LREPS operating system. The inputs included the GASP card deck of GASP run control cards and the exogenously set BOCYC and EOCYC events for the GASP filing arrays NSET and QSET. The remaining inputs were the order file and the exogenous input magnetic tapes. The outputs of the program included the GASP run control listings plus the RPG magnetic tape.

Using the approach for computerizing the model plus the input-output system flowchart illustrated in Figure 84, the objective of the first version of LREPS, LREPS-1, was to operationalize primarily the D&E subsystem. All D&E subsystem supporting data and operating system program routines were programmed. Besides the D&E program routines, the GASP subprograms plus the user-written routines MAIN, EVENTS, BOCYC, EOCYC, YR, HYR, QUAR, MONTH, DAILY, SCHED, a simple PUTDCN routine to initialize the starting DC configuration, a basic DCDU routine to link the DU's to the starting DC's and to generate their cumulative weighted indices, a basic EOQ routine to output

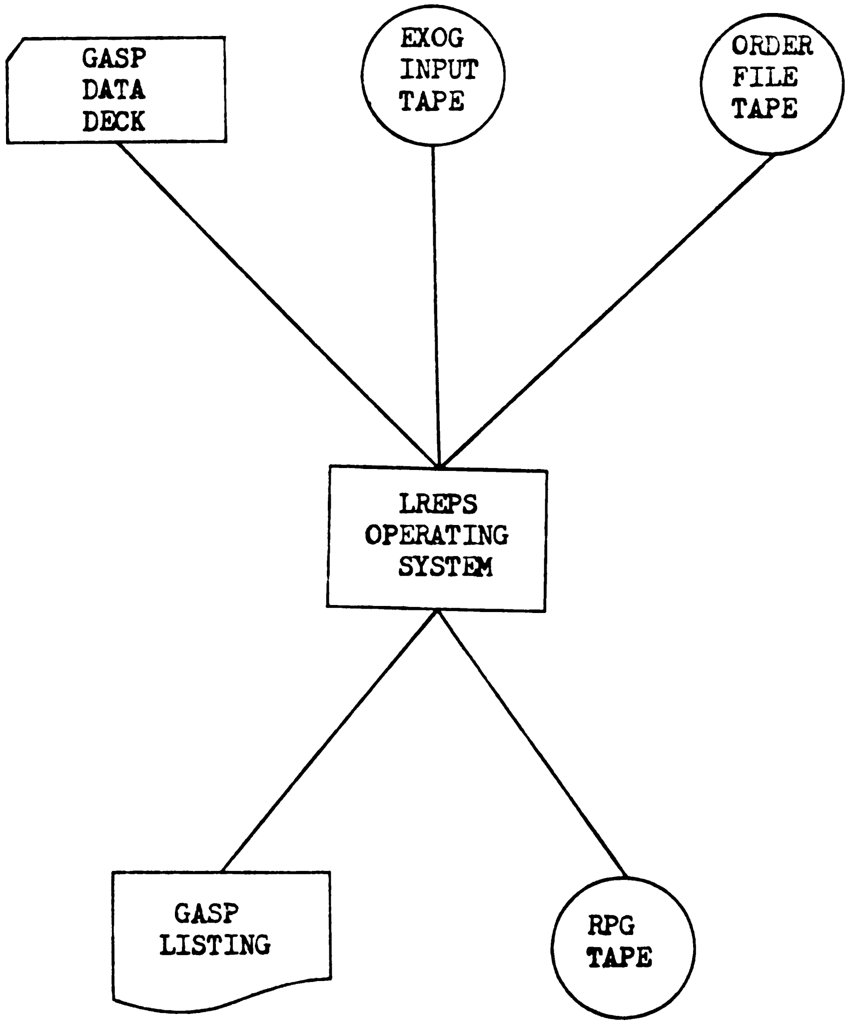


Figure 84.--LREPS Operating System Input/Output Flowchart

the LREPS data base, EXOG and BOQ, used to initialize the DU and DC data base variables, were developed. Besides operationalizing the D&E supporting data system programs, the M&C supporting data system programs for preparing the exogenous input tape plus the program that packed the feasible DC's per DU were also programmed.

The primary reason for operationalizing LREPS-1 was to begin experimenting with alternative techniques for processing the customer order file. As was discussed in Chapter III, the design of the processing procedure for the order file required considerable time and effort. This tape file could have reduced the operating system to an input-bound program resulting in excessively long computer execution time to process a simulation cycle. Therefore, LREPS-1 was a critical step in the overall research since solving the problem about the potentially long input time was fundamental to the operationalization of the mathematical model.

The output of LREPS-1 was sales information at the DC and DU stages of the physical distribution network. Sales information was also accumulated at the regional and domestic levels. This output was printed using the RPG "Variables List" routine.

The purpose of the second version of LREPS, LREPS-2, was to operationalize the merged D&E and OPS subsystems. The system flowchart contained in Figure 84 applied although more inputs were now added to the EXOG input tape.

The OPS subsystem computer program routines were programmed and added to those routines already programmed and tested in LREPS-1. At this point several of the routines used to operationalize LREPS-1 were sophisticated. In particular, DCDU was sophisticated to calculate road distances plus calculate the constant customer service times. The M&C operating system subprograms RWRC, used to calculate the ratio of total to tracked product weight sales, and a basic INVMGT, used to calculate the inventory control variables, were also added to the LREPS operating system.

LREPS-2 had the capacity to process customer orders through the DC's, generate customer service times, and effect MCC to DC interactions as a result of customer sales. The outputs were printed using the RPG "Variables List" program. Besides the sales information listed for LREPS-1, LREPS-2 outputted DC inventory information, MCC to DC reorder information, basic customer service times data and domestic tracked product sales information. In addition to the OPS subsystem supporting data system program, the M&C routine to calculate the DC X and Y rectangular coordinates was operationalized.

The third version of LREPS merged the MEAS subsystem program routines with LREPS-2. In addition, the M&C operating system subprogram EOQ was revised to output not only the LREPS data base but also the quarterly control and DC files of information. The MEAS supporting data system cost information was also added to the exogenous input tape in order to determine the measures of cost for the physical distribution system configuration that was initialized at the beginning of cycle.

Because the DC file of information was added to the RPG tape output, the RPG "DC-based" report program was operationalized in LREPS-3 and used to print the DC reports. With the operationalization of LREPS-3, a given or specified physical distribution system configuration could be initialized at the beginning of cycle and simulated in the LREPS operating system over the planning horizon. The LREPS-3 computer model could not, however, add or delete DC's or modify the sizes of DC's.

In LREPS-4, the remaining M&C subprograms were merged with LREPS-3 to develop the first complete version of LREPS. LREPS-4 did not, however, include the LOCATE, EXPAND and INVMGT routines described in Chapter VI. Rather LREPS-4 included a basic LOCATE routine that only allowed the addition of DC's and an INVMGT routine that calculated inventory control variables according to the policies being used by the industrial research supporter. The output of this version was the same class of output as developed in LREPS-3. The supporting data system inputs were expanded to include the management control parameters and the constraints on allowable physical distribution system changes. All RPG programs were at this point operationalized.

LREPS-5 is the final version of LREPS with which this dissertation is concerned. The substitution of the sophisticated LOCATE routine plus the addition of the EXPAND routine were implemented in LREPS-5. LREPS was now capable of adding, deleting or modifying the DC facility configuration either exogenously on a fixed basis or endogenously on a free basis. With this more sophisticated version of LREPS was the requirement of setting additional constraints on the allowable changes to the physical distribution system configuration.

Further versions of LREPS are planned for development. LREPS-6 will be the version in which the use of partial-line distribution centers is completely operationalized. The basis for LREPS-7 will be a substitution of the theoretical INVMGT routine for the presently operationalized routine. Additional versions of LREPS can be developed by substituting for and adding program modules to those presently developed.

In order to maintain and execute any version of LREPS, all LREPS program modules were contained in a LREPS program library tape. Using control cards, program changes and corrections were updated to this tape. If a certain version of LREPS was desired, then, through a set of control cards, the correct program modules could be linked and compiled together.

Preliminary Model Validation Results

The preliminary validation of the results of the simulation model centered primarily on those activities involved in the calibration of the model's output to a specified base year's data. The model was initially set to run over a four-year period. The first year was the year with which the model was calibrated. The following three years were used to check the model for steady-state implications including no wild fluctuations in the model's output.

Table 1 lists the categories of variables that were checked for reasonableness, the specific data base variables that were checked, an indication as to whether the variables were within a reasonable percent of the base year data plus the physical distribution stage for which the information was checked. This basic validation of the model results was static in nature. One full year's actual results were compared to the simulated results. No sophisticated techniques for simulation validation were used that, for instance, analyzed the model's outputs for autocorrelation or for statistically significant relationships between base year and model data. Dissertations that are forthcoming will devote their attention to an in-depth validation of the model.[1] It is in these dissertations that sophisticated methods and approaches for validating a simulation model will be investigated.

TABLE 1.--One-Year Validation Results.

INFORMATION CATEGORY	DATA BASE VARIABLES	SIMULATED VERSUS ACTUAL	PD STAGES
Cust Sales	DU(13-14) DCIS(16-21)	Within Limits Within Limits	DU, DC and Domestic
Cust Dollar Sales/Order	DCIS(16)/ DCIS(21)	Within Limits	DC and Domestic
Cust Wt Sales/Order	DCIS(17)/ DCIS(21)	Within Limits	DC and Domestic
Line Items per Order	DCIS(20)/ DCIS(21)	Within Limits	DC and Domestic
Cust Serv--			
NOCT-Avg	DCIS(9)	Within Limits	DC and Domestic
NOCT-Std Dev	DCIS(10)	No Data Avail.	DC and Domestic
T4-Avg	DCIS(11)	Within Limits	DC and Domestic
T4-Std Dev	DCIS(12)	No Data Avail.	DC and Domestic
Dollar Props	OCTDIS(1)	No Data Avail.	DC only
Order Props	OCTDIS(2)	Within Limits	DC only
DC-MCC Reorder	DCMCC(1) DCMCC(2) PRDC(6) PRCTDC(2)	Within Limits Within Limits Within Limits Within Limits	DC only DC only DC only DC only
DC Stockouts	PRCTDC(1) DCIS(13-15)	No Data Avail. No Data Avail.	DC only DC only
DC Avg IOH	PRDC(3)	Within Limits	DC only
Cust ship Accums	WTACUM	Difficult to Compare Because of Small Sample Avgs. in Cust Order Blocks	DC and Domestic
MCC Ship Accums	DCMCC1	Within Limits	MCC only
Total Prod Demand	TPDEM(2)	Within Limits	Domestic only
PD Cost	DCCOST(1-6) DCIS(6) PDTCST	Within Limits Within Limits Within Limits	DC and Domestic
Cum Wt Indices	DU(7) DCIS(5) REG(8)	Sales Allocation Basis Within Limits	DU, DC and Regional

CHAPTER VIII--FOOTNOTES

[1]Bowersox, _et al_., Monograph.

CHAPTER IX

RESULTS AND IMPLICATIONS

The first section of this chapter focuses on a discussion of the results in relation to the research questions presented in Chapter I. These questions served as general guidelines and constraints on the formulation of the computer model. The following section of the chapter discusses implications for future research. Initially, implications for research similar to that conducted in this dissertation are presented. This treatment broadly discusses the methodology used in developing large-scale computer simulation models. Next, more specialized, narrow areas of potential research are identified. These potential areas of research resulted from the development of certain aspects of the computer simulation model.

Results Concerning Research Questions

Research Question 1

Research question one states:

> What programming languages will facilitate the development of a reliable computer model

including the supporting data system and the report generator system?

In reference to the LREPS operating system of the computer model, the GASP IIA simulation package facilitated and expedited the computerization of this system. The user-defined and simulation-defined concepts provided by this language assisted in the conceptualization of the computer model. The GASP EXECUTIVE routine alleviated the need to design routines to direct the model through simulated time.

In implementing the GASP IIA simulation package on Michigan State University's CDC 6500, a small delay was experienced because some of the subprograms required minor modifications. Any subprogram that used the pseudo-random number generator DRAND had to be changed to use the pseudo-random number generator, RANF, for the CDC 6500 computer operating system. In addition, the GASP IIA subprograms TMST, COLCT and OTPUT had not been completely modified for the floating point array, QSET. Although these three GASP IIA subprograms were not used in the LREPS operating system, they were used in implementing and debugging the GASP IIA simulation package and, therefore, required modification. One other subprogram had to be modified. The GASP IIA subprogram DATAN had to be renamed as DATAIN since the previous name was already used by CDC as the name of the double precision, arc tangent function. In summary, the GASP IIA simulation package was sound even though it required a few minor modifications.

In reference to the supporting data and report generator systems, the use of the general compiler language FORTRAN was generally the most appropriate programming language. The primary exception to the use of FORTRAN was the use of the CDC 6500 assembly language, COMPASS, for tape handling. The use of COMPASS for tape handling was necessary since FORTRAN is generally an inefficient language where considerable tape handling is required. The FORTRAN tape handling routines are usually slow plus their use of magnetic tape storage space is inefficient.

In summary, the use of a simulation language, a general compiler language and an assembly language was necessary to computerize LREPS. The complexities of the system model, the desire for efficiency in program execution and core memory utilization plus the use of magnetic tapes required the use of all three programming languages. Whether high-level simulation languages such as GPSS or SIMSCRIPT II could have been efficiently and effectively used in LREPS computerization is a possible area of more in-depth research.

Research Question 2

Research question two states:

> What model building procedures will facilitate the development of the computer model, the later sophistication of the model's defined activities, the broadening of the model to encompass additional horizontal and vertical aspects of the total business system, and

satisfy the strong desire for universal applicability among packaged-goods firms?

The use of building blocks as the common basis for many of the model's activities facilitated the development of the computer model. In particular, the basic building blocks programmed in the LREPS operating system served four major purposes. These purposes were:

1) The use of these blocks allowed the project team to allocate efficiently their time among the specific activities of the mathematical model.
2) The blocks facilitated the changing of the computer program when a better approach to the computer modeling of an activity was developed.
3) Basic building blocks forced the examination of common elements of system activities. Instead of developing a subprogram for each potential DC location, the general or universal aspects of the functions of a DC were modeled and computerized. Thus, the same computer subprograms could be used for the functions of many DC's.
4) Basic building blocks saved computer memory while execution time increased very, very slightly because of the additional subprogram linkages.

In general, the use of basic building block subprograms facilitated the overall development of the computer model and especially the LREPS operating system. Changes in a major activity of the mathematical model were easily made by changing the subprogram associated with that activity. Initially, the direction was to have a proliferation of subprograms with its accompanying set of program linkages' problems. However, later the subprograms were redefined in respect to the major components of the mathematical model. This redefinition partially alleviated the problem of many, little subprograms.

The future use of computer subprograms for model sophistication would seem to be easily and practically achieved. Their use for broadening the computer model's operating system to encompass additional horizontal and vertical aspects of the total business system is limited only by the remaining amount of computer memory and the applicability of the present building blocks to the possible activities that would be added to LREPS.

The universal implications of the present building blocks for packaged-goods firms is limited by their compatibility with the activities defined in the model framework of a physical distribution system. The overall design of the model would not have to be modified although some of the model's activity subprograms may require slight modifications.

Research Question 3

Research question three states:

> What computer software and hardware features will allow the structuring of the simulation model's data base and its input/output requirements that will minimize reprogramming these structures for alternative LREPS versions?

With the use of both the special hardware features and the sophisticated software features of the CDC 6500, the required reprogramming for alternative versions of LREPS was minimized. The hardware features centered on the effective use of magnetic tapes, disks and punched cards where speed and controls were needed in programming and debugging LREPS. The software features centered on the definition of the LREPS common data base plus the effective use of the program and information, storage and retrieval systems, that are discussed in Chapter VIII, for MSU's CDC 6500. Although the use of many of these storage and retrieval systems is somewhat difficult to learn, MSU's systems were easily learned and expedited the computerization process of alternative versions of LREPS. This was true even though the files had to be recreated several times when they were lost because the computer system faltered.

Research Question 4

Research question four states:

> What computer software and hardware features will promote the efficient processing of the great amounts of information that will accompany a model spanning a maximum ten-year planning horizon?

Research considering alternative methods for storing, retrieving and processing the computer model's data base of information resulted in the use and disuse of certain computer hardware and software features. The more important features that were investigated are discussed below.

Sequential, magnetic tape files were majorly used for the inflow and outflow of information manipulated in the LREPS operating system. Initial tests using random-access, disk files for banks of information related to the order file showed this method of information processing to be much slower and more expensive than the tape-handling procedure that was developed. The remaining information in the data base that could effectively be random-accessed was so small that this information was kept in high-access, core memory. The use of low-access, extended core storage (ECS) for this information was not possible since ECS was unavailable on MSU's CDC 6500.

The periodic input and output of information into the LREPS operating system via the M&C gatekeeper proved to be efficient and effective. Magnetic tape was also used as the medium for this input and output. The only exception was the limited use of punched cards and printer listings.

The use of computer program overlays for those portions of the LREPS operating system that were only referenced quarterly or less frequently was also investigated.

Program overlays were not used because the present overlay procedure for FORTRAN programs on the CDC 6500 is not easily implemented and quite complicated.

The procedure of packing more than one piece of information in a computer word of memory was used. Much of the information in the model was based upon a zero-one, binary representation. Where a certain category of information was based upon this representation plus there was a considerable amount of information involved, packing was used.

Research Question 5

Research question five states:

> What is the most efficient way to write the model's output in order to provide reports of the essential data and to provide storage that can be easily accessed later as input to programs for further simulation results' analysis?

The periodic output of simulation results via magnetic tape has proved to be very efficient and effective. The reports described in Chapter VII were easily prepared plus the basic simulation data is available for further analysis. Also by establishing the DAILY and MONTH fixed-time events in the LREPS operating system, the use of these routines to print any additional data was easily effected. In summary, the periodic orientation to model results has been satisfactory.

Research Question 6

Research question six states:

> Can a LREPS model be developed that will by highly compatible among different computer manufacturing systems?

LREPS is presently machine dependent and has low compatibility among other computer operating systems. Low compatibility among computer operating systems is stressed because it would be fairly difficult to implement the present computer model on even another CDC 6500. Large computer operating systems for machines such as the CDC 6500 generally have peculiarities that have been developed by university personnel and which cause conversion problems.

Considering other computer manufacturers' operating systems, LREPS' implementation would be very difficult on their systems. Because more than one piece of information was packed into many computer words, the availability of EXTENDED FORTRAN routines to unpack the information would be required. Associated with the packing of information are the differences in the physical sizes of the computer memory words among different computers. The CDC 6500 has sixty bits per word while the IBM 360 series has thirty-two bits per word. Since some of the LREPS data base variables have more than thirty-two bits of information packed into one computer word, then the LREPS data base would have to be redesigned for two words of IBM memory.

The level of the FORTRAN compiler itself would prohibit some computer operating systems from being used for LREPS. MSU's FORTRAN compiler is an advanced version that includes such features as "logical" storage and instructions, tape buffering routines, plus NAMELIST and variable formatting procedures.[1] The use of such features requires a large, sophisticated FORTRAN compiler.

Research Question 7

Research question seven states:

> Can a model be developed that will run on medium-size computer operating systems?

The results concerning research question seven are closely related to the results of research question six. The present design of the LREPS computer model requires a large-scale system such as the CDC systems available at Michigan State University. LREPS requires a large-scale computer system for fast compilation and execution. LREPS also requires a machine with large amounts of high-access computer memory. To retain the universal aspects of LREPS, a sophisticated computer operating system that allows one to retrieve and compile quickly the LREPS program routines is also needed.

With these requirements on the size and speed of the computer operating system, a medium-size system would generally not be capable of processing effectively and efficiently the LREPS simulator.

Research Question 8

Research question eight states:

> Can a computer program of the main operating system of LREPS be developed that will:
>
> a) Cycle through a ten-year planning horizon within a total elapsed computer time of 30 minutes?
>
> b) Fit within a computer memory limitation of 36K decimal words?
>
> c) Require no more than three input and output files?

In reference to the computer memory utilized in the execution of the LREPS operating system, the most sophisticated version presently requires a little less than thirty-two thousand decimal, computer words of memory. Without primarily the capability of packed computer words, the restriction of fitting the LREPS operating system within 36K decimal words of computer memory would never have been satisfied.

In order to restrict the amount of information that was inputted and outputted from the LREPS operating system, it was desired that the developed computer program of the LREPS operating system require the usage of only three input/output files. The restricted use of only three input/output files would promote the efficient use of computer operating systems plus allow the possibility of converting LREPS easily to other computer operating systems. However, Figure 84 illustrates the use of five input/output files. Originally, it was planned that no cards be read from the card reader or no output be printed during program execution of the LREPS operating system. The fact that GASP required card input and outputted control listings on the printer was overlooked. However, considering the small amount of card input and printed output by the GASP subprograms, the desire for the use of only three input/output files should be considered satisfied.

The first part of research question eight is concerned with the execution time of the LREPS operating system. In order to minimize the cost of executing the LREPS operating system, the desire was to develop a computer program that would process a ten-year cycle of information in thirty minutes. This execution time is in reference to both computer central processing and peripheral processing time since they are presently about equal. Examining the computer times of several test runs, the length of time for running a ten-year cycle was primarily a function of two factors.

The first factor was the total sales dollar forecast for the market regions being processed. If all the regions were being processed, then it was the domestic total sales dollar forecast. The running time had a high, direct correlation to this factor when considering the method for processing customer sales from the order file. With the buffering of the order file into the LREPS operating system, the central processing time of program execution was just slightly greater than the peripheral processing time associated with input and output.

Given a larger or smaller sales forecast, then the model took respectively more or less time to process the sales.

The second major factor affecting execution time was the average sales dollar amount per customer order. This factor had an inverse relationship to the amount of time that it took to execute a simulation cycle. If the average dollar amount per customer order was large, then the sales dollar quotas per day were filled much faster which caused the operating system to execute faster. Other factors were considered in the investigation. One of the more important factors was the number of tracked products. After running the program with twelve tracked products and then fifty tracked products, there was no appreciable difference in running time. Most of the tracked product processing was done while the order file was being buffered in.

In order to obtain some idea as to how fast the program would run given various levels of the two major factors, the execution times of several LREPS tests were analyzed in detail to derive a formula that could be used to approximate the running time of the LREPS operating system. The formula which gives the computer time in minutes of computer central processing unit (CPU) time is:

$$T = \sum_{i=1}^{NYRS} \frac{45 \cdot TDSF_i}{ADPCO_i}$$

where:

T is the total amount of CPU time per simulation cycle,

NYRS is the total number of simulated years to be run,

$TDSF_i$ is the total sales dollar forecast (in millions of dollars) for the ith year and the market region(s) under analysis,

and $ADPCO_i$ is the average sales dollars per customer order in year i.

As an example of the use of this formula, if the annual sales forecast were one hundred million dollars and the average sales dollars per customer order were five hundred dollars, the expected running time for that year of the cycle would be nine minutes. If there were ten years that had about the same sales forecast and average dollars per customer order, then the simulation cycle would have taken approximately ninety minutes. This is much greater than the desired thirty minutes. However, with that same forecast for all ten years plus the same average dollars per order, approximately two million customer orders would have been processed. For that amount of customer order processing, the time does not seem unreasonable.

A similar example requiring a total computer run time through a ten-year cycle of thirty minutes or less could have been constructed. However, when considering the consumer-oriented firms who might be using such a model as LREPS, the computer run times per cycle seem unreasonable. This is especially true if firms with large sales forecasts and fairly small customer orders were to use the LREPS operating system.

Implications for Future Research

This section of the chapter is subdivided into two major sections. The first section focuses on a discussion of critical aspects in the design, implementation and testing of large-scale computer simulation models. The identification of these critical aspects will hopefully assist others in the development of the approach to and the actual implementation of computer models. The second section discusses future potential areas of research. The identified areas of possible research resulted from the implementation of various computer modeling concepts plus computer limitations and inefficiencies that exist in relation to the presently operationalized computer model.

Critical Aspects in Computer Modeling

In computerizing the LREPS mathematical model, several aspects of this process, subject to the constraints outlined in Figure 4 of Chapter I, were critical. In order to assist others who may formulate similar computer models, these critical aspects are discussed in a step-wise procedure that system analysts should perform.

After completing the feasibility study and early in the formulation of the specifications of the mathematical model, the selection of the predominant computer programming language should begin. The modeling concepts that are offered by the available simulation languages aid in the determination of the type of simulation model to be designed plus aid in determining the information requirements of the model. In addition, if a simulation language, rather than a general purpose programming language, is considered as the predominant language, then the development of simulation timing routines will not require the attention of the system analysts.

In defining the type of simulation model and its related information requirements, those areas of the system model which could require the handling of large amounts of information should be identified. The type and amount of information can be a major factor in the decision as to which simulation language will be used. The design of computer procedures for processing the potentially large files of information should begin even before the final selection of the predominant computer language. In LREPS, the customer order file is an example of one of these areas of the model that required the processing of large amounts of information. The processing of customer information could have required considerably more time in both the supporting data and the operating systems of LREPS than is presently required. Associated with the customer order file was the processing of tracked products for inventory control and management purposes. The information requirements of a tracked product necessitate large amounts of computer memory plus peripheral storage capacity. The procedure for handling and extrapolating the tracked product information required considerable time to design.

The approach in reference to the above areas should be to design and test various alternative computer procedures as quickly as possible. Testing should take place even though major revisions in the procedures may take place many times. The investigated procedures should include, for example, the use of tape and random-access peripheral equipment, packed information in computer words and sampling approaches to estimation.

The result of these activities should be the development of efficient procedures for processing the large files of information. The procedures should be efficient from the perspectives of both computer memory and execution time.

As computer procedures are investigated and designed for model areas requiring the handling of large amounts of information, the type of the simulation model should be explicitly defined. The simulation model can be defined as either a fixed-time of variable-time model, continuous or discrete, dynamic or static and whatever other classifications are necessary. The type of system model limits the applicable simulation or programming languages and, therefore, should then be determined at this point in the formulation of the computer model.

To further substantiate the defined type of system model, the next critical aspect of designing the computer model should be a preliminary specification and estimation of the size of the model's data base of information. This aspect is also very important in the selection of the programming languages to be used. The preliminary layout of the data base forces the definition of initial estimates of the number of data base variables plus dimensions of these variables. If large amounts of information are required, then the use of packed computer words or random-access files may be required and, thus, the applicable programming languages for implementing the simulation model are further restricted.

Before the final selection of the predominant programming language, one aspect that is often forgotten in the early stages of the development of the computer model is the type of managerial reports, including their format, that will be supplied to the decision makers. Preliminary layouts of the managerial reports should be designed and reviewed with top management. The initial report layouts should not be overly sophisticated but contain enough detail to satisfy the required categories of target variables identified in the mathematical model. Preliminary layout of report formats forces a review of data base specifications to ensure the availability of the necessary data. If special reports are desired, then again the applicable simulation languages are limited.

Finally, the selection of the predominant computer programming language should be made based on a set of criteria similar to those in Figure 10 of Chapter II. After selecting the predominant programming language, the available compiler should be tested immediately via the use of documented, sample problems. The basic features of the language that will most likely be used should also be tested. The above activities should be done if available programmers have not used the simulation language extensively or recently.

After selecting the predominant programming language, the next critical aspect in the design of the computer model should be the specific definition of the model's data base of information. Associated with this activity should be the development of the major subprograms of the computer model. By striving to get the detailed specifications of the data base, the system design activities of specifying subprogram transformations should proceed from the abstract to the concrete.

In determining the dimensions of data base variables associated with a certain system activity, the system design alternative that requires the least amount of core memory should be initially selected. If this procedure is followed, the reliability of other components of the computer model can often be enhanced because additional computer memory is available for sophisticated modeling concepts.

In developing the detailed specifications of the computer model and ensuring that the data base is sufficient for preparing the desired managerial reports, the data base will be redefined many times. Redefinition is often required because certain modeling activities require too much core and/or processing time. By far, the most frustrating aspect in the design of the computer model is the redesign of the specifications of the data base. The end result of the data base definition should be the assignment of variable mnemonics plus the dimensions of each variable similar to that given in Appendix 1.

After designing the data base and segmenting the model's subprograms, the programming of the model should move swiftly. The testing and subsequent merging of each of the major subsystems of the model facilitates the computerization task. By testing each major subsystem with the previously computerized subsystems plus skeleton or basic activities from the remaining subsystems, concentrated effort aids in getting the model operationalized and validated.

One last aspect in the computerization of large-scale simulation models at universities is the beneficial use of undergraduate students for basic, academic research similar to that conducted for the LREPS project. The research project provided the students good training and financial support plus provided the project team with a talented group of programmers. Training was sometimes carried to an extreme when their desire to become familiar with some sophisticated aspect of the CDC 6500 computer operating system often made computer programming tasks more complicated and sophisticated than necessary. Their desire for sophistication caused many tasks to be redone in a simpler method.

Future Potential Research

An important area for additional research is the potential impact upon model outputs resulting from alternative groupings and categories of tracked products. Tracked products were used for pseudo-customer order generation, inventory control purposes and inventory management purposes. The traditional ABC analysis based on sales dollar volume per product may not be the most representative, categorical basis for estimating average inventory characteristics. Weight sales, product density, shipping characteristics, shipping and packing characteristics are examples of alternative categorical bases. Instead of using a multi-level manufacturing control center and distribution center stage's model, a small-scale, single-level system model could be used for this analysis.

A related area of research is an analysis of model results using one categorical basis for grouping the total product line but different samples of tracked products. Different size samples as well as more than one sample of the same size could be investigated. The purpose of the research would be to determine the sensitivity of model results to alternative product samples.

The validity of the hypothetical basis for calculating the sales modification factor presently used in LREPS plus the hypothetical basis for modifying forecasted sales should be determined via controlled research. The present relationship between speed of customer service and forecasted sales is just one of many relationships that could be tested in LREPS. In addition, sales-to-service relationships that incorporate the consistency or reliability of customer service could also be investigated. Consistency of customer service may be more important than speed of service on realized sales.

The investigation of the bases that top management utilize in making decisions concerning changes in their physical distribution system configurations is an important area of potential research. For example, the LREPS procedure for adding or deleting distribution centers should be validated. Top management may not review and change its facility configuration based on the modeled customer service and sales criteria. Perhaps, cost considerations along with the above criteria would be a more representative basis for changing the facility configuration in the model. Combinations of customer service, sales and cost could be investigated for their validity and representativeness.

The next area of potential research is the effect on the design of the LREPS computer model if low-access, extended core storage would have been available. The possibility of redesigning the computer model to more efficiently process the customer order file or store the tracked product detail could have important ramifications on the execution time of the model plus the compatibility of the model among other manufacturer's computer operating systems. Also the storage of the LREPS operating system subprograms used only on quarterly or less frequent basis in extended core storage should be investigated. Large savings in computer memory and execution time costs might be realized.

From the computer programming perspective, an analysis as to the amount of effort to convert the present LREPS computer model from GASP IIA to SIMSCRIPT II would be beneficial. SIMSCRIPT II offers high potential for large-scale simulation models such as LREPS. LREPS requires the use of packed computer words of memory, the use of auxiliary tape files of information plus the ability to process variable-time activities. LREPS also requires large arrays of information, many three dimensional, for storing information related to the model. The applicability of SIMSCRIPT II could be investigated given the above system model requirements. The investigation must be conducted in conjunction with large-scale IBM 360 systems since the SIMSCRIPT II compiler is presently only available on those systems.

The use of LREPS as an educational, business game could also be developed. Supporting teaching materials could be developed that emphasized an understanding of such concepts as the use of tracked products for inventory purposes, the total and normal order cycle time elements, and the five basic elements of a physical distribution system. Other concepts incorporated into LREPS could be illustrated through the use of the computer model.

The extension of the model to take into consideration additional aspects beyond its present scope could also be the subject of additional research. Related to the present LREPS computer model would be an analysis of the computer memory and execution time requirements of the computer model if analytical techniques, such as general linear programming algorithms, were used as part of the facility location subprogram. The basic question would be the efficient use of these analytical techniques to approach optimum locations concerning the DC facility configuration.

Other possible extensions of the model would be an analysis of the incorporation of other stages of the total channel of distribution. Two possible areas would be of immediate concern. The first possible extension of the model would be the incorporation of the physical supply aspects related to the MCC stage of the physical distribution network illustrated in Figure 1. The primary areas of inquiry would be an analysis of the computer requirements and the possible use of present computer subprograms or building blocks if this extension were to be implemented.

Another possible extension of the model would be the incorporation of activities beyond the demand units' stage or the point of ownership transfer. To incorporate the costs and service time requirements outside the immediate system of control might have an important effect on top management decision making. Again the computer requirements would have to be investigated and, assuming that this extension could be implemented, the sensitivity of model results to an extended system of consideration could be investigated.

The next two areas of possible research are related to the information system that was designed for LREPS. The applicability of additional information system concepts for LREPS could be investigated. The presently utilized concepts that include supporting data system, operating system, report generator system and data base of information could be expanded to consider other aspects of information storage, retrieval and manipulation.

Related to the LREPS information system is an investigation into the further integration of its system with business internal and external information systems. The volatility of the model to a firm over time is a function of the degree with which the model is an integral part of the firm's information systems. Changes in costs, elements of customer service, the marketing components of price, product, promotion and physical distribution must be easily effected through the model's data base for model volatility. As part of a firm's information system, the degree with which the model can be readily utilized at decision points in time is critical.

Related to the integration of the model within business information systems is the timing of the updating and revision of the information utilized in the model. The assistance given to management is only as good as the data inputted into the model. How often the model's data base should be updated could be investigated.

The last general area of possible research is related to the aspects of project management for research similar to that conducted for LREPS. LREPS was developed as basic, academic research utilizing a faculty and student project team. Further research could focus on an investigation as to the effectiveness of such research in a university atmosphere. The project research generally has a secondary position in reference to the student's goals to obtain a degree.

These areas of potential research are exemplary of the benefits achieved from a large-scale project conducted at a university.

CHAPTER IX--FOOTNOTES

[1] Control Data 6400/6500/6600 Computer Systems: FORTRAN Extended Reference Manual, Control Data Corporation, Palo Alto, California, printed in the United States of America, 1969.

SELECTED BIBLIOGRAPHY

Asimow, Morris. Introduction to Design. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1962.

Bowersox, D. J., et al. Dynamic Simulation of Physical Distribution Systems. Monograph. East Lansing, Michigan: Division of Research, Michigan State University, forthcoming.

Bowersox, D. J.; Smykay, E. W.; and LaLonde, B. J. Physical Distribution Management. New York: The MacMillan Co., 1968.

Buxton, J. N. and Laski, J. G. "Control and Simulation Languages." Computer Journal. Vol. 5 (Oct., 1962), pp. 194-200.

Clementson, A. T. "Extended Control and Simulation Languages." Computer Journal. Vol. 9, 3 (Nov., 1966) pp. 215-220.

Control Data 6400/6500/6600 Computer Systems: FORTRAN Extended Reference Manual. Palo Alto, California: Control Data Corporation, printed in the United States of America, 1969.

Efron, R. and Gordon, G. "A General Purpose Digital Simulation-Description." IBM Systems Journal, Vol. 3, No. 1 (1964), p. 57.

Helferich, O. K. "Development of a Dynamic Simulation Model for Planning Physical Distribution Systems: Formulation of the Mathematical Model." Unpublished Doctoral Dissertation, Michigan State University.

Kiviat, Philip J. and Pritsker, A. A. B. Simulation with GASP II: A FORTRAN Based Simulation Language. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1969.

Kiviat, Philip J. Digital Computer Simulation: Modeling Concepts, RM-5378-PR. The Rand Corp., Aug., 1967.

Kiviat, Philip J. Simulation Language Report Generators. The Rand Corp., April, 1966, p. 33-49.

Kiviat, Philip J.; Villanueva, R.; and Markowitz, H. M. SIMSCRIPT II. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1968.

Krasnow, Howard S. and Merikallio, R. A. "The Past, Present and Future of General Simulation Languages." Management Science. Vol. II (Nov., 1964), pp. 236-267.

Krasnow, Howard S. "Computer Languages of System Simulation." Digital Computer User's Handbook. Edited by Melvin Klerer and Granino Korn. New York: McGraw Hill Book Co., 1967, Section I, pp. 258-277.

Markowitz, H.; Hausner, B.; and Karr, M. SIMSCRIPT; A Simulation Programming Language. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1963.

McNeley, J. "Simulation Languages." Simulation. Vol. 9 (Aug., 1967), pp. 95-98.

Naylor, T. H.; Balintfy, J. L.; Burdick, D. S.; and Chu, Kong. Computer Simulation Techniques. New York, London, Sydney: John Wiley and Sons, Inc., 1968.

Parente, R. J. and Krasnow, H. S. "A Language for Modeling and Simulating Dynamic Systems." Communications ACM Vol. 10 (Sept., 1967), pp. 559-560.

Tocher, K. D. "Review of Simulation Languages." Operations Research Quarterly, Vol. 16 (June, 1965), pp. 189-218.

Teichroew, Daniel and Lubin, J. F. "Computer Simulation--Discussion of Techniques and Comparisons of Languages." Communications ACM. Vol. 10 (Oct., 1966), pp. 732-741.

Teichroew, Daniel; Truitt, T. D.; and Lubin, J. F. "Discussion of Computer Simulation Techniques and Comparison of Languages." Simulation. Vol. 9 (Oct., 1967), pp. 181-190.

Underwood, R. S. and Sparks, Fred W. Analytic Geometry. Boston: Houghton Mifflin Company; Cambridge: The Riverside Press, 1956.

APPENDIX 1

DATA BASE INFORMATION

The data base of the LREPS operating system is listed on the following pages. The coding scheme for each column of information is:

1) Column one, "NAME," contains the name or mnemonic of the variable. If the variable is multi-dimensional, a more descriptive name of the array is footnoted at the bottom of the page.

2) Column two, "COL," refers to the dimensions of the variable. If the variable has only one dimension, then this column is blank. Multi-dimensioned variables are given a number from one to n for each of the n dimensions of the variable.

3) Column three, "DLT," refers to the delta-time unit or frequency with which a variable can be changed. "C" refers to cyclic; "A" refers to annual; "Q" refers to quarterly; and "D" refers to daily.

4) Column four, "TYPE," designates the variable as either exogenous, "X," or endogenous, "N." The "N" variables could have been set exogenously at the beginning of the simulation cycle. During cycle execution, the "N" variables are only altered within the operating system.

5) The next four columns, "M&C," "OPS," "D&E," and "MEAS," indicate the data base variables used or altered by the model's major subsystems. "U" signifies that the data base variable is only used by the specified subsystem. "S" signifies that the data base variable is used and altered in the subsystem.

6) Column six, "MODE," signifies whether a variable is "R," a real, fractional variable, "I," an integer variable, or "P," a packed variable containing more than one piece of information per computer word.

7) Column seven, "CONTENTS," is a description of the contents of each of the variables in the data base. If the variable is multi-dimensional, then each dimension or column of the variable is described.

APPENDIX 1.--Data Base Information.

Name	COL	DLT	TYPE	M&C	OPS	D&E	MEAS	MODE	Contents
DU[1]	1	C	X	U		U		R	Special customer indicators
	2	C	X	U				R	Rectangular x-coordinate converted from latitude for hub city
	3	C	X	U				R	Rectangular y-coordinate converted from longitude for hub city
	4	A	X	U				R	Weighted index used for sales allocation
	5	Q	N	S	U			P	Customer service time ring numbers T1 & T4 for majorly assigned DC and T4 for regional PDC
	6	Q	N	S		U		I	Sorted DU code for DU's assigned to in-solution DC
	7	Q	N	S		U		R	Cumulative weighted index used for selecting randomly DU's for sales allocation
	8	Q	N	S	U	U	U	I	Code identifying the row of the in-solution DC array to which the sorted DU code is linked
	9	Q	N	S				R	Exponentially averaged quarterly dollar sales
	10	Q	N	S				R	Exponentially averaged quarterly sales in pounds
	11	Q	N	S			U	R	Calculated highway distance from the full line DC
	12	Q	N	S			U	R	Calculated highway distance from the partial line DC
	13	D	N	S	S			R	Quarter-to-date dollar sales for the full line DC
	14	D	N	S	S			R	Quarter-to-date dollar sales for the partial line DC
	15	D	N	S	S		U	R	Quarter-to-date sales in pounds for the full line DC
	16	D	N	S	S		U	R	Quarter-to-date sales in pounds for the partial line DC
DCP[2]	1	C	X	U				R	Rectangular x-coordinate converted from the latitude
	2	C	X	U				R	Rectangular y-coordinate converted from the longitude
	3	C	X	U			U	R	Constant "a" coefficient used for calculating the outbound transportation rates
	4	C	X	U			U	R	Variable "b" coefficient used for calculating the outbound transportation rates based on distance
	5	C	X	U			U	R	Real transportation rate modifier "R1" for the above "a" coefficient
	6	C	X	U			U	R	Real transportation rate modifier "R2" for the above "b" coefficient
	7	C	X	U	U			P	Customer service time variance pointers for T1, T2, T4
	8	Q	X	U	U	U	U	I	Type of DC plus PDC assignment code
	9	Q	X	U				P	Mileage ring set indicators for customer service time elements T1 and T4
	10	Q	N	S				R	Exponentially averaged quarterly dollar sales
	11	Q	N	S				R	Exponentially averaged quarterly weight sales
	12	Q	N	S				R	Time when a DC was added, deleted or its capacity changed
DCIS[3]	1	Q	N	S			S	R	Approximated backorder penalty time (T3)
	2	Q	N	S	U	U	U	I	Code of the potential DC location that is in-solution
	3	Q	N	S		U		I	Number of demand units assigned to this DC
	4	Q	N	S		U		R	Sales modification factor used in determining DC simulated sales dollars--exponentially averaged
	5	Q	N	S		U		R	Sum of the weighted indices for the demand units in this DC's geographic area
	6	Q	N	S			S	R	Total physical distribution cost for the past quarter's activities
	7	Q	N	S			S	R	Average total order cycle time
	8	Q	N	S	U		U	R	Indicator of the sales dollar capacity of this distribution center
	9	D	N	S	S		S	R	Normal average order cycle time that is computed from the sum of the T1 + T2 + T4 for all orders in the last quarter
	10	D	N	S	S		S	R	Standard deviation of the normal average order cycle time which is computed from the sum of the square of T1 + T2 + T4 for all orders in the last quarter
	11	D	N	S	S		S	R	Average outbound transportation time (T4) which is computed from the sum of the T4's for all orders in the last quarter
	12	D	N	S	S		S	R	Standard deviation of T4 which is computed from the sum of the square of T4 for all the orders in the last quarter
	13	D	N	S	S		S	R	Percent of case units backordered of the total case units sold for the tracked products
	14	D	N	S	S		S	R	Average stockout delay, given a product stocked out, which is computed from the sum of the individual product delays
	15	D	N	S	S		S	R	Standard deviation of the stockout delay which is computed from the sum of the square of the individual product delays
	16	D	N	S	S		U	R	Quarter-to-date dollar sales
	17	D	N	S	S		U	R	Quarter-to-date weight sales
	18	D	N	S	S		U	R	Quarter-to-date cube sales
	19	D	N	S	S		U	R	Quarter-to-date case sales
	20	D	N	S	S		U	R	Quarter-to-date line sales
	21	D	N	S	S		U	R	Quarter-to-date order sales
REG[4]	1	Q	X	U				I	Maximum number of DC's allowed in system
	2	Q	X	U				I	Maximum number of DC's that can be added
	3	Q	X	U				I	Maximum number of DC's that can be deleted
	4	Q	N	S	S			R	QTD tracked product, weight sales
	5	Q	X	U				R	Desired level of customer service
	6	Q	X	U				R	Maximum allowed dollar investment in DC's
	7	Q	X	U				R	Maximum allowed dollar investment for in-process DC's
	8	Q	N	S		U		R	Sum of the weighted indices for the region's demand units
	9	Q	N	S				I	Number of DC's in the process of being added

	10	Q	N	S				I	Number of DC's in the process of being deleted
	11	Q	N	S				R	Total dollar investment for DC's in solution
	12	Q	N	S				R	Total dollar investment for in-process DC's
	13	Q	N	S			S	R	Actual service achieved for the last quarter
	14	Q	N	S				R	Exponentially averaged sales modification factor
	15	Q	N	S	U			R	Ratio of total product sales to the tracked product sales in pounds
	16	Q	N	S			S	R	Total physical distribution cost for activities over the last quarter
	17	Q	N	S				I	Number of DC's that are presently in-solution
MCC[5]	1	Q	X	U	U			I	T2 variance pointer for processing and preparing reorders at the MCC stage
	2	Q	X	U	U			I	MCC scheduled shipment dispatch policy that is given in number of days
	3	Q	X	U			U	R	Cost of preparing and processing a multiple-product reorder at a MCC
DCMCC[6]	1	D	N	S	S		U	I	Total number of multiple-product reorders
	2	D	N	S	S		S	R	Accumulated total reorder lead time plus later used for calculating the average reorder lead time
	3	D	N		S			P	Packed computer word containing the products that are presently on reorder at the MCC from the DC
	4	D	N		S			I	Number of multiple product reorders that are outstanding at the MCC
	5	D	N		S			R	Total weight on order at the MCC plus an indicator of whether a shipment had been dispatched within the past scheduled shipment review period
PRCT[7]	1	C	X	U	U			I	Indicator of the categories handled by partial line DC's
	2	C	X	U	U			I	Beginning product model code for this category
	3	C	X	U	U			I	Ending product model code for this category
	4	C	X	U			U	I	Total number of products that this category is representing
	5	Q	X	U	U			I	Type of inventory policy
	6	Q	X	U	U			R	Review period length for above policy
	7	Q	X	U				R	Safety stock factor for above policy
PRDC[8]	1	Q	N	S	U			P	Packed computer word containing reorder point 1 (ROP1) and reorder point 2 (ROP2)
	2	Q	N	S	U			R	Replenishment S-level
	3	D	N	S	S		S	R	Time-integrated inventory on hand plus outstanding reorder quantities used in calculating the average total inventory per day
	4	D	N	S	S			R	Inventory on hand
	5	D	N		S			I	Number of days that a product has been stocked out given a stockout
	6	D	N		S			R	Reorder quantity (ROQ) plus an indicator that a reorder is presently outstanding
PRCTDC[9]	1	D	N	S	S		S	I	Quarter-to-date stockouts
	2	D	N	S	S		S	I	Quarter-to-date single-product reorders
	3	D	N	S	S		S	R	Time-integrated stocked-out case units which will be divided by total case unit sales to get the approximated customer service time T3 by category--tracked product basis
	4	D	N	S	S		S	R	Total number of case units sold for the tracked products
CGSCS		C	X	U			U	R	Cost of goods sold per case unit by tracked product
CUBCS		C	X	U			U	R	Average cube per case unit by tracked product
WTCU		C	X	U	U		U	R	Average weight per case unit by tracked product
OBTNRM[10]	1	C	X	U			U	R	R3 modifier of the "a" coefficient by region
	2	C	X	U			U	R	R4 modifier of the "b" coefficient by region
DCP1[11]	1	C	X	U	U			P	Packed computer word containing the tracked products linked to a specific MCC per potential DC
	2	C	X	U	U			R	Proportion of total weight that will be supplied via this link
	3	Q	X	U	U			P	Inbound reorder transmission time and transit time factors
MCCWB		C	X	U	U			R	Maximum weight breaks for shipment dispatches from each MCC
DCWB		C	X	U	U			R	Maximum weight breaks for the DC shipment accumulators
RING		C	X	U				R	Alternative customer mileage ring sets for 1, 2 and 3 day average service times
DCASGN		C	X	U				P	Packed computer words containing the feasible DC's, in decreasing priority, that can service a specific DU
CSTBLK		C	X	U		U		I	Number of order blocks per major customer type per order group
CLF		Q	X	U			U	R	Potential DC cost of living factors

[1] Demand Unit--DU(COL)

[2] Potential DC Location--DCP(COL)

[3] DC's in Solution--DCIS(COL)

[4] Regional--REG(COL)

[5] Manufacturing Control Center--MCC(COL)

[6] DC-MCC Link--DCMCC(COL)

[7] Product Detail by Inventory Category--PRCT(COL)

[8] Product Detail by DC--PRDC(COL)

[9] Product Category by DC--PRCTDC(COL)

[10] Outbound Transp Negotiated Rate Modifiers--OBTNRM(COL)

[11] Potential DC-MCC Product Link--DCP1(COL)

APPENDIX 1.--continued.

Name	COL	DLT	TYPE	M&C	OPS	D&E	MEAS	MODE	Contents
IUDF[12]	1	Q	X	U				I	PD facility changes flag
	2	Q	X	U				I	Transportation related changes
	3	Q	X	U				I	Inventory management policy changes
	4	Q	X	U				I	Unitization or materials handling changes
	5	Q	X	U				I	Communication's related changes
DCCAPC		Q	X	U	U		U	R	Capacity constraint by size of DC
DCP2		Q	X	U			U	R	Potential DC-MCC freight rates by weight class
COMMCR[13]	1	Q	X	U			U	R	Fixed cost factor
	2	Q	X	U			U	R	Variable cost per order
	3	Q	X	U			U	R	Variable cost per line item
COMMCD[14]	1	Q	X	U			U	R	Fixed cost factor
	2	Q	X	U			U	R	Variable cost per order
	3	Q	X	U			U	R	Variable cost per line item
CSPLTP		Q	X	U		U		R	Regional customer dollar split percentages
THRUPC		Q	X	U			U	R	Throughput cost rate per pound by DC size and type
INVEST		Q	X	U			U	R	Average investment dollars by size and DC type
COMMC[15]	1	Q	X	U			U	R	Fixed cost factor
	2	Q	X	U			U	R	Variable cost per order
	3	Q	X	U			U	R	Variable cost per line item
DCCOST[16]	1	Q	N	U			S	R	Outbound transportation cost
	2	Q	N	U			S	R	Inbound transportation cost
	3	Q	N	U			S	R	Throughput cost
	4	Q	N	U			S	R	Communications cost
	5	Q	N	U			S	R	Facility cost
	6	Q	N	U			S	R	Inventory carrying and handling cost
REGSAC		Q	N	U			S	R	Regional, quarterly dollar and weight sales accumulators
DOMSAL		Q	N	U			S	R	Domestic, quarterly dollar and weight sales accumulators
OCTDIS[17]	1	D	N	S	S		S	R	Proportion of sales dollars within 3, 5, 7, 9 and 11 days
	2	D	N	S	S		S	R	Proportions of orders within 3, 5, 7, 9 and 11 days
WTACUM		D	N	S	S			R	Accumulators of the weight shipped from each of the in-solution DC's by weight class
DCMCC1		D	N	S	S		U	R	Accumulators of the weight shipped from each of the MCC's to the in-solution DC's by weight class
TPDEM[18]	1	D	N	S	S			R	Quarter-to-date total domestic demand for each tracked product
	2	Q	N	S				R	Exponentially averaged total domestic product demand for each tracked product
NSCH		C	X	U				I	Day and event numbers for the scheduled fixed-time events
PARM		Q	X	U	U			R	Service time variance functions for generating customer and reorder lead time variances
DCREGP		Q	X	U				I	Regional priority list of DC's to be added via the DC location algorithm
ITNSP		C	X	U	U		U	I	Total number of tracked products for cycle
ITNPC		C	X	U	U		U	I	Total number of inventory categories
NDUS		C	X	U		U	U	I	Total number of DU's being processed
IREGNO		D	N	S		U	U	I	Number of region being processed
TDSFLG		C	X	U		U		I	Flag indicating whether the domestic daily sales quota is a constant or modified variate
DICC		C	X	U			U	R	Daily inventory carrying charge--proportion
MXINVS		C	X	U				I	Maximum domestic dollar investment for in-solution DC's
MXIPDS		C	X	U				I	Maximum domestic dollar investment in facilities that can be deleted
NWKDYS		C	X	U		U		I	Number of simulated workdays per year
ADTMRD		C	X	U				R	Delay time to add a RDC-F facility
ADTMPD		C	X	U				R	Delay time to add a RDC-P facility
ADTMCP		C	X	U				R	Delay time to add a CSP facility
DLTMRD		C	X	U				R	Delay time to delete a RDC-F facility
DLTMPD		C	X	U				R	Delay time to delete a RDC-P facility
DLTMCP		C	X	U				R	Delay time to delete a CSP facility
TDSF		A	X	U		U		R	Total annual domestic sales dollar forecast
IYEAR		A	N	S			U	I	Year of simulated time
SDSM		Q	X	U	U			R	Maximum dollar shipment size that can be considered for daily shipments from the PDC's
CSDP		Q	X	U	U			R	Percent of customer orders less than dollar SDSM that can be shipped daily from a PDC
MAXDCS		Q	X	U				I	Maximum number of DC's allowed domestically
MXIPAS		Q	X	U				I	Maximum number of in-process DC's domestically

MXIVPS		Q	X	U				I	Maximum domestic dollar investment allowed in process
NMINPS		Q	N	S				I	Number of in-process DC's being added domestically
INVSTS		Q	N	S				I	Total domestic dollar investments in DC's
INVSPS		Q	N	S				I	Domestic dollar investment in DC's being added to system
NUMDCS		Q	N	S				I	Number of DC's in PD system
PDTCST		Q	N	S			S	R	Total cost for PD system over last quarter
DOMAST		Q	N	S			S	R	Domestic average total order cycle time for past quarter
NMBDLS		Q	N	S				I	Number of in-process DC's being deleted domestically
IDUNO		D	N	S		S	S	I	Number of DU being processed
IDUNO1		D	N	S	U	S	S	I	Number of DU being processed based on present DC assignment
IDCNO		D	N	S	U	U	U	I	Number of in-solution DC being processed
IDCNO1		D	N	S	U	U	U	I	Code of potential DC being processed
IDCAGN		Q	N	S				I	Code of DC being set in solution
DSQ		D	N	U		S		R	Domestic daily sales dollar forecast
IORD		D	N		U	S		I	Number of order being processed
IBLOCK		D	N		U	S		I	Number of order block that has been randomly selected
OBM		D	N			S		R	Order block modifier to attain a customer's sales allocation
ICTYPE		D	N			S		I	Major customer type being processed
IPBLKS		D	N			S		I	Number of past customer order blocks processed
IBLKS		D	N			S		I	Number of order blocks for this major customer
WIACC		Q	N	S				R	Variable used to accumulate weighted indices by DU and DC
CSTSAL		D	N			S		R	Sales dollar allocation by major customer
CSTSAC		D	N			S		R	Sales dollar accumulator by major customer
DCSALS		D	N			S		R	DC simulated sales for day
NATCON		Q	X	U				I	Flag as to domestic constraints on DC decisions or not
REGCON		Q	X	U				I	Flag as to regional constraints on DC decisions or not
TYPEAD		Q	X	U				I	Flag as to type of algorithm to be used in ADC of DC's
NDCADD		Q	X	U				I	Maximum number of DC's that can be added domestically and quarterly
TMINPR		Q	X	U				R	Delay time to expand an existing DC facility
NMCCS		Q	X		U		U	I	Number of in-solution MCC's
NREGS		Q	X	U				I	Number of in-solution regions
ORDGRP[19]	1	D	X		U	U		R	Order block sales dollar summary
	2	D	X		U	U		R	Order block sales weight summary
	3	D	X		U	U		R	Other order sales summary data
	4	D	X		U	U		R	Order block tracked product detail
RDCPC		D	N		S			R	Percent of order block allocated to full-line DC
PLPC		D	N		S			R	Percent of order block allocated to RDC-P
IDCTEMP		D	N		S			I	Temporary storage of PDC-identification number
DAYCON		Q	X	U			U	R	Customer total OCT decision parameter used for SMF
CM		Q	N				S	R	Inventory category extrapolation modifier
RCF		Q	N				S	R	DC-type rate correction factor
DIST		Q	N				S	R	DC to DU road distance
WT		Q	N				S	R	DC to DU quarterly, accumulated weight
POOLFC		Q	N				S	R	PDC pooled shipment rate mod factor
IVOLP		Q	N				S	R	Number of PDC weight categories
IVOLR		Q	N				S	R	Number of RDC weight categories
FC		Q	N				S	R	Total fixed cost for all geographic units
FCO		Q	N				S	R	Total variable cost/order for all geographic units
VCL		Q	N				S	R	Total variable cost/line for all geographic units
PDSCHG		Q	N	S				I	Indicator of PD system change(s)
SUMWT		Q	N	S				R	Total QTD regional weight sales
DCACTS		Q	N	S				R	DC actual level of service proportion used for calculating DC sales modification factor
DEM		Q	N	S				R	DC tracked product average daily demand
SLT		Q	N	S				R	Standard deviation of reorder lead time
SRP		Q	N	S				R	Standard deviation of review period length
SPD		Q	N	S				R	Standard deviation of tracked product avg daily demand per DC
BUF		Q	N	S				R	DC tracked product buffer stock
EOQ		Q	N	S				R	DC tracked product economic order quantity
ORDCST		Q	N	S				R	Tracked product ordering cost
XINC		A	N	S				R	Monthly domestic sales forecast increment
SRATIO		Q	N	S				R	DC dollar to sales capacity ratio
RLIST		Q	N	S				R	Array used for M&C working lists of regions and DC's
DCSUM		Q	N	S				R	Sum of DC expected DU sales for eligible DC's
NDCIN		Q	N	S				R	Number of DC's put into process this quarter

[12] PD Update Flags for System Changes--IUDF(COL)

[13] Regional COMM Costs--COMMCR(COL)

[14] Domestic COMM Costs--COMMCD(COL)

[15] Potential DC COMM Costs by Size and DC Type--COMMC(COL)

[16] In-Solution DC Cost for PD Activities--DCCOST(COL)

[17] Normal Order Cycle Time Proportions--OCTDIS(COL)

[18] Domestic Product Demand--TPDEM(COL)

[19] Order Block Detail by Group--ORDGRP(COL)

APPENDIX 2

RPG DC-BASED SAMPLE REPORT FORMATS

```
 *****************************************************
*                                                     *
*                                                     *
*                                                     *
*    *         *****    ******   *****    *****       *
*    *         *    *   *        *    *   *           *
*    *         *****    ****     *****                *
*    *         *  *     *        *        *****       *
*    *         *   *    *        *            *       *
*    ******    *    *   *****    *        *****       *
*                                                     *
*                                                     *
*                                                     *
 *****************************************************
```

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR

MICHIGAN STATE UNIVERSITY

TABLE OF CONTENTS

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 3
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

END OF CYCLE SUMMARY

DC	SALES	COST	PROFIT	OC TIME	INV OH
LOCATION 1					
LOCATION 2					
LOCATION 3					
LOCATION 4					
LOCATION 5		WAS NEVER IN SOLUTION			
LOCATION 6					
LOCATION 7		WAS NEVER IN SOLUTION			
LOCATION 8		WAS NEVER IN SOLUTION			
LOCATION 9		WAS NEVER IN SOLUTION			
LOCATION 10		WAS NEVER IN SOLUTION			
LOCATION 11		WAS NEVER IN SOLUTION			
LOCATION 12		WAS NEVER IN SOLUTION			
LOCATION 13		WAS NEVER IN SOLUTION			
LOCATION 14		WAS NEVER IN SOLUTION			
LOCATION 15		WAS NEVER IN SOLUTION			
LOCATION 16		WAS NEVER IN SOLUTION			
LOCATION 17		WAS NEVER IN SOLUTION			
LOCATION 18		WAS NEVER IN SOLUTION			
LOCATION 19		WAS NEVER IN SOLUTION			
LOCATION 20		WAS NEVER IN SOLUTION			
LOCATION 21		WAS NEVER IN SOLUTION			
LOCATION 22		WAS NEVER IN SOLUTION			
LOCATION 23		WAS NEVER IN SOLUTION			
LOCATION 24		WAS NEVER IN SOLUTION			
LOCATION 25		WAS NEVER IN SOLUTION			
LOCATION 26		WAS NEVER IN SOLUTION			
LOCATION 27		WAS NEVER IN SOLUTION			
LOCATION 28		WAS NEVER IN SOLUTION			
LOCATION 29		WAS NEVER IN SOLUTION			
LOCATION 30		WAS NEVER IN SOLUTION			
LOCATION 31		WAS NEVER IN SOLUTION			
LOCATION 32		WAS NEVER IN SOLUTION			
LOCATION 33		WAS NEVER IN SOLUTION			
LOCATION 34		WAS NEVER IN SOLUTION			
LOCATION 35		WAS NEVER IN SOLUTION			
TOTALS					

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 4
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

SUMMARY FOR QUARTER 1

DC	SALES	COST	PROFIT	OC TIME	INV OH
LOCATION 1					
LOCATION 2					
LOCATION 3					
LOCATION 4					
LOCATION 5					
LOCATION 6					
TOTALS					

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 44
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 SALES REPORT

QTR	DOLLARS	WEIGHT	CUBE	CASES	LINES	ORDERS
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
TOT						

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 45
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 COST REPORT

QTR	OBC	IBC	TPC	COM	FAC	INV	TOT
1							
2							

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 46
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 ORDER CYCLE TIME REPORT

QTR	TOT OCT	BO PEN	NOCT	SD NOCT	OBT	SD OBT
1						
2						

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 47
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 REORDER REPORT

	MULTI PRODUCT				AVG LEAD TIME			
QTR	MCC 1	MCC 2	MCC 3	MCC 4	MCC 1	MCC 2	MCC 3	MCC 4
1								
2								

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 48
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 INVENTORY CONDITION REPORT

QTR	IOH	TSO	TRO	PCUBO	AVG SD	SD ASD
1						
2						

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 49
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 IDENTIFICATION REPORT

QTR	NO OF DUS	S-M FACTOR	DU WI SUM	DOL SIZE IND
1				
2				

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 50
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 NORMAL ORDER CYCLE TIME DOLLAR PROPORTIONS
PROPORTION OF SALES DOLLARS WITHIN DAYS

QTR	3	5	7	9	11
1					
2					

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 51
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 NORMAL ORDER CYCLE TIME ORDER PROPORTIONS
PROPORTION OF ORDERS WITHIN DAYS

QTR	3	5	7	9	11
1					
2					

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 52
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

LOCATION 1 WEIGHT CLASS ACCUMULATIONS IN THOUSANDS

QTR	0-50	51-200	201-1000	1001-UP
1				
2				

LONG RANGE ENVIRONMENTAL PLANNING SIMULATOR PAGE 53
MICHIGAN STATE UNIVERSITY
DATE 09/05/70

DC-MCC WEIGHT CLASS ACCUMULATIONS IN THOUSANDS FOR
LOCATION 1 FROM LOCATION 3

QTR	2M-5M	5M-24M	24M-UP
1			
2			

GLOSSARY OF ABBREVIATIONS

AVG	AVERAGE
AVG SD	AVERAGE STOCKOUT DELAY
BO PEN	BACK ORDER PENALTY TIME
COM	COMMUNICATIONS COST
DC	DISTRIBUTION CENTER
DOL SIZE IND	DOLLAR SIZE INDICATOR
DU WI SUM	SUM OF THE DU WEIGHTED INDICES
FAC	ALLOCATED FACILITIES COST: LAND, EQUIP
IBC	INBOUND SHIPPING COST
INV	INVENTORY COST
INV OH	AVG CUBIC INVENTORY ON HAND
IOH	AVG CUBIC INVENTORY ON HAND
MCC	MANUFACTURING CONTROL CENTER
NO OF DUS	NUMBER OF DEMAND UNITS
NOCT	NORMAL ORDER CYCLE TIME
OBC	OUTBOUND SHIPPING COST
OBT	OUTBOUND TRANSIT TIME
OC TIME	ORDER CYCLE TIME
PCUBO	PERCENT CASE UNITS BACK ORDERED
PROFIT	SALES LESS DISTRIBUTION COST
QTR	QUARTER
S-M FACTOR	SALES MODIFICATION FACTOR
SD ASD	STANDARD DEVIATION AVG SD
SD NOCT	STANDARD DEVIATION NOCT
SD OBT	STANDARD DEVIATION OBT
TOT	TOTAL
TOT OCT	TOTAL ORDER CYCLE TIME
TPC	THROUGHPUT COST
TRO	TOTAL REORDERS
TSO	TOTAL STOCKOUTS