OCCASION

This publication has been made available to the public on the occasion of the 50th anniversary of the United Nations Industrial Development Organisation.

DISCLAIMER

FAIR USE POLICY

CONTACT

Please contact publications@unido.org for further information concerning UNIDO publications.

For more information about UNIDO, please visit us at www.unido.org

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
Vienna International Centre, P.O. Box 300, 1400 Vienna, Austria
Tel: (+43-1) 26026-0 · www.unido.org · unido@unido.org

RESTRICTED

15580

12 MAY 1986
ENGLISH

India:

COMPUTER SIMULATION. (Machine tool factory).

DP/IND/82/019/11-03/31.9.B
INDIA

TECHNICAL REPORT : PROJECT PLANNING, ANALYSIS, DESIGN AND SPECIFICATION

Prepared for the Government of India by the United Nations Industrial Development Organization acting as executive agency for the United Nations Development Programme.

Based on the work of A. Aziz Merabet, Expert
in simulation of job-shop manufacturing

United Nations Industrial Development Organization
Vienna

CONTENTS

LIST OF FIGURES

LIST OF TABLES

OVERVIEW

My three month assignment at the Central Machine Tool Institute in Bangalore, India, is part of a larger project with purpose of establishing a CAD/CAM centre and developing consulting capabilities in this area.

Specifically my duties were in the field of computer simulation (DP/IND/82/019/11-03/31.9.B), including :

- -Imparting know-how on simulation techniques for modelling and analysis of descrete event manufacturing.
- -Organizing, initiating, guiding and developing a general job-shop simulation model for a machine-tool factory. The resulting software must be flexible for extension to capacity planning and other manufacturing decisions making functions.
- -Advising on the evaluation and selection of simulation languages and software packages for manufacturing systems.

In order to realize all possible objectives of my mission a project plan for my activities was developed and is described in section 2. Two identifiable research and development ideas resulted from the above task, and can be incorporated in a single industry based project for a local firm. The analysis and specification of this project is given in section 3 of this report, and the overall project is realizable as planned in section 2.

A baseline design strategy is developed, justified and presented in section 4.

The detailed design of the different functions to be implemented was performed in collaboration with four engineers of CMTI. The results are presented in section 5.

These four engineers, after training, will become project leaders for the remaining tasks (software development and testing) of the corresponding sub-projects. These functions are discussed in section 6.

Some conclusions and recommendations are given in section 7.

All objectives of my assignment were completed satisfactorily. At all times my plans and suggestions were discussed with the management of the Institute.

The major difficulty I encountered was the lack of recent documents (books, technical reports and papers) on computer integrated manufacturing systems in general, and simulation,

I. PROJECT PLAN

The following proposal outlines the three options concerning my activities at CMTI during my assignment (3 months).

Every project described in this document, including my work, is organized in three phases (with possible sub-phases) [1] :

Phase 1 : Definition and Specification
- Analysis of problem
- Baseline design of a solution

Phase 2 : Development of Modules

Phase 3 : Validation and Installation
- Testing of modules
- Integration testing

These three phases are of equal length and are controlled effectively by a set of milestone documents. A detailed presentation of this phase-plan approach to project management will be given as a seminar to all project leaders of CMTI and will be the strategy used for evaluation and control of progress.

My duties could be carried out in three possible ways :

Option 1 : The end product of my activities is a complete software program. In that case, I will be designing, coding and testing a specific computer integrated function For a period of 3 months and for 1 person (3 man/month effort) the resulting product will not be more than 300 lines of software = 15 lines per day x 20 days (programming productivity is in the order of 5 to 15 lines per day per person of designed, coded and tested software).

Option 2 : My 3 months assignment is considered as a definition phase only of a larger project (therefore of a 9 months length). I will be then specifying in detail a software function and training a person to complete it when I am gone. The resulting program will be from 1800 lines to 2000 lines of code (estimated as follows : 10 lines per day per programmer x 60 days - i.e. 3 months of development - x 3 to 4 programmers).

Option 3 : During the analysis sub-phase of the definition and specification phase, I organize my work so that for three months I am engaged in analysis (and projects initiation) - This approach will allow me to initiate two research and development projects at least as estimated by the same metrics as previously. These projects will be chosen from a list of seminars I will be giving (with 1 seminar resulting in 2 projects on average).

My recommendation is to take the third option. This will allow me to complete all three duties of my job description. I have identified 6 seminars (therefore 12 projects would be initiated). Two of these projects will be oriented research and development, and the ten others will be oriented as engineering procurements (i.e. evaluating available packages, acquiring and installing them). This identification of seminars (and projects) will be performed more systematically, by first analysing the existing machine-tools, computer hardware and software and projects underway, then submitting a baseline design of a desirable computer integrated cell, and finally proposing a strategy to reach this goal.

It appears that this effort is of the order of 3 years in length and 102 man-months, estimated as follows : 2 R and D projects x (9 months per project leader + 3 months for 3 programmers + 3 months for 1 analyst) + 10 engineering projects x 1 engineer x (3 months of evaluation + 3 months of testing).

The possible seminars and projects are listed below :

Seminars :
- Software Engineering Project Management
- Computer Networks
- Distributed Systems
- Intelligent Systems
- Robotics/Image Processing
- Databases

Projects :

. Research and Development

- Real-time Control(modelling, simulation)
- Real-time Scheduling (classical, heuristic)

. Procurement of packages

- Software Project Management
- Simulation
- Optimum Scheduling
- Planning (MRP)
- Manufacturing Databases
- Local Area Networks
- Robotic Systems
- Expert Scheduling
- Expert Process Planning
- Robot Vision Systems

If Option 2 is chosen, the corresponding project can be one of the two research projects listed under option 3.

Option 1 is definitly not recommended as a 300 lines program can not be of any industrial value.

II.PROBLEM ANALYSIS

The two research and development projects mentioned in the plan of section 2, can be incorporated in a specific industrial application for a local tool manufacturing company. In this project it is required to modify the existing Pre-planning, Planning and Control (PPC) computerized system shown in figure 1 in order to improve the performance of shop-floor operations.

Specifically the new system will :

- Optimize part production and resource utilization
- React effectively to enviroment changes

The approach taken is also shown in figure 1, with these functional levels :

- Simulation : for sequencing of operations in time
- Scheduling : for dynamic allocation and optimization under constraints of resources.
- Monitoring : for real-time update of manufacturing changes.

- Data reduction and acquisition : for on-line observation and update of parameters and configuration

- Organizing of production : for productivity improvement by evaluating manufacting options.

The resulting software package could be implemented on a dedicated micro-processor located in the shop-floor and therefore accessible for interactive feedback by operators of real-time monitored operational data, and/or urgent production requests.

With this approach the new and existing structures will be functionally similar as shown in table 1 and figure 2 respectively.

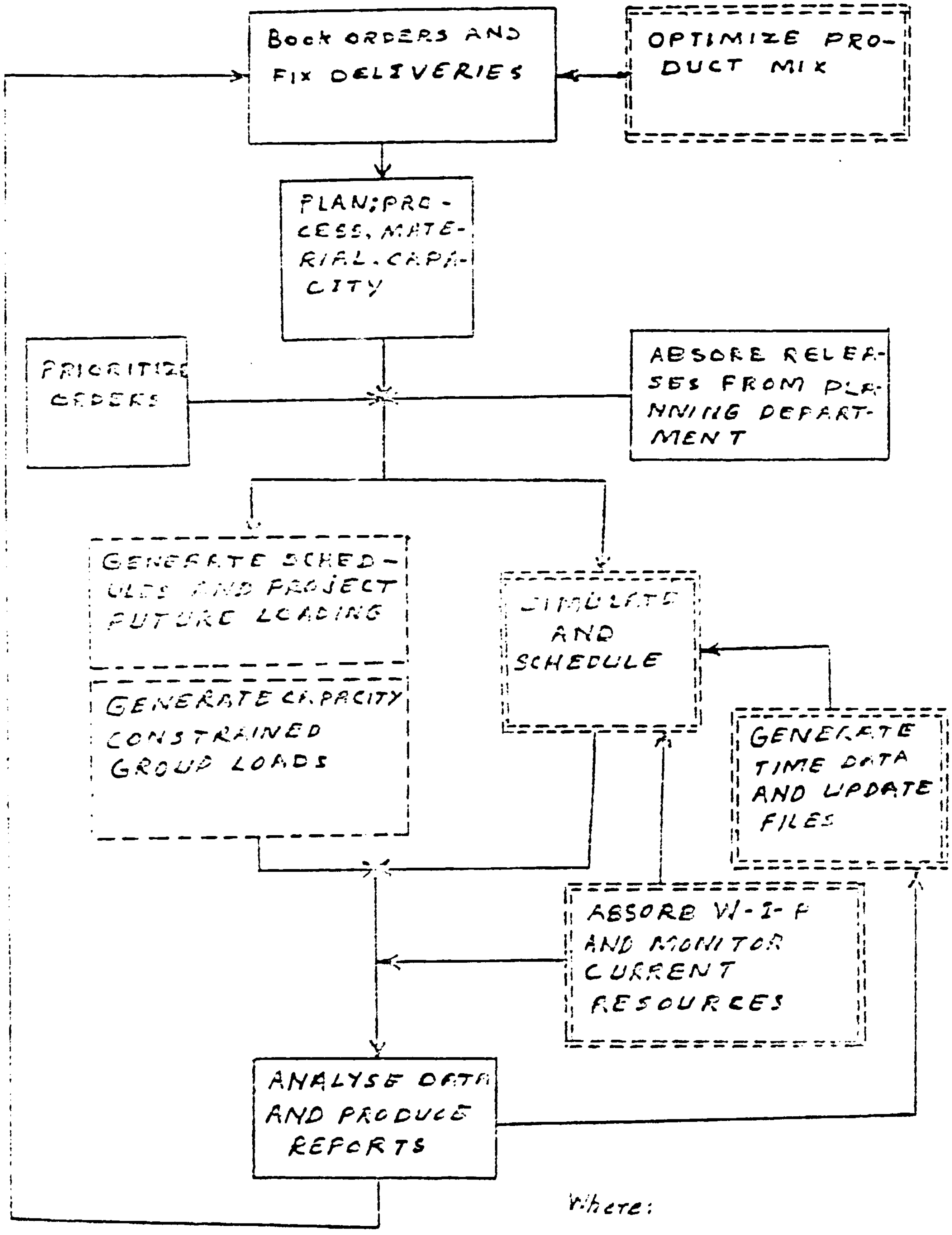


FIG.1 PPC SYSTEM

Where:

---- OLD FUNCTIONS

===== NEW FUNCTIONS

—— UN CHANGED FUNCTIONS

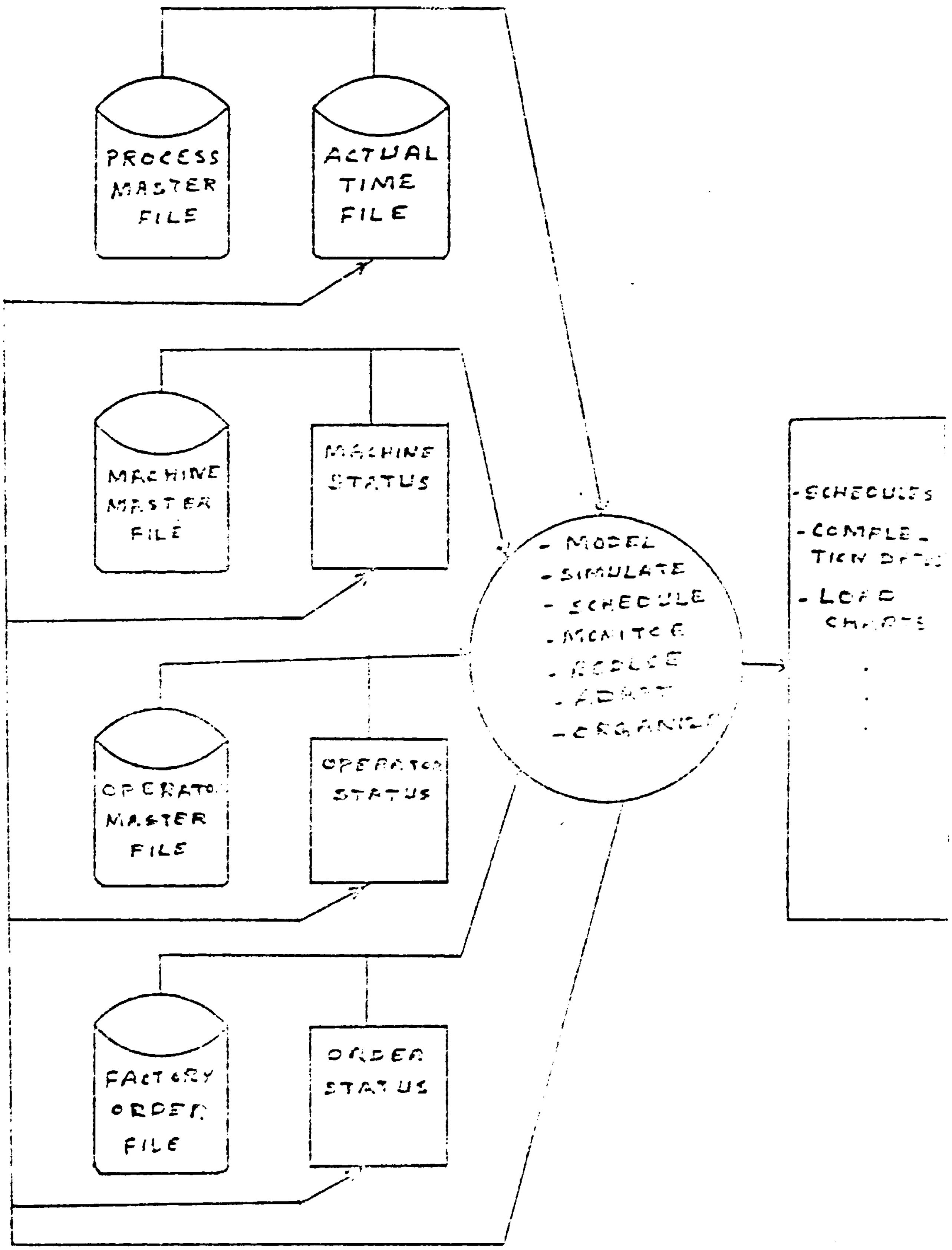


FIG.2 PFC STRUCTURE

III. BASELINE DESIGN

The approach is to realize an adaptive shop-floor controller using simulation, dispatching rules and operation heuristics for designing almost real-time schedules. This design is justified and discussed in what follows.

Driving and controlling operations effectively on a manufacturing shop-floor requires the development and application of a job sequencing strategy satisfying some global production criterion. The sequencing of shop-floor operations are obtained from a scheduling algorithm and applied dynamically as events occur in real-time.

Scheduling of shop-floor operations has traditionally been formulated as an optimization problem. Classical static or dynamic scheduling techniques compute a global performance index (e.g. mean flow-time) or implement a local optimization (non-delay schedule generation). In all cases the optimization strategy may be exact (optimal) or approximate (truncated) and uses search techniques such as: branch-and-bound algorithms, linear integer or dynamic programming as shown in figure 3. Because of the enumerative nature of these mathematical methods, the generation of constrained n/m job-shop schedules results in np-complete problems and for any practical situation computing times are too long to allow for any rescheduling due to changes in values for n, m or parameters to be performed in real-time, so that events are monitored and controlled as they occur.

Event randomness in industrial job-shop environments can be incorporated in an adaptive scheduling strategy based on heuristics (common sense) such as: dispatching rules, simulation, expert systems or real-time emulation (hierarchical or distributed) as shown previously in figure 3. These approaches do not guarantee a global performance (no optimization criterion exists). The expert system approach (based on a directed search and a knowledge representation of artificial intelligence), developped at Carnegie-Mellon University, produces a pseudo-optimal schedule with ability to react in almost preal-time to production perturbations. The other approaches generate different non-delay schedules (not necessarily optimal) but react in real-time to the manufacturing environment.

In this project an on-line scheduling strategy is developped. It is based on a simulation of the job-shop environment, combined with different dispatching rules for every machine-tool(no dispatching rule is best at all times) as shown in figure 4. Results of the simulation are grouped and displayed for evaluation. The acceptance or rejection of the suggested operational schedule (simulation) is based on explicit constraints (deadlines, etc.) and/or experience (possible improvments). While simulation results are evaluated, manufacturing operations are monitored in real-time and the production environment and parameter values are updated [2,3].

When rescheduling is required or desirable (inacceptable delays, change of configuration, better overall performance possibility, rerouting necessity, etc.), the shop-floor parameters are modified interactively and according to production heuristics then simulation is initiated again. Production heuristics can be formulated as simple decision rules (IF........ THEN.....) and will be established off-line using the experience of the foreman. The updating of the data required by these decision rules is performed interactively in real-time.

This design of on-line shop-floor driving and control with foreman in the loop and feedback heuristics implemented elsewhere and could result in a CMTI package product.

The software will be developed entirely in Fortran on a micro-processor (Vax) and will use the existing job-shop modelling and simulation capabilities of CMTI. It is estimated that the detailed design can be completed in the remaining two months of my assignment and that the coding and testing phases will require four programmers and two analyst respectively over two periods of three months.

If this software is designed in a top-down structured manner and developed in modular layers, it can be integrated in a more general package by adding a flexible cell design capability using different heuristic decision rules (part mix selection, machine-tool evaluation, cell configuration, etc.).

The heuristic decision rules can later be incorporated more formally in a knowledge base system with a proper meta-language and inference engine, or even in an expert system schell with reasoning and learning capabilities.

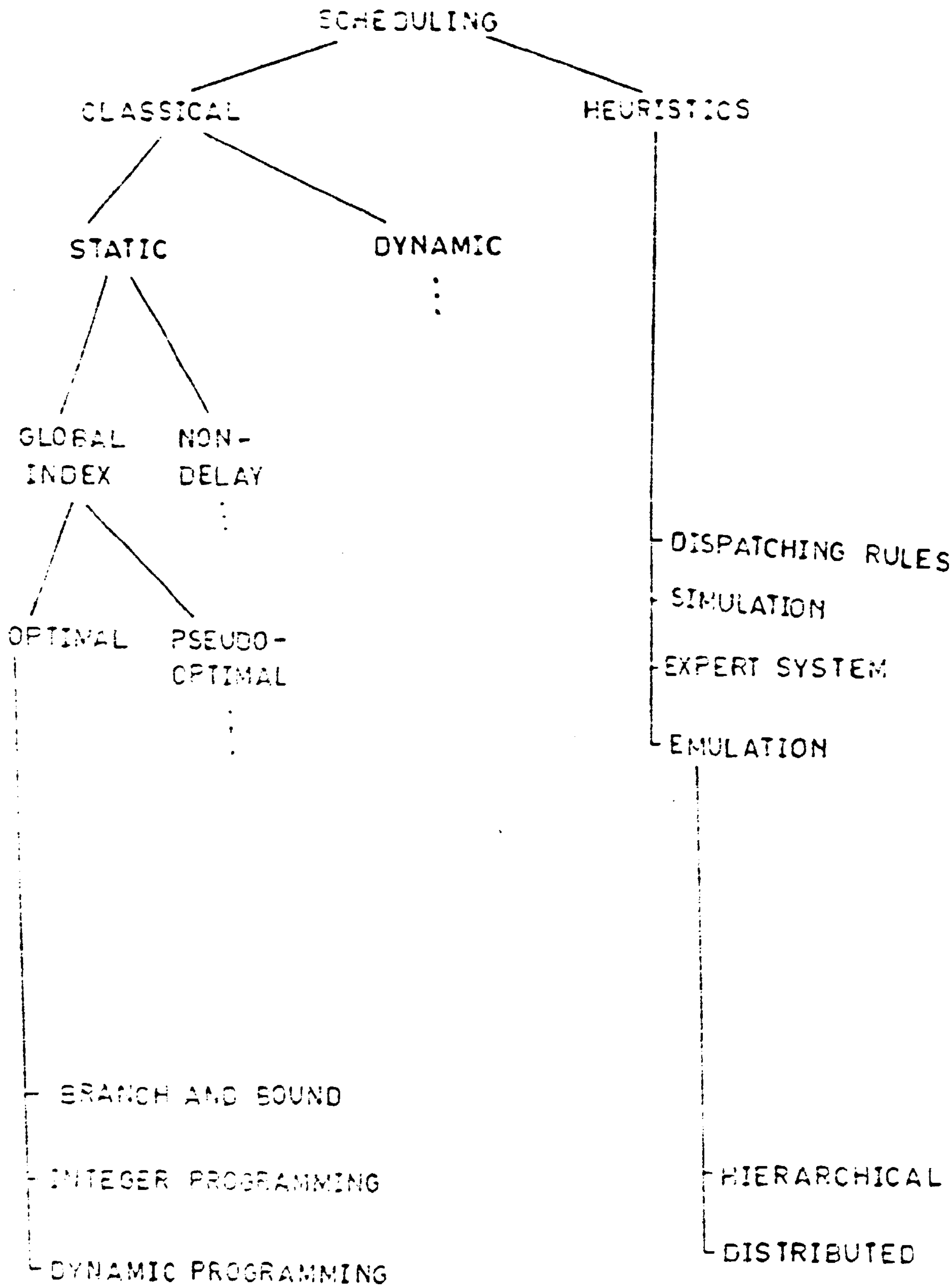


FIG.3 SCHEDULING METHODS

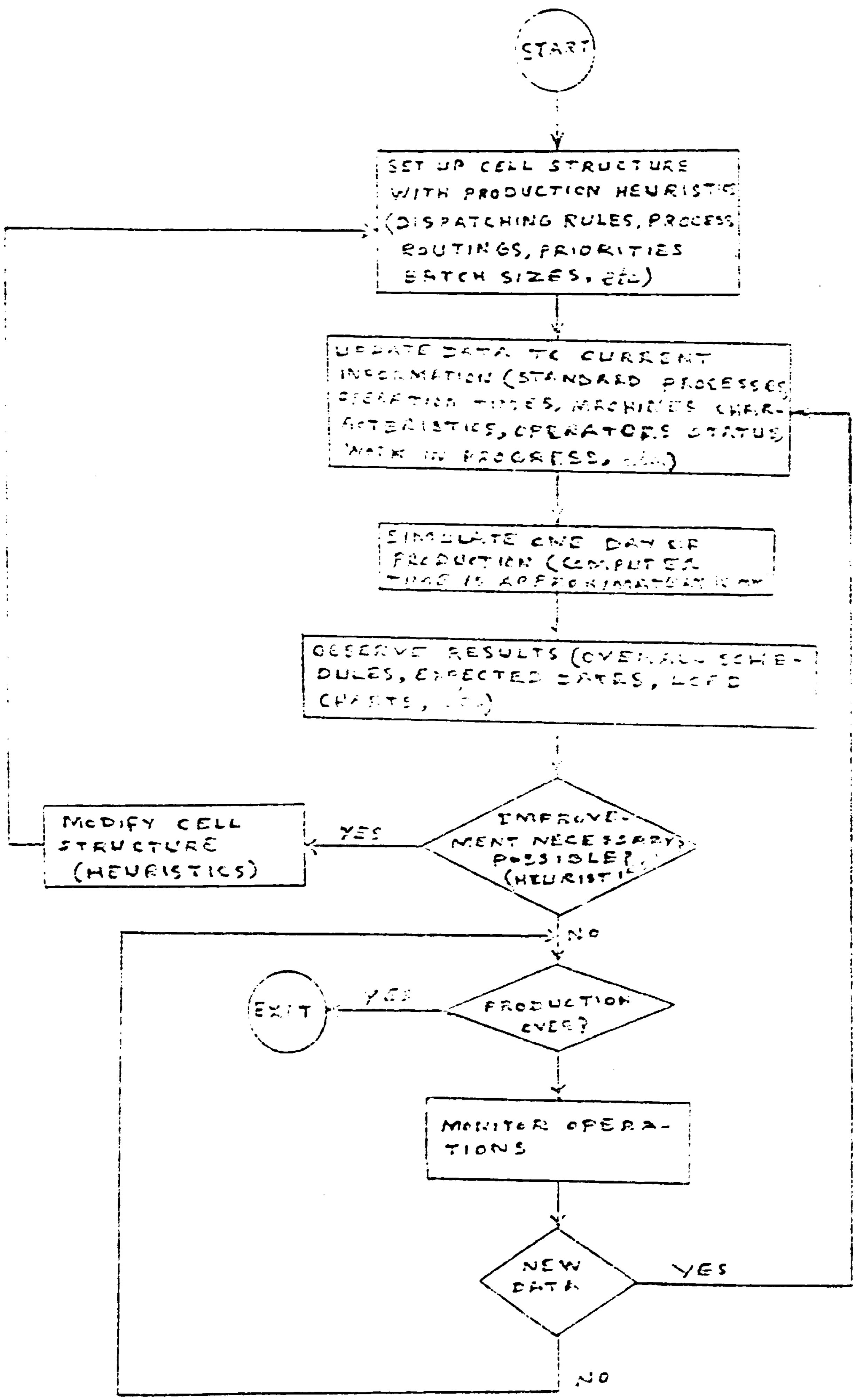


FIG.4 ON-LINE SCHEDULING

IV. DETAILED SPECIFICATION

In order to achieve the software engineering goals of flexibility, expandability and ease-of-use, the input, process and output functions of figure 2 are designed in a top-down fashion and developed in modular, menu-driven interactive manner. Since the system will be integrated in a real-time environment, the testing will also be conducted in a top-down exhaustive manner.

The detailed specification of the different functions of this software package are described using data-flow diagrams (an upgraded state transition technique).

These data-flow diagrams interface to the data structures of figure 2, and implement the overall flow-chart of figure 4. They will be completed by system level flow-charts or pseudo-code whenever necessary. A description of these functions is given next.

A. Shop-Floor Controller: This module performs the highest level function of choosing among monitoring events, inputing data, sequencing operations, organizing production, and outputing results as shown in figure 5. When invoked using the name "floor", the program displays a menu of options (monitor, input, operate, produce, output and exit) and waits for an answer, the user then chooses an option and initiates the corresponding module. If no appropriate choice is made an error is printed and the session is restarted (continued). Such logic may be formulated by the flow-chart of figure 6. It is then simple to translate this flow-chart (or an equivalent pseudo-code) to the target language chosen (in this case Fortran).

B. Event Monitoring: This function consists of simple updating of the following files:

- Actual Times
- Machine Status
- Operator Status
- Order Status

The data is entered by the operators. A data-flow diagram for this transaction is shown in figure 7.

C. Input Processing: Similarly a data-flow diagram for the interactive menu-driven input processing is given in figure 8. This module manipulates the work files and data of the factory and produces, by adding, deleting, changing, restoring or examining records, an active file for the show-floor.

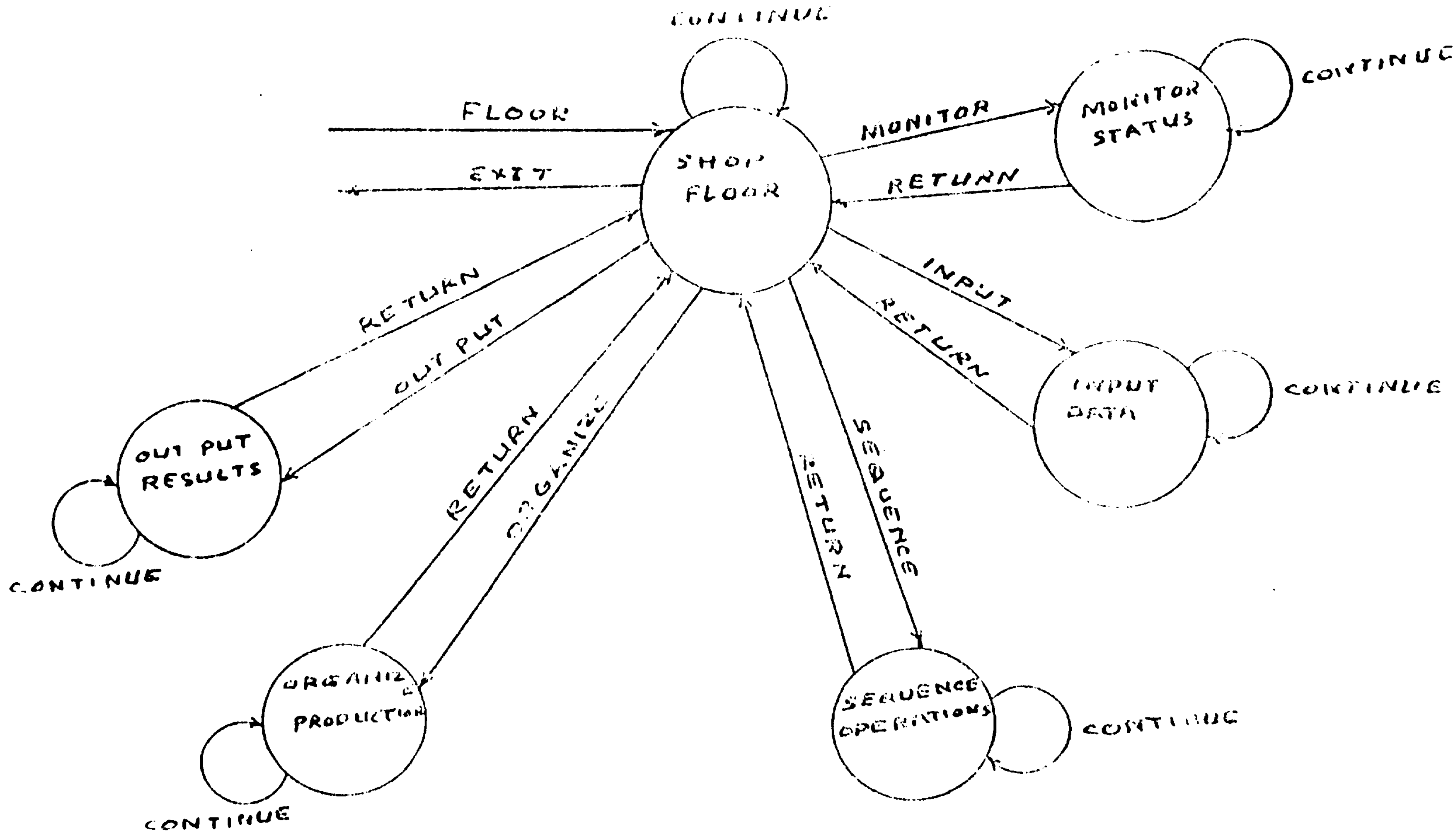


FIG.5 SHOP FLOOR CONTROLLER MODULE

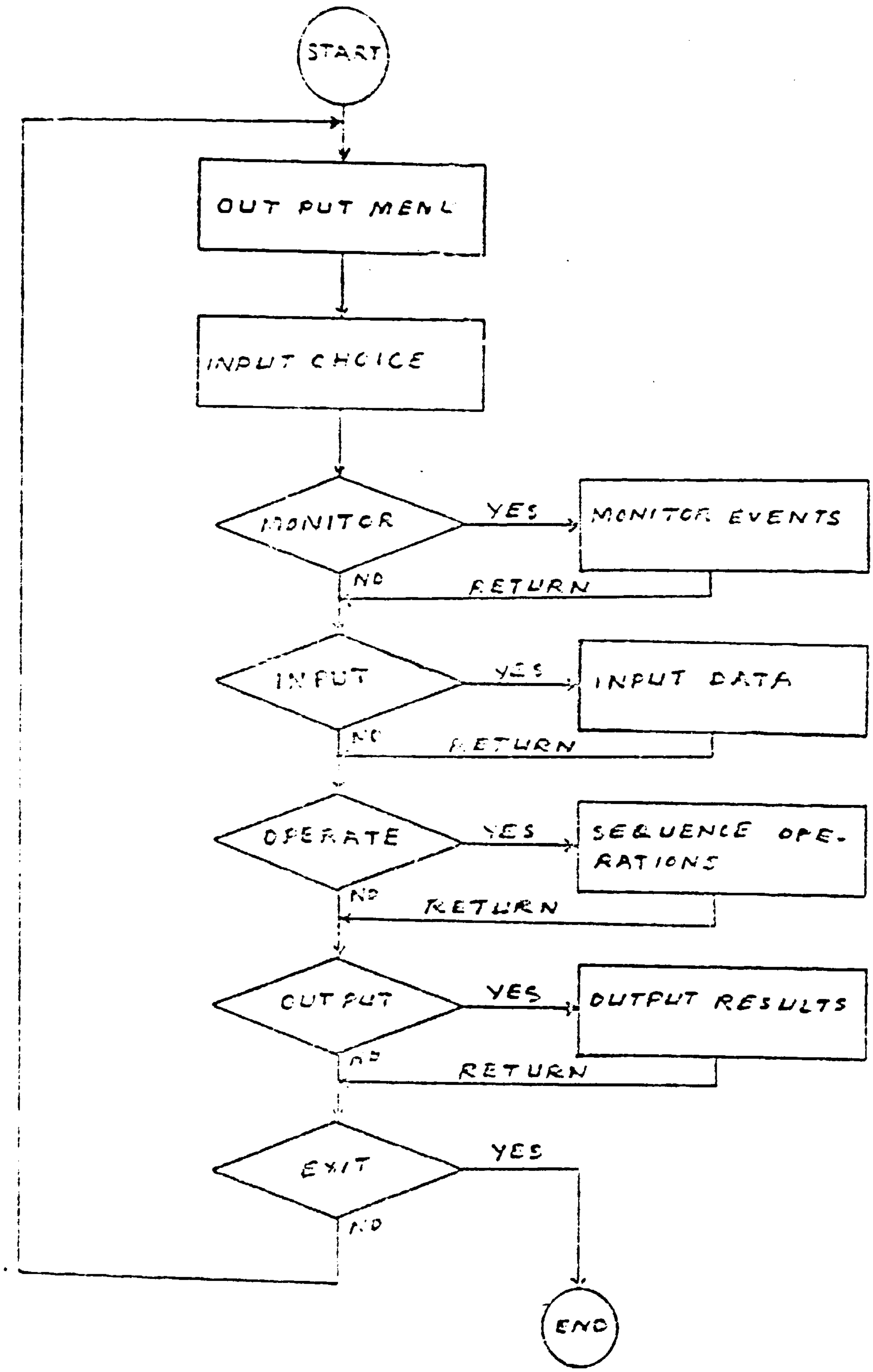


FIG.8 SHOP FLOOR FLOW CHART

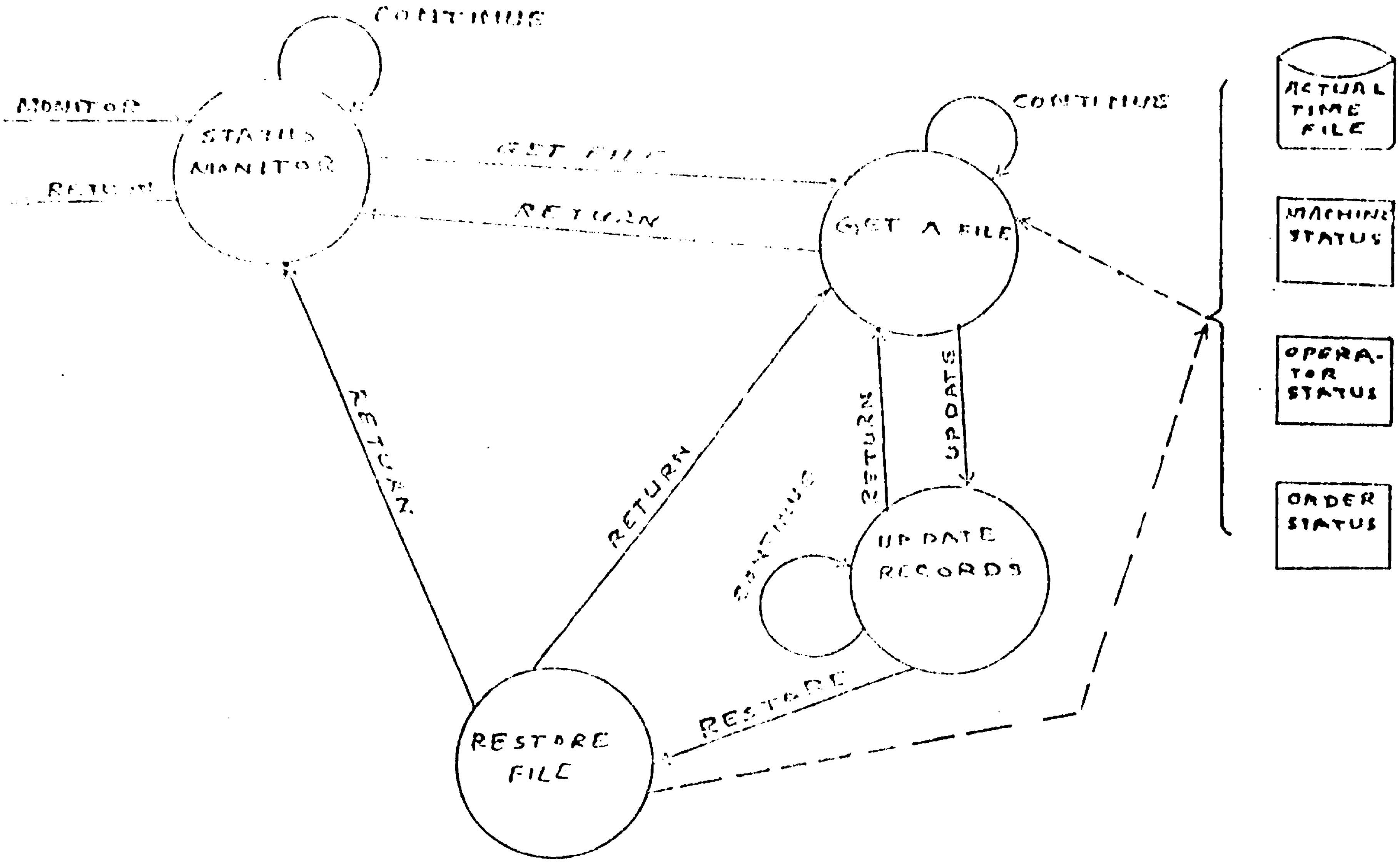


FIG. 7 EVENT MONITORING MODULE

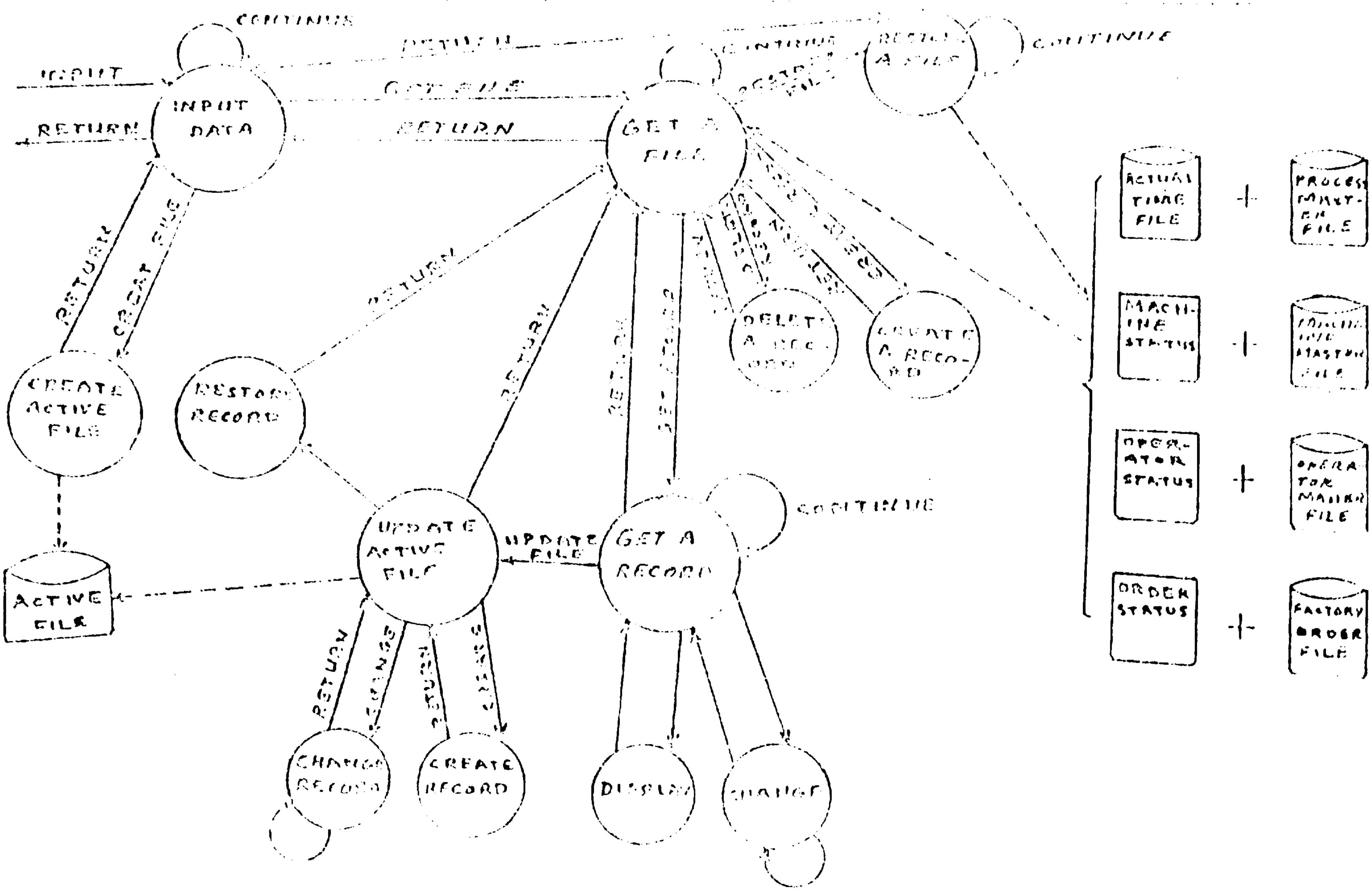


FIG 8 INPUT DATA PROCESSING MODULE

The specification of these record manipulating basic functions is not presented at this stage, but will be menu-driven for ease of use by the operator. They consist of displaying and eventually modifying the data in the record.

D. Operation Sequencing : This module consists of the following functions (figure 9):

- Setup of the shop-floor manufacturing model
- Simulation of the operations in scaled time
- Reduction of the resulting data
- Evaluation of the results
- Reorganization of the configuration

The shop-floor configuration is established and a dispatching rule is assigned to evey machine-tool. This function, is interactive and will be specified using data-flow diagrams.

The simulation subroutine uses the defined shop-floor configuration and generates time histories of production. This subroutine is not interactive and is therefore specified using a flow-chart.

The resulting data from the simulation is then transformed to more compact graphs, tables, etc.; This subroutine is also specified using a flow-chart.

The evaluation (observation) of results and the reorganization of the configuration can be combined in a list of IF...... THEN ..... cases where the IF and THEN parts are subroutines, specified using flow-charts and representing the manufacturing heuristics.

E. Production Organizing : This module sets up a test database first, then chooses the tests cases (e.g. load balancing, batch spliting) in order to optimize production by selecting the appropriate types and mix of parts [10,11,12,13,14]. These batching and balancing functions must be used itteratively since interdependant (batching requires balancing and balancing may not be possible for a given batch). This module uses the operation sequencing module for evaluation of the results then accepts or rejects them. The corresponding subroutines are specified using flow-charts. The overall data-flow diagram for this module is shown in figure 10.

F. Output Results : The data-flow diagram for this interactive module is shown in figure 11. The differents files, simulation results, reduced data, static and dynamic snapshots and management reports are produced on request. These functions are specified using flow-charts.

G. Shop Model: This sub-module generates the manufacturing environment and initializes it. The state of the system and its dynamic properties are represented by the following [4]:

1. Shop image: This function models the elements of the shop-floor using entities (machines, batches,tools, etc.) and their attributes (availability,number in buffer,etc.). A detailed list of entities and attributes is given in table 2.

   In this model production is organized into orders, jobs, batches and parts. One day of production therefore may consist of several orders, orders are composed of one or several jobs, jobs consist of batches contained in pallets and batches are composed of one or several parts to be machined using fixtures. Jobs are of two different types: manufacturing requests (existing, new and cancellation) or maintenace tasks (preventive or service). Machines may also be of different types (machine-tools, carts or conveyors and inspection stations or operators).

2. Shop activities: This function creates discrete events (machine, queue) according to some dispatching rule (priority,first-come first- served,etc.) and affects the system image accordingly (subject to other contraints, e.g. process plans). A detailed definition of these dispatching rules is given in table 3 [5,6,14].

   In this approach, each machine is followed by queues (a queue identifies uniquely a part to be manufacutred) so that only two types of events can occur: loading a machine-too with a batch of parts or unloading the batch of parts into the corresponding queue upon machining completion.

   A data-flow diagram representing this sub-module is given in figure 12.

H. Operation Simulation: This function produces the behavior, in time, of the shop manufacturing. The program scans the representation of the system every time-interval, selects the events according to activities and tries to implement the status change in the system image. If the event cannot be immediatly executed it is recorded for a later trial. The outcome of the execution of events and other relevant statistics (Gantt chart, counts, occupency rates, etc.) are recorded. A flow-chart for this

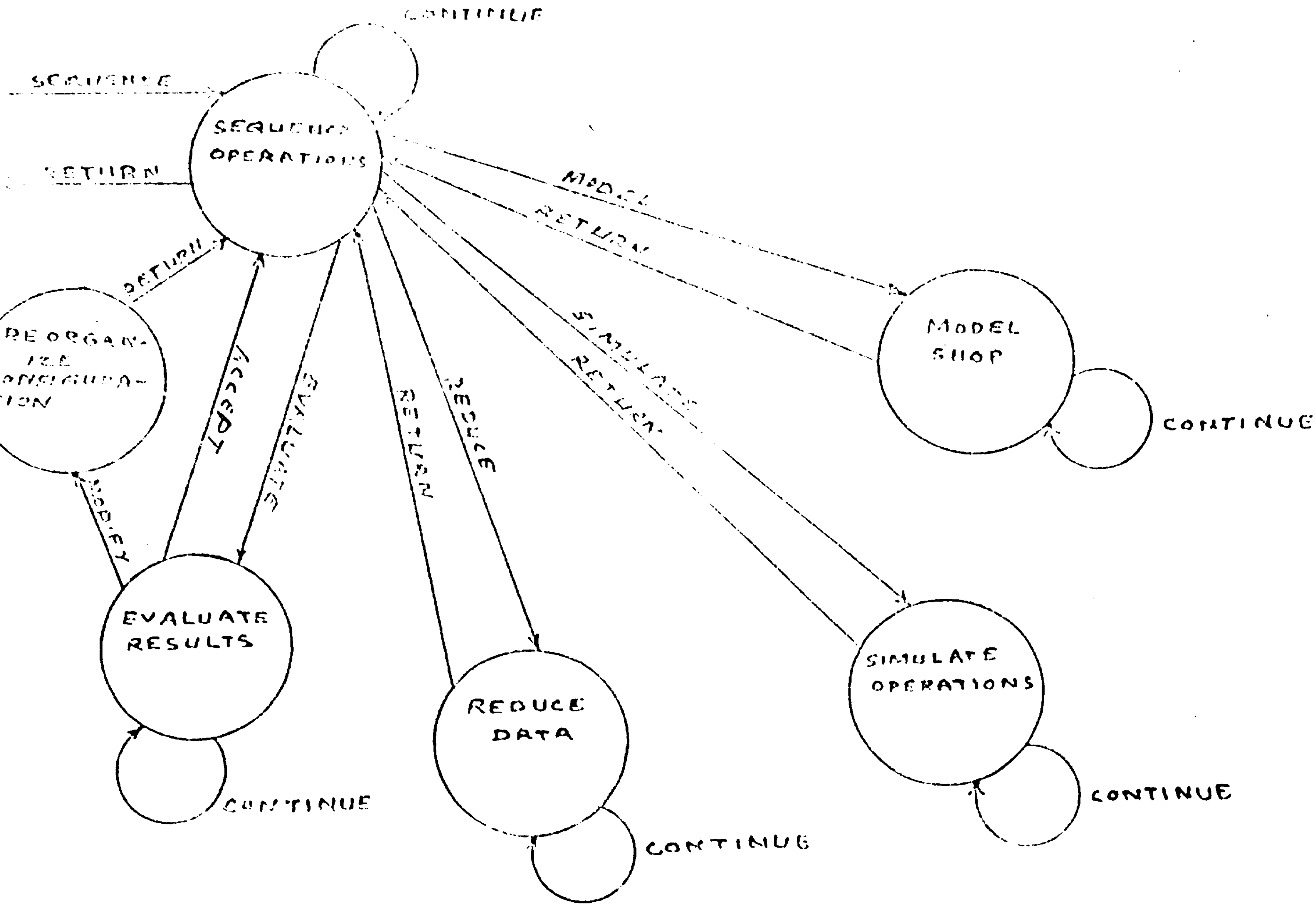


FIG.9 OPERATION SEQUENCING MODULE

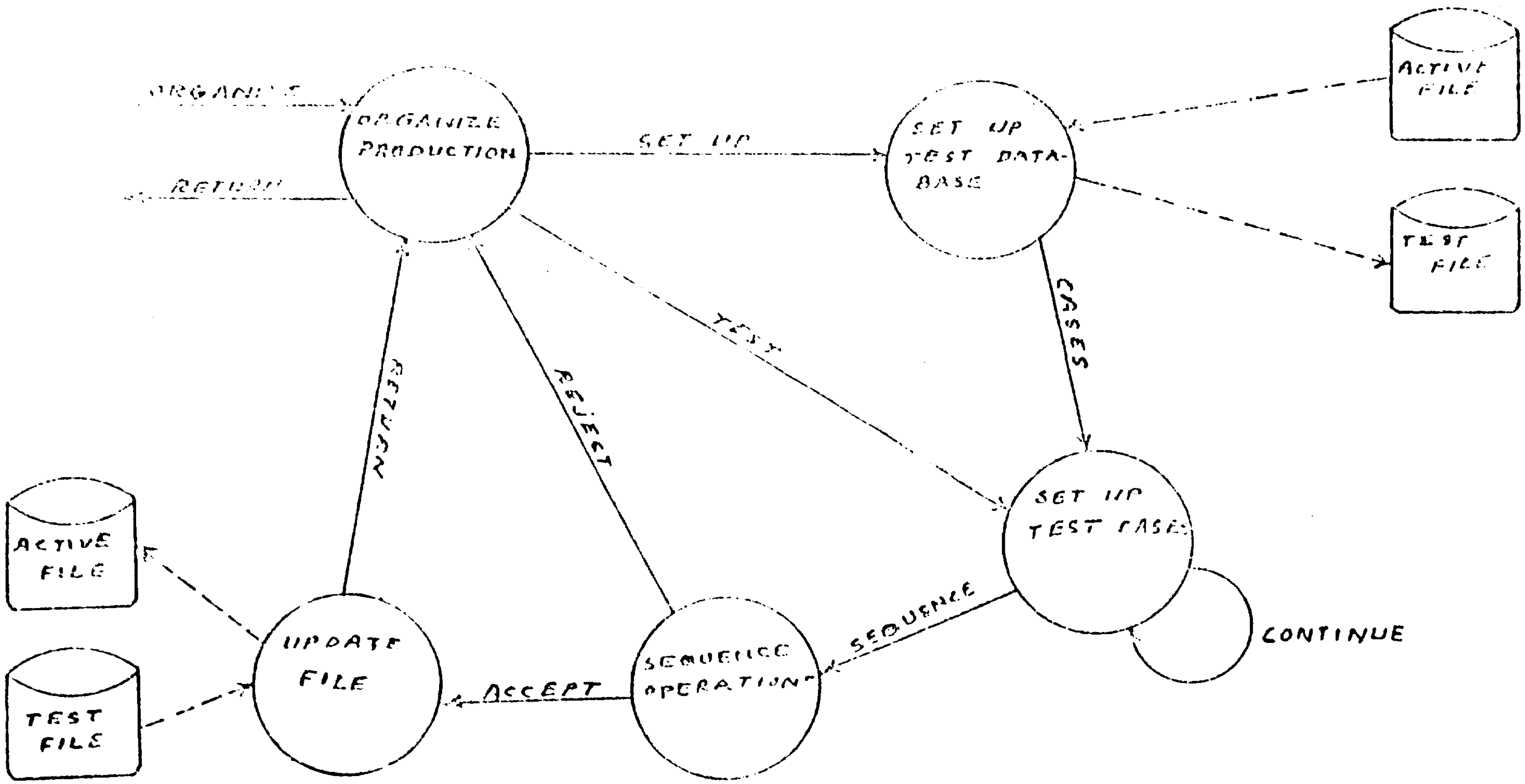


FIG.10 PRODUCTION ORGANIZING MODULE

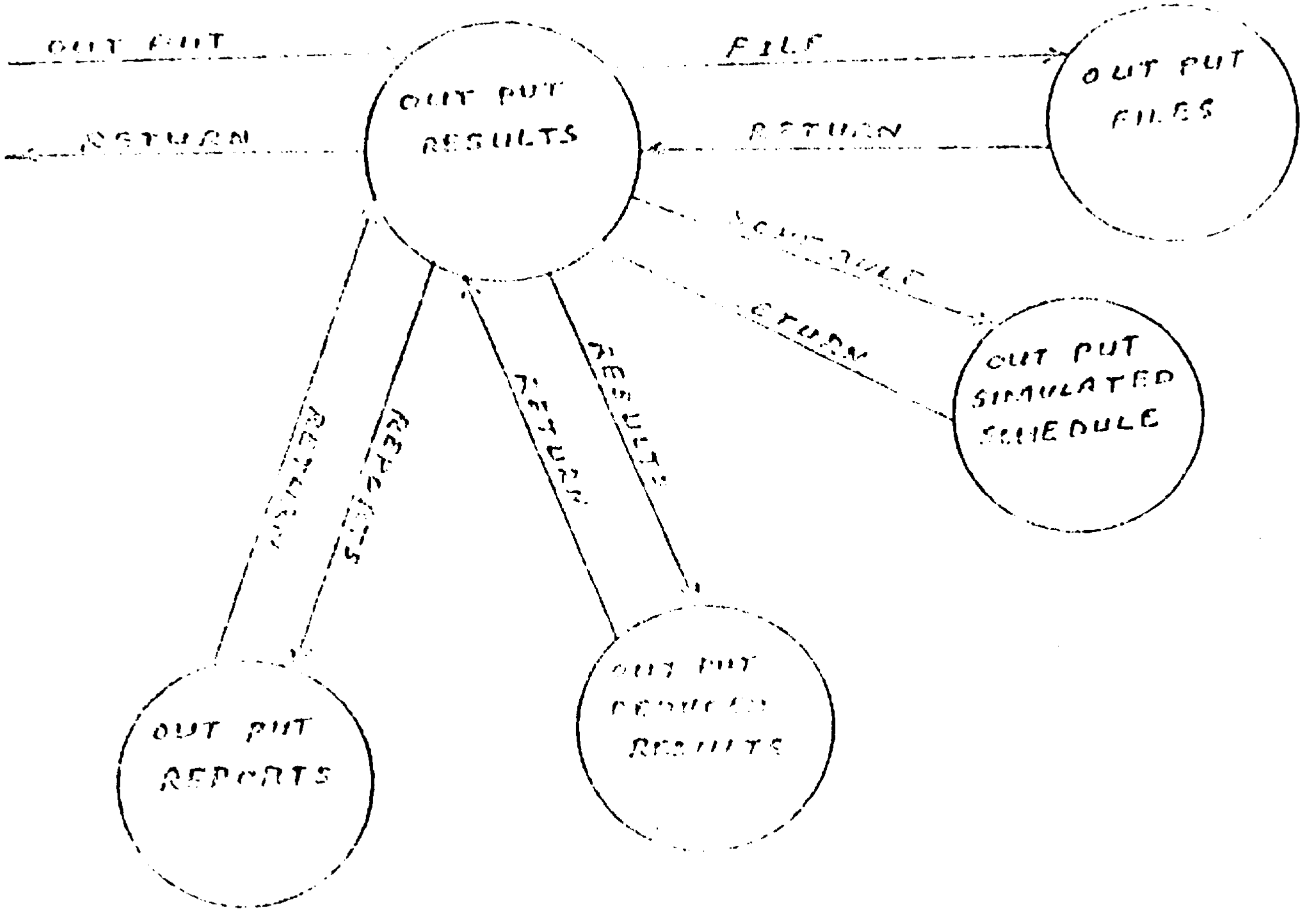


FIG.11 OUTPUT PROCESSING MODULE

sub-module is given in figure 13 [4]. This conditional event approach presents the risk of deadlocks (one event waiting for a condition to be satisfied by another event and vis-versa). This situation can be avoided by using a global (multi-valued) condition (semaphore) as shown in figures 14, 15, 16 and 17.

In this design, the model image is altered by recording changes in the attributes of the entities (machines, operators, queues, etc.) and time. The recording is performed on the production active file.

Choosing the simulation time-shop can be performed in several ways: small or based on experience or as large as possible and approximately equal to half the length of the shortest operation.

The outcome of the execution of events (Gantt chart) and other relevant statistics are recorded. A list of statiscal parameters useful for the evaluation of the performance is given below:

- Number of orders completed
- Number of jobs completed
- Number of manufacturing requests completed
- Number of manufacturing requests newly arrived
- Number of manufacturing requests cancelled
- Number of preventive maintenance requests completed
- Number of service maintenance requests completed
- Number of batches completed
- Number of parts completed
- Number of orders waiting in shop
- Number of jobs waiting in shop
- Number of batches waiting in shop
- Queue sizes
- Manufacturing start times
- Manufacturing completion times
- Idle times for machines
- Idle times for operators
- Preventive maintenance times
- Service maintenance times
- Operators absenteism
- Machine down times

Table 2 Entities and Attributes

Entities	Attributes
Orders	- Code - Number of jobs - Delivery schedule - status
Jobs - requests - Existing	- Code

Table 2 (Continued)

- New	- Number of batches
- Cancellation	- Delivery schedule
- Maintenance- Preventive	- Process sheet number
- Service	- Status
Batches	- Code
	- Sizes
	- Pallets
	- Fixtures
	- Transfer time
	- Minimum batch quantity
	- Number of parts
	- Status
Machines - Machine tools	- Code
- Carts (no tools)	- Number
- Inspection stations	- Capacity
	- Capability
	- Number of shifts
	- Location
	- Buffers
	- Dispatching rule
	- Status
Buffers	- Code
	- Size
	- Status
Operators	- Code
	- Number
	- Preferred machines
	- Skills
	- Performance index
	- Status
Tools	- Code
	- Operation number
	- Machine number
	- Status
Process	- Code
	- Number of operations
	- Operation details (sequence, tool code, etc.)
	- Set up time
	- Operation time
	- Inter transfer time
Time	- Calender
	- Holidays
	- Shifts
	- Hours

Table 3 Dispatching rules

Weighted priority [5]	$P_i = P_{e,i} \sum_{n=1}^{4} W_n P_{n,i}$; where $P_{e,i}$ = external priority for the ith job $P_{1,i}$ = is given by Caroll's rule [5] $P_{2,i}$ = remaining processing time/ current processing time $P_{3,i}$ = estimated size of the next queue W_1, W_2, W_3, W_4 = weighting factors (may be obtaining by optimization [5] or heuristically)
Urgency number [14]	$N = \frac{RT-(RPT+RQT)}{OPS}$; where RT = remaining time to due date RPT = remaining processing time RQT = remaining expected queue time OPS = remaining number of operations
First-come first-served[6]	R_{ij} = time at which the ith job becomes ready for the next imminent jth operation
Random [6]	Y_{ij} = uniformly distributed value for the next imminent jth operation of the ith job
Shortest processing time [6]	P_{ij} = processing time of the next imminent jth operation for the ith job
Greatest total work [6]	$\sum_{j=1}^{g_i} p_{ij}$; where g_i = total number of operations for the ith job
Earliest due date [6]	d_i = due date for the ith job
Least slack time remaining [6]	$d_i - t - \sum_{k=j}^{g_i} P_{i,k}$; where t= time at which a machine is selected j= the next imminent operation for the ith job
Fewest operations remaining [6]	$g_i - j + 1$; where
Least work remaining [6]	$\sum_{k=j}^{g_i} P_{i,k}$;

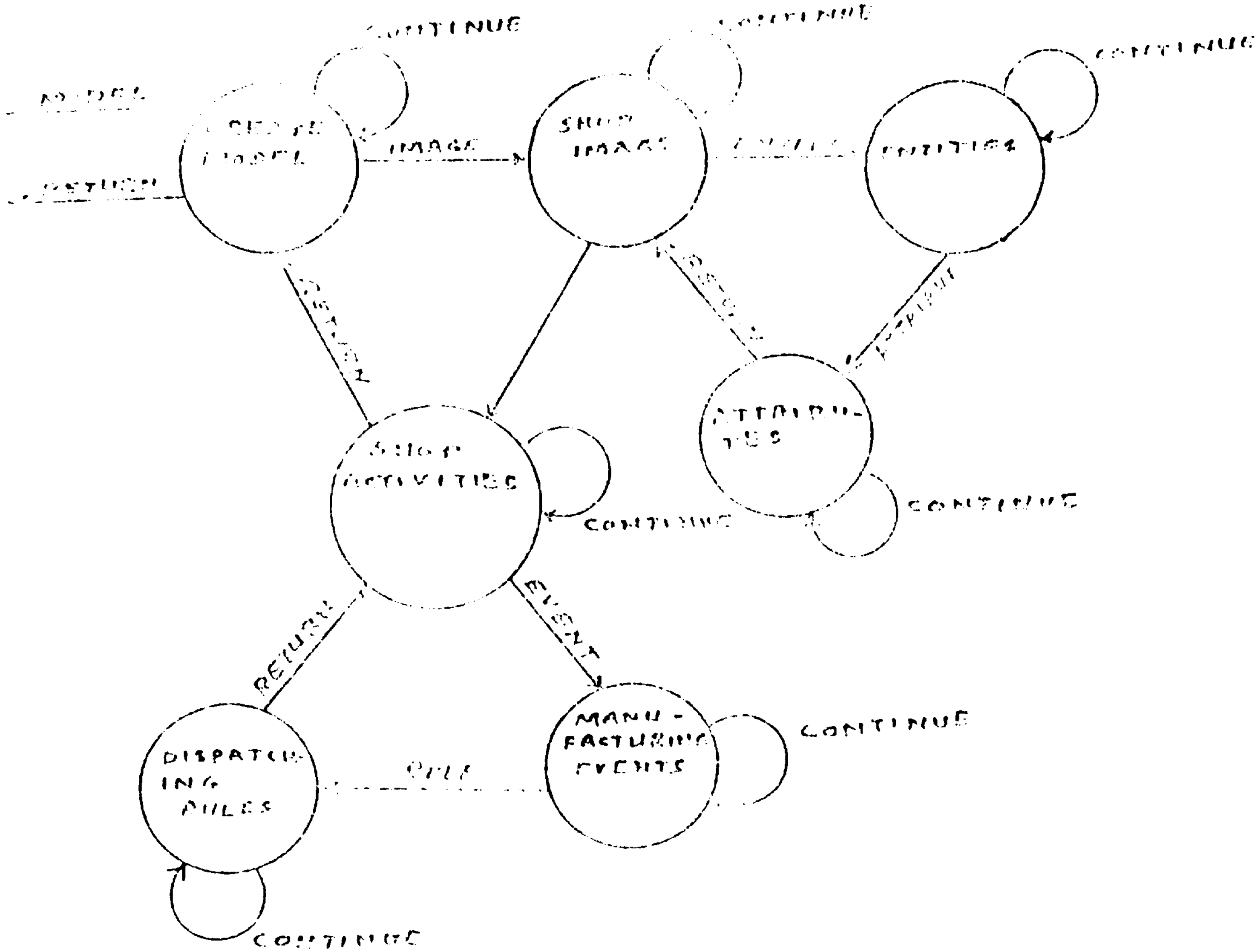


FIG. 12 SHOP MODEL

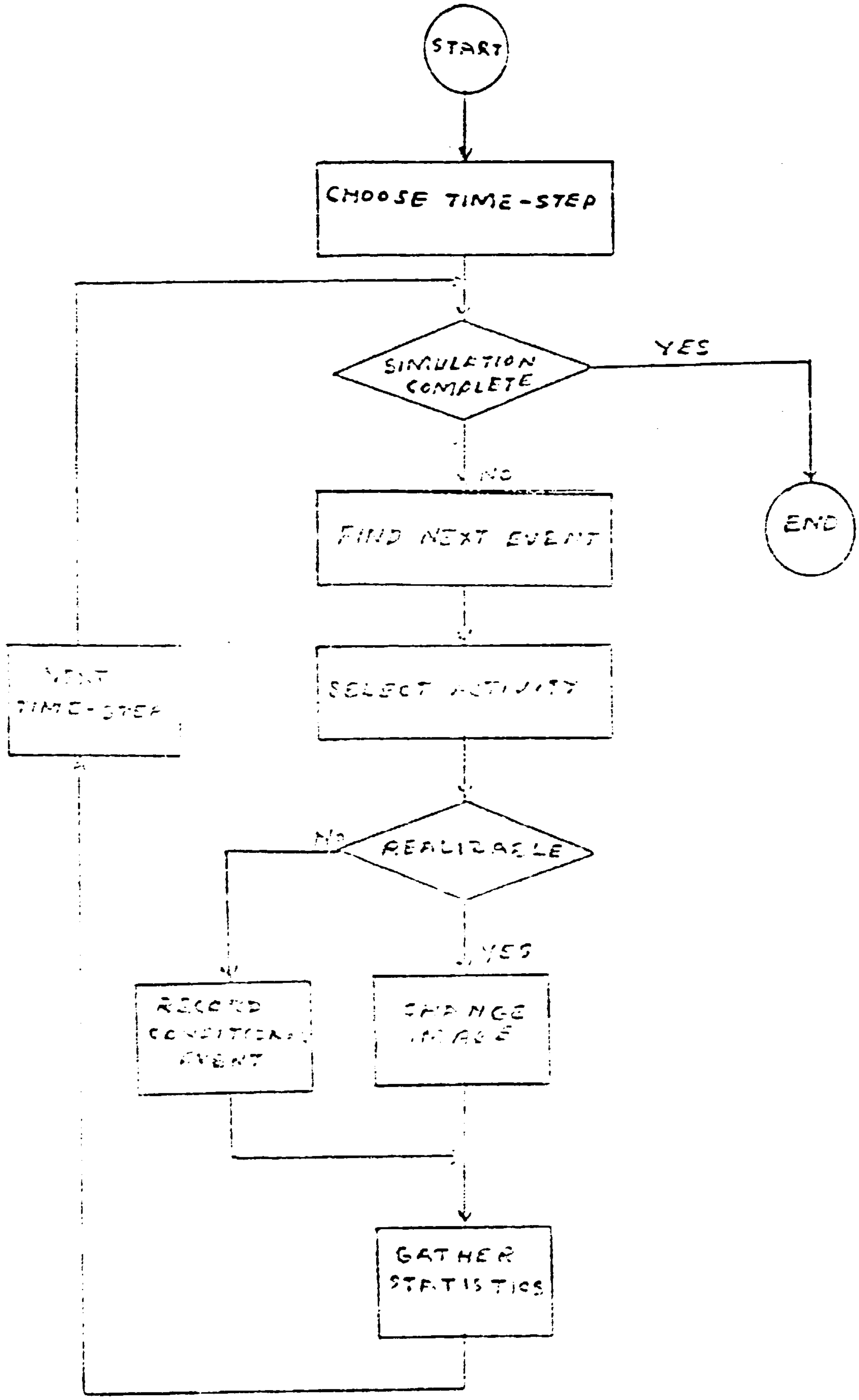


FIG. 13 OPERATION SIMULATION

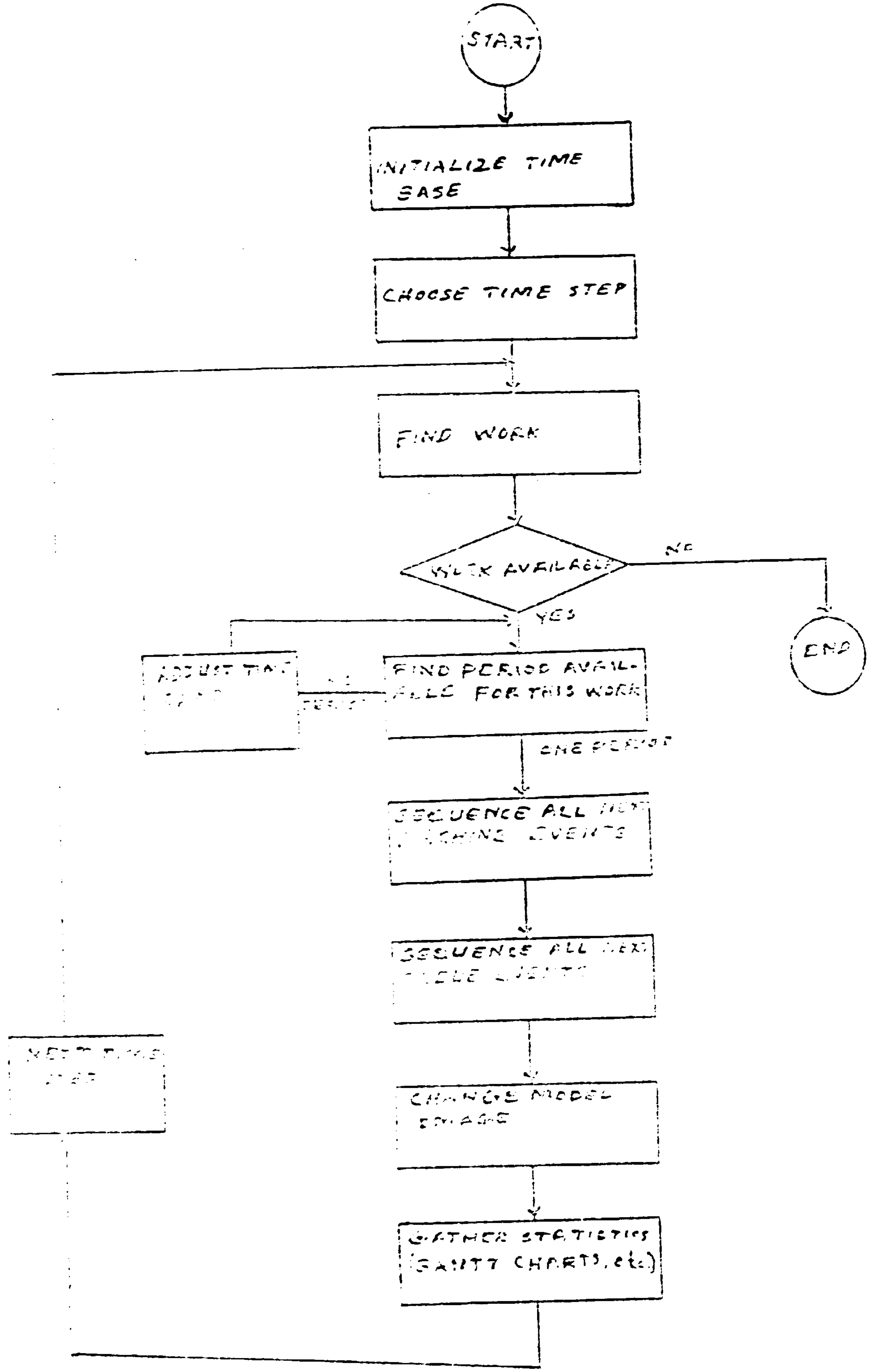


FIG.14 DEADLOCK FREE SIMULATION

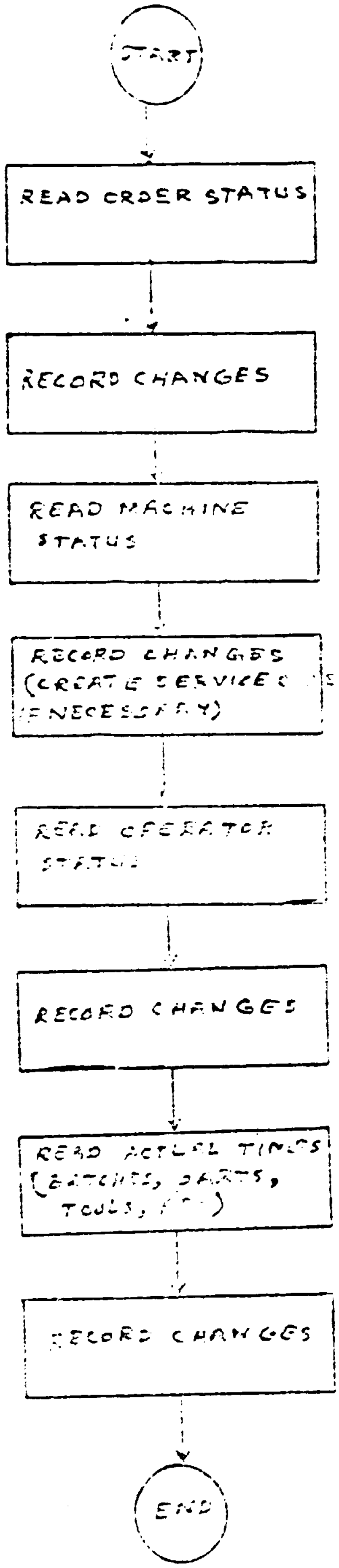


FIG 15 FIND WORK SUBROUTINE

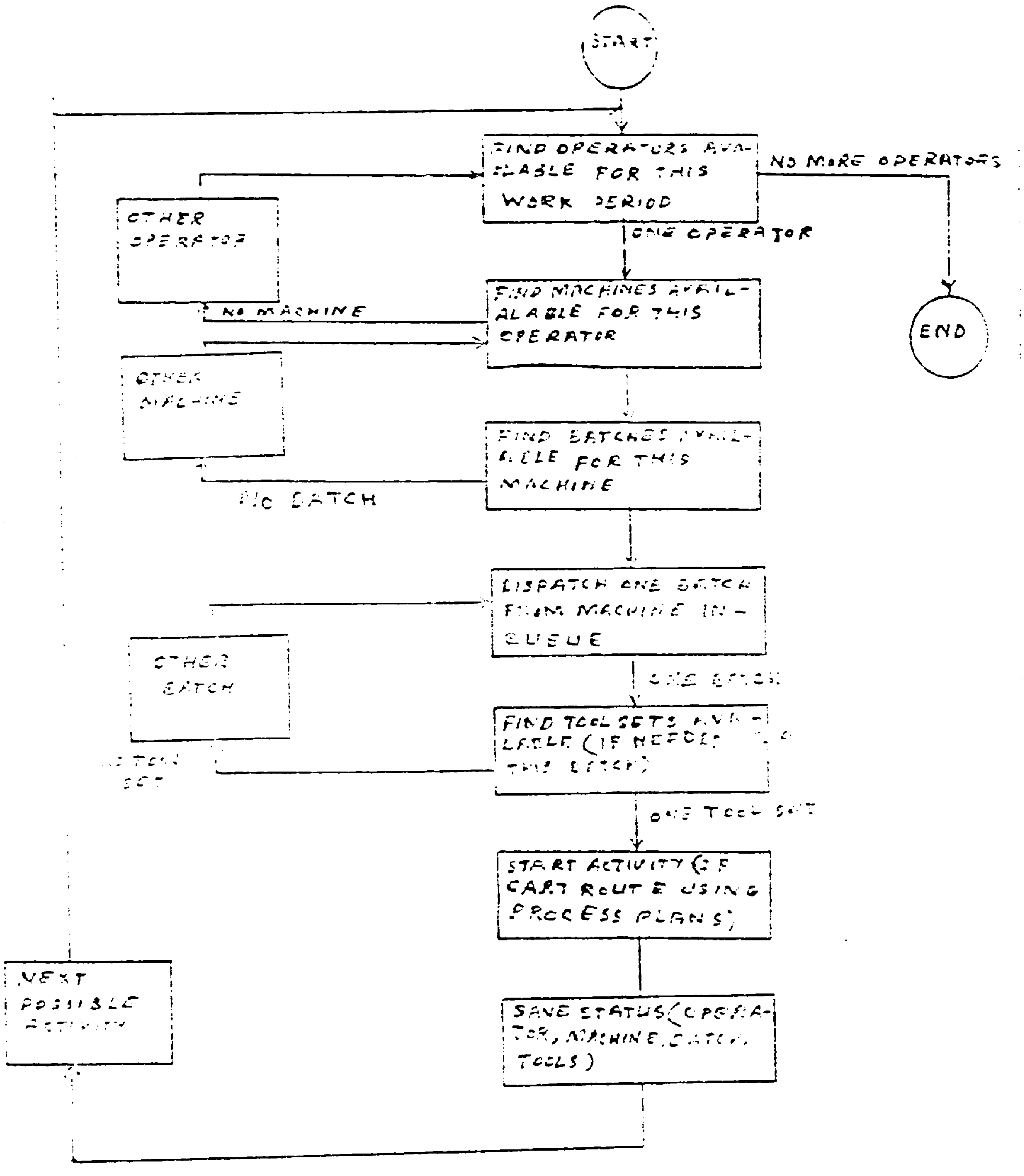
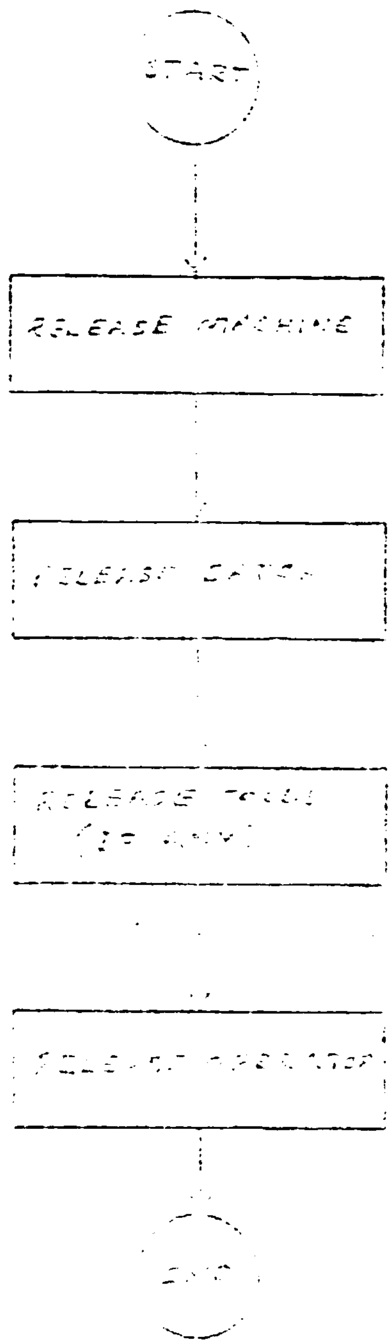


FIG.16 MACHINE EVENT SUBROUTINE


START
RELEASE MACHINE

[illegible]

I. Data Reduction : The gathered statistics must be reduced to meaningful performance evaluation criteria so that appropriate corrective action can be decided upon if necessary. A list of data reduction parameters are given below [7,8,9]:

- Percentage of orders completed
- Percentage of jobs completed
- Percentage of orders behind schedule
- Percentage of jobs behind schedule
- Percentage of machines utilization
- Percentage of operators utilization
- Percentage of queues utilization
- Work in progress levels
- Job sequences for machines
- Average throughput time for jobs
- Average waiting time for jobs
- Bar chart loadings for machines
- Bar chart loadings for operators
- Delivery dates forecast

J. Result Evaluation :The evaluation of results and alteration of the manufacturing configuration are grouped in two sets of subroutines accessible to the shop foreman who chooses the performance criterion to be observed (displayed) and the remedial action to be taken. The functions of these subroutines are listed below [7,8,9,10]:

1.Performance:
- Jobs behind schedule
- Jobs completed ahead of schedule
- Orders cancelled
- New orders
- Machines underutilized
- Machines overutilized
- Machines unavailable
- Operators underutilized
- Operators overutilized
- Operators unavailable
- Throughput time more than standard
- Work in progress more than standard

2.Actions :
- Allocation of more machines
- Allocation of more operators
- Choose alternate routes
- Choose alternate work centre
- Reallocation of machines
- Reallocation of operators
- Subcontract jobs
- Hold back jobs
- Choose alternate dispatching rules
- Change job priorities
- Allocate new jobs

K. Batch Splitting: The main criterion the batching procedure must satisfy is to minimize the total time for processing all parts. This is formalized as follows:

- Minimization of the number of batches required to process all parts (this minimizes the time associated with batch changeovers).

- Maximization of the average utilization over all machines (this minimizes the time required to work through an individual batch).

The second of these issues suggests the need for balancing the work evenly among the machines.

L. Line Balancing: The main issues associated with line balancing are :

- Minimization of the differences in time required for workload assigned to different machines.

- Quarantee that all the work for each batch is assigned to some machine.

The second issue brings up the possibility of conflict since it may be impossible to assign the work prescribed for a given batch. If balancing fails batching must be tried again until a trade-off situation is reached.

V. REMAINING TASKS

Because of the lack of appropriate documentation, the modules on batch splitting and line balancing have not been designed in as much detail as necessary. These tasks will be completed by one of the senior analysts taking part in the project.

The previously developed data-flow diagrams and flow-charts and the finalized flow-charts for the incompletly designed modules need to be augmented with the required data, in order for the different modules to perform their functions.

The coding and documenting of these modules and the corresponding data structures will require four programmers (analysts) for a period of three months (the batch splitting and line balancing modules will require and extra month for completing the design). The documentation will also include the user manuals, the programming manuals, the test results, etc. as described in the project management seminar.

Each module must be tested exhaustively and separately, then in integration with the other related modules. A global test run with all modules is necessary.This phase will require two analysts for a period of three months.

The installation of the packages will consists of transporting the software, its documentation and its test cases to the appropriate hardware/firmware (in this case a micro-Vax system, to be acquired).

The names of the project analysts and programmers and the their assigments is given in annex 1.

CONCLUSIONS AND RECOMENDATIONS

1. The two research and development projects were designed satisfactorily within the time planned (except for two modules).

2. The seminars suggested were not all given. An outline for the software management seminar is given in annex 2[1]. The elaboration of three other seminars (distributed systems[15], computer networks[16] , intelligent systems [17]) has been started in collaboration with three CMTI engineers. When completed the seminars will be given by the CMTI engineers. Outlines of these seminars are given in annexes 3,4 and 5 respectively.

3. Two other seminars should be initiated in the areas of robotics [18] and databases [19] (necessary in computer integrated manufacturing) and guidelines are given in annexes 6 and 7.

4. Contacting research institutions in the U.S, Canada, France and Switzerland for training of CMTI scientists, acquisition of software packages and possible colloboration arrangements were also initiated. Because of time, not many responses were received (it is possible to arrange for training of two scientists at the National Research Council of Canada) . Therefore the procurement types projects were not completed.

5. The documentation for research and solution tools (mathematics, logic, etc.) is not adequate (not many recent books, reports, papers, proceedings) and should be augmented. A partial list of topics important in computer manufacturing is given in annex 8. A more detailled list of specific books, reports etc., will sent upon my return to Canada. Some of the documentation I brought was made available to CMTI. A list of these documents is given in annex 9.

6. The planned hardware/firmware application software is adequate for the different projects in progress. Other possible research and development projects could be initiated and a partial list of such projects is given in annex 10.

REFERENCES

[1] Metzger, P.W., Managing a Programming Project, 2nd Edition, Prentice - Hall, Englewood Cliffs, 1981.

[2] Pera, Production Systems, Report No. 309, November 1976.

[3] NTIS, Flexible Manufacturing System Handbook, Report No. AD/A127928, Feb. 1983.

[4] Gordon.G, System Simulation, 2nd Edition, Prentice-Hall,New-Delhi, 1978.

[5] Filip, F.G., Neagu, G., Donciulescu, D.A., Job-Shop Scheduling Optimization in Real-Time Production Control, Computers in Industry, Vol.4 , No.4. Dec. 1983.

[6] Conway, R.W.,Maxwell, W.L.,Miller, L.W., Theory of Schudling Addisson-Wesley Reading, 1967.

[7] Pera, Scheduling in a Subcontract Machine Shop, Report No.355, April 1981.

[8] Pera, Real-Time Scheduling, Report No. 367, June 1982.

[9] Pera, Production Systems, Report No. 379, September 1983.

[10] NTIS, Flexible Manufacturing System Handbook, Report No. AD/A127 929, February 1983.

[11] NTIS, Flexible Manufacturing System Handbook, Report No. AD/A127 930, February 1983.

[12] Kochhar, A.K., Development of Computer-Based Production Systems, Edward Arnold, London, 1979.

[13] Bedworth, D.D., Bailey, J.E., Integrated Production Control Systems, John Wiley ,New York, 1982.

[14] Colley, J.L., Landel, R.D., Fair, R.R., Production Operations Planning and Control, Holden-Day, San Francisco, 1971.

[15] Champine, G.A., Distributed Computer Systems, Allied Publications, New Delhi, 1983.

[16] Tanenbaum, A.S., Computer Networks, Prentice-Hall, New Delhi, 1981.

[17] Rich, E., Artifical Intelligence, McGraw-Hill, Singapore, 1983.

[18] Paul, R., Robot Manipulators, Manufacturing Technology, Naperville, 1981.

[19] Date, C.J., Introduction to Date Base Systems, 3rd Edition, Addisson-Wesley, Reading, 1981.

ANNEXES

Annex 1 - People and Assignments

Mr. Shanbhogue, H.B.V	: Coding, testing of model, simulator, monitor modules
Mr. Suresh, M.S.	: coding, testing of dispatching rules, input, output modules
Mr. Somashekar, B.S.	: coding, testing of data reduction, result evaluation modules
Mr. Subramanian, K.	: detail designing, coding, testing of batch splitting, line balancing modules
Mr. Srikantiah, M.S.	: intelligent system seminar
Mr. Ramaiah, V.S.	: distributed systems seminar
Mr. Shekar, M.	: computer networks seminar

Annex 2 - Project Management

Programmable development cycle

- Definition phase
- Design phase
- Programming phase
- System test phase
- Acceptance phase
- Installation phase

Project plan outline

- Phase plan
- Organization plan
- Test plan
- Change plan
- Documentation plan
- Reporting plan
- Operation plan
- Deliverable plan

Annex 3 - Distributed Systems

- Functional decomposition
- Hierarchical structuring
- Distributed realization
- Modelling
    - IDF 0,1,2
    - Petri nets
- Validation techniques
- Workstations

Annex 4 - Computer Networks

- Theory
  - Finite state machine
  - Petri nets
  - Others
- International Standards Organization (ISO)
- Open System Interconnection (OSI)
- Local Area Network (LAN)
- Manufacturing Automation Protocol (MAP)

Annex 5 - Intelligent Systems

- Languages
  - Lisp
  - Prolog
  - Others
- Artificial intelligence
  - Searching
  - Knowledge representation
  - Natural language processing
  - Theorem proving
- Expert systems
  - Paradigms
  - Inference engines
  - Knowledge engineering
  - Truth maintenance, etc.,
  - Learning systems
  - distributed systems

Annex 6 - Robotics

- Structures
- Kinematics
- Dynamics
- Languages
- Vision
- Path planning
- Intelligence
- Multi robots

Annex 7 - Databases

- Architecture
- Hierarchical model
- Network model
- Relational model
- Database management systems
- Industrial systems
- Databases in manufacturing
- Building a database

Annex 8 - Books/Reports/Papers/......

- Modelling/Simulation
- Scheduling
- Planning
- Artificial intelligence
- Expert systems
- Robotics
- Image processing
- Networks and protocols
- Operating systems
- Databases
- Distributed systems and architectures
- Production manufacturing
- Project management

Annex 9 - Documentation

- GE master plan
- CAD/CAM literature search
- Expert systems in manufacturing
- NBS/AMRF
- Project management
- Real-time shop-floor control
- Selected articles in CAD/CAM
- CAD/CAM databases (in French)
- Standards in CAD/CAM
- ICAM architecture
- ICAM microfiches (10 sets)
- IPAD microfiches (2 sets)
- Expert systems microfiches (2 sets)
- Manufacturing information system microfiche (1 set)

Annex 10 - R and D Projects in CIM

1. CAE

   - Finite element methods
   - Dynamic system control and optimization

2. CAD

   - Expert systems for drafting and drawing
   - Group technology
   - Generative planning

3. CAM

   - Real time simulation(emulation) and scheduling
   - Distributed numerical control
   - Expert systems for part process planning
   - Expert system for adaptive scheduling

4. CAT

   - On line testing
   - CAD directed inspection

5. Expert and Intelligent Systems for

   - CAE
   - CAD
   - CAM
   - CAT
   - Adaptation and monitoring
   - Image processing and pattern recognition
   - Voice and speech interfaces
   - Learning

6. Distributed Systems

   - Architecture : hardware, software, networks .....
   - Tools : finite state machines, Petri nets, IDF's, DFD, ...
   - Distributed intelligence and expertise
   - Distributed databases

7. Robotics :

   - Off line programming (accura[illegible] path generation)
   - Task level programming
   - 3 D sensors and vision (recog[illegible])
   - Trajectory tracking (moving target)
   - Path planning (collision avoidance)
   - Mobile and multi robots