THE DEVELOPMENT OF A TOTAL SYSTEM SIMULATION MODELLING APPROACH FOR MANAGEMENT DECISION SUPPORT

DAVID ISRAEL GOODEN
Master of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

May 1988

SYNOPSIS

One of the main managerial tasks in a manufacturing environment is that of organisation. This involves the arranging and rearranging of limited resources so that the production activities contribute to the overall objectives of the business. As a result, the management need to make operational policy decisions in order to run their business efficiently.

With the increasing complexity of manufacturing information flow and the increased diversity in the requirements of the customer, the manager is faced with the prospect of investing in soundly based tools for controlling production operations. MRPII is such a tool and typically offers a wide range of policy combination choices. The problem lies in deciding which policy combination is suitable for the production system in question.

Many simulation studies have attempted to analyse this problem. However the applicability of their conclusions have been limited by the narrow scope of the models used. Current research now accepts that the complexity of the interactions found in the real production system is such that their representative inclusion in the models used is vital to the validity of the results. This approach has been applied to a specific company, Fulcrum Communications Limited(FCL), who having recently been affected by the privatization of their parent company, showed the criteria necessary to conduct an extensive research study. The particular circumstances of FCL were characterised by a lack of relevant historical data. This, in conjunction with their recent investment into an automated manufacturing facility and an MRPII system lead to the development of an experimental facility which was designed to evaluate system control policy combinations. The resulting model produced a detailed simulation of the manufacturing facility at FCL which communicated in real time with an integrated model of the currently implemented MRPII system. This 'total' model approach first considered the bounds of the system model and then used sophisticated computer hardware and comprehensive computer software to simulate the factors which had a significant effect on the efficient running of the production system.

To this end the development of the total model and its potential application as a management policy decision aid in a production system environment is discussed.

Key Words: Simulation, MRP, Policy Design, Total modelling, Manufacturing system, Production Control.

DEDICATION

To Mum and Dad for their patience and support during the writing of this thesis.

ACKNOWLEDGEMENTS

God is my strength and refuge.

The author is extremely grateful to Dr D. Love of Aston University for his close guidance and valuable suggestions throughout the course of the project. Special thanks are also expressed to Dr J. Penny for his involvement in supporting the extension for further work at Fulcrum Communications Ltd.

Thanks are also expressed to all members of the Information Services department at Fulcrum Communications Ltd. for their co- operation and assistance during the model design and experimental analysis stages. The author would particularly wish to thank;

Mr D. Pinwell for the use of the computer resources and his encouraging and positive approach throughout the project;

Mr D. Gore for his involvement in the initiation of the project;

Mrs L. Fenn for her professional assistance with TMS;

Mr R. Evans and Mr R. Hitchin for their professional help with the TMS communication programs;

Mr J. Hall for his help with computer hardware set-up;

Finally, Special thanks are extended to my colleague and friend, Mr S. R. Clarke for his guidance and invaluable contribution to the research.

CONTENTS PAGE

CONTENTS **PAGE**

CONTENTS **PAGE**

CONTENTS	PAGE

FIGURES PAGE

FIGURES	PAGE

TABLES **PAGE**

APPENDIX	PAGE

1. Introduction

1.1 Research overview

Over recent years market trends in Britain have contributed toward creating an environment within manufacturing which is becoming increasingly frantic. Product lives are under compression, manufacturers are having to support greater product diversity and despite rising overheads, the markets have demanded that unit costs be minimised. Market demands, together with the associated competition has forced manufacturing industry to undertake a radical self examination of its operating performance. This , to a large extent, has been aimed at creating a more efficient and cost effective manufacturing base by minimising waste and maximising the potential of available resources including time, labour, money, materials, energy, plant and equipment. Moreover, to maintain a lead in this environment, organisations are beginning to ensure that their manufacturing resources offer flexibility as well as fast response.

In a bid to meet the challenge of present day market conditions, manufacturers have turned increasingly toward production facilities which offer responsive, short lead time performance and enable materials investment to be tightly controlled. However, the complexity of these systems is often such that they are beyond the ability of most people to gauge their capability intuitively.

At the managerial level, the main task in a manufacturing environment is that of organisation. This involves the arranging and rearranging of the limited resources in the manufacturing facility so that the production activities contribute to the overall objectives of the business which include satisfying the market and competing with the opposition. Management is therefore required to make operational decisions in order to run their business more efficiently. The output considerations over which management must seek its optimal balance include products, employment, profit and economic growth. It is against this background that analytical approaches to decision making have become more widely accepted.

One of the techniques which has received considerable publicity over recent years has been computer based simulation. Many simulation studies have attempted to address the problems associated with the manufacturing environment. However, the applicability of some conclusions have been limited by the narrow scope of the models used. Current research now accepts that the complexity of the interactions found in the real production system is such that their representative inclusion in the models used is vital to the validity of the results and subsequent conclusions.

This approach has been applied to a specific company, Fulcrum Communications Limited(FCL), who having recently been affected by the privatization of their parent company, showed the criteria necessary to conduct an extensive research study. The particular circumstances of FCL were characterised by a lack of relevant historical data. This, in conjunction with their recent investment into an automated manufacturing facility and an MRPII system lead to the development of an experimental facility which was designed to evaluate system control policy combinations. The resulting model produced a detailed simulation of the manufacturing facility at FCL which communicated in real time with an integrated model of the currently implemented MRPII system. This 'total' model approach first considered the bounds of the system model and then used sophisticated computer hardware and comprehensive computer software to simulate the factors which had a significant effect on the efficient running of the production system.

This dissertation discusses the development of the production system model and its potential application in a production system environment.

2. A Production System Problem

2.1 System Integration

The Production Control System and the Manufacturing Facility are often considered as independent functions of the business. Consequently, they are largely designed and implemented in isolation of one another. The criteria, therefore for control policy and parameter selection under these circumstances is based on sub- system evaluation rather than whole system performance.

This state of affairs is highly contentious since sub-system optimisation does not imply whole system optimisation. The potential for conflict between the individual requirements of these two sub-system, and the complexity of their interactions, further adds to the lack of confidence in system evaluation without whole system consideration.

2.2 System Control Policies and parameters

The successful implementation of the two systems as mentioned above demands that a number of major issues are addressed. In generalised terms, MRP policies and shop parameters form an interacting chain descending from market demand to raw material supply. Consequently their impact can be felt at all business levels. Those at the manufacturing operations level include;

- lead time allowances;

- inventory levels;

- product mix;

- bottleneck identification;

- priority/scheduling rules;

- batch sizing rules.

In addition, a further group of issues exist which by their nature influence the interaction between Manufacturing, Marketing and Finance. These relate to the policies applied to customer demand (actual and perceived), in the generation of the Master Production Schedule (MPS) and include ;

- order/stocking policy;

- MPS horizon;

- MPS variability;

- level in BOM at which MPS is applied;

- firm/tentative ratio in the MPS;

- MRP frequency;

- forecasting algorithm to be applied.

The performance of the MRPII system in terms of the primary control outputs (i.e. the ability of the orders to reflect the market requirements) together with order size, frequency and timing is a function of the control policies used. In turn the control policy outputs have a direct influence on the efficiency of the manufacturing facility and its ability to achieve the lead times and capacity levels assumed. Thus, the performance of the two systems is interdependent and again a function of the control policies used.

Most Production Control systems are highly configurable and consequently allow many control policy combinations to be selected. This can offer the essential flexibility for the many differing business requirements. Associated with this degree of flexibility however, is the diversity of operational performance. For this reason the design of control policies in this environment must be seen as fundamentally important for the successful integration of a Production Control System and the Manufacturing Facility. Furthermore, the selection of system control policies and parameters in the manufacturing environment must be undertaken with the view to improving overall system performance.

2.3 Conflicting objectives in the manufacturing environment

The selection of optimum control policies is further hampered by the conflicting objectives of each of the three business functions ; Marketing, Manufacturing and Finance. If we consider the sub-system alone then one of the main objectives of the Marketing function is to satisfy customer requirements. This is normally achieved by offering and achieving good delivery dates and providing a good range of products to satisfy the customers ever increasing demand for variety. It follows therefore that if Marketing were given the task of selecting control policies to achieve their particular objectives then they would need to formulate stock control policies that would ensure a level of finished goods inventory to satisfy unpredictable customer demand.

Producing to stock however, is not one of the objects of the Finance department. Instead they are concerned with reducing operational expenses to the minimum and recovering as much profit from investment as is practically possible. Work in Progress (WIP) and finished goods inventory are seen as 'cash tied up'. The loss of interest on this stock translates to reduced profits and should, the Finance department would argue, be kept to a minimum. Clearly there is a conflict of interest between both the Marketing and the Finance departments over what stocking policy to use. In addition to the stocking policy dilemma The Finance department require that the machinery and plant be fully utilised to satisfy the return on investment (ROI) measures.

The Finance department, in trying to achieve its objectives is therefore concerned with those control policy combinations which will ensure maximum utilisation of machine and plant whilst WIP stocks are simultaneously minimised.

The Manufacturing function further complicates the choice of system control policy combination by the consideration of practical limitations. The objective of increasing machine and plant utilisation is synonymous with that of Finance. The reasons for high utilisation include achieving good throughput and high piece rate bonuses. Traditionally however, the view to achieving this objective has been to maintain high levels of inter-workcentre buffer stocks to negate any flow imbalances which may occurs due to workcentre starvation after a bottleneck machine, for example. Also the customer increasingly requires a greater product range. This greatly reduces the Manufacturing system's ability to run its facility efficiently due to the change-over time invested in process set-ups.

The Manufacturing function would, given the choice select system control policies that would maintain high levels of workcentre efficiency with the view to flowing jobs through the shop in the least amount of time possible whilst maintaining saleable product quality.

The conflicting nature of each of the three business functions makes control policy evaluation a very difficult task. As a consequence of this control policy evaluation methods have traditionally relied on existing knowledge and experience of the production system processes for the management of limited resources in a manufacturing environment.

2.4 Control policy evaluation - traditional view

Early MRP implementers selected policies and parameters which reflected the traditional way of running the business. This is evident in existing policy design in many production systems today. Clarke(1988) discusses some of the reasons behind the adoption of particular policy designs and concludes that their relevance in today's environment is extremely limited. Clarke identifies the characteristics of the era into which MRP was conceived and makes the following observations ;

i) end product requirements were planned against predominantly forecast demand in order to minimise the occurrence of stockouts and consequently increase customer satisfaction;

ii) long production runs were planned to minimise standard costs and maximise individual work centre utilisation;

iii) the low cost of capital was such that buffer stocks and high levels of WIP were used to negate any flow imbalances and further maintain high levels of utilisation;

iv) early manufacturing systems were not designed to operate with minimal queues and small batch sizes.

In comparison to today's standards, he argues ;

"...the characteristics of the time were typified by; low cost of capital, low international competition, and long product life cycles. Factors such as, high levels of engineering change, and customers demanding short delivery times, were not major issues."

It is clear from the above that traditional methods of policy evaluation are of limited value in the current situation. In order to assess the success or otherwise of particular policy combinations a measure of system performance is required. The accepted business measures include WIP level, lead times, stock-outs and due date accuracy. A business would normally strive to reduce WIP and flow times whilst at the same time achieving good due dates and satisfying customer demand.

2.5 The MRPII approach to policy evaluation

Britain has already invested millions of pounds in designing and installing MRP, the primary purpose being for planning and scheduling production and materials. The goal of the MRP process is to achieve a valid and realistic Master Production Schedule (MPS) which is immune from disruption. MRPII recognises that any scheduling system must consider not only the needs of the customer (customer requirements), but also the capability (capacity) of the plant to meet these needs. MRP assumes infinite capacity when actually processing, so in order to incorporate plant capacity, MRPII goes through two iterative processes, namely Rough Cut Capacity Planning (RCCP) and Capacity Requirements Planning (CRP). A conceptual overview of the MRPII process is illustrated in figure 2.1.

As the name suggests the RCCP is designed to roughly estimate whether or not a given production plan can be achieved by the manufacturing facility. The market requirements in terms of products are translated into load profiles for key production areas. These production areas are normally groups of machines or types of manufacturing process that are thought to be production constraints.

The load profiles resulting from rough cut capacity planning are usually evaluated by a management committee, composed of the key functional people in the plant. i.e. Manufacturing Manager, Materials Manager, Sales Manager and Manufacturing Controller. The purpose of this group is to judge the viability and financial soundness of this production plan under the particular policy combinations.

Manual adjustments are made in the production plan to compensate for underload and overload conditions in the plant, the purpose being to arrive at a production plan that considers both market requirements and capacity constraints.

Once the adjustments are made the process is repeated. A number of iterations may be necessary before the management group is satisfied with the production plan.

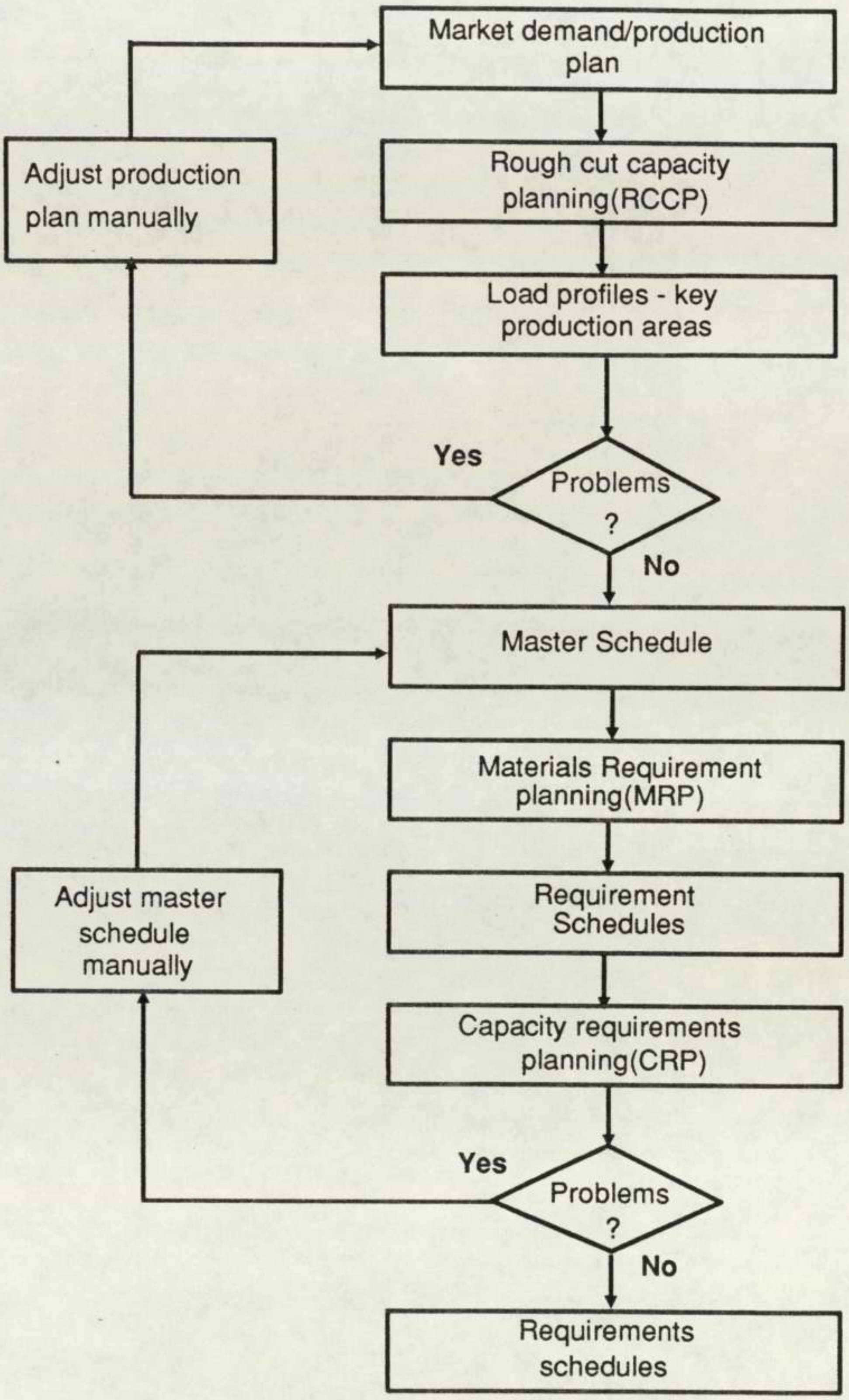


Figure 2.1 Conceptual overview of the MRPII iterative process

RCCP takes a gross view which is in fact essential in making the technique feasible. It considers product families (not individual items), key production areas (not individual machines) and long time frames (usually months or quarters). Embedded in the technique are also a number of simplifying assumptions. Although it nets the market requirements against any finished goods stocks and explodes the balance on a period by period basis, it totally ignores WIP by assuming that the plant is empty. Furthermore, it does not take into account lead times of sub-components (anything below MPS demands). The assumption is made that all assemblies can be completed in the period they are demanded. Finally it ignores any MRP or shop control policies. For our purposes, this is perhaps its most significant disadvantage since the design of control policies at all levels of the production system is essential to the integration of the major sub-systems.

2.6 Capacity Requirements Planning

Capacity Requirements Planning (CRP) is a technique which is specifically designed to 'fine tune' the master schedule produced by the RCCP process. Figure 2.2 shows the key steps in the CRP process.

The Master Schedule is exploded into its sub-assembly component parts and raw material requirements. These requirements are offset by predetermined lead times, and netted off against any available inventory. The MRP control policies are then applied against the net requirements. These requirements are then back scheduled from their due dates so that shop load profiles can be generated for the machines and manufacturing process required by these schedules.

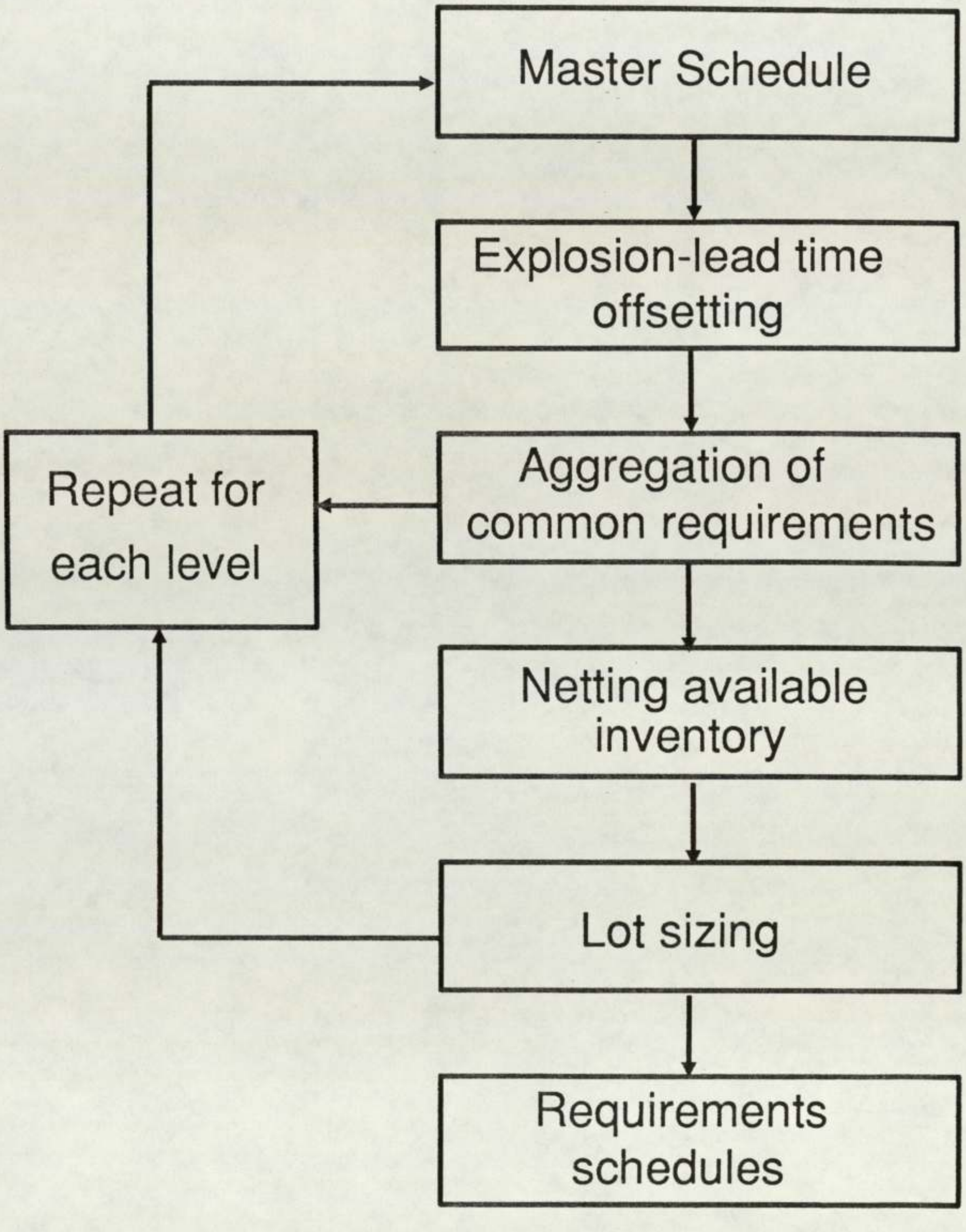


Figure 2.2 The capacity requirements planning process

This technique goes a long way towards overcoming the simplistic assumptions of RCCP. Its use is however confined to comparing alternative master production schedules. If the load generated by a proposed MPS is unsatisfactory (because of significant overload or underload in one or more periods), the schedule is changed (usually on a trial-and-error basis) and the procedure is repeated. Moreover, the fundamental performance measures of the selected control policy combinations, even with many iterations can only be considered in terms of basic shop load profiles as illustrated in figure 2.3. In addition CRP cannot be considered as an interactive method for assessing the success or otherwise of particular control policy combinations because of the excessive time and processing necessary to produce relevant information.

In practice the real system is normally used as a vehicle for determining the effect of control policy decisions on shop floor performance. Practical limitations however, make this method of analysis extremely undesirable. The situation can be made worse by running the system under control policies that are incompatible with the objectives of system integration. This can manifest itself in increased WIP levels, inflated flow times and poor customer service. An alternative method of policy evaluation could be to use an experimental facility. This in turn would offer a suitable vehicle for assessing the success or otherwise of selected control policies with the view to increasing overall system performance.

3 Selection of an appropriate modelling technique

3.1 The need for a suitable modelling technique

Chapter 2 argued that MRPII alone is incapable of assessing the success or otherwise of selected control policy combinations. This is due to the assumptions made about the manufacturing facility in terms of resource limitations. The MRP process, inherent in MRPII does however generate (as one of its major outputs) suggested WIP orders. These suggested orders, once confirmed are used to specify the production plan for a given period in the manufacturing facility. Consequently the production plan reflects not only the control policies used at the MPS and MRP levels, but also the requirements of the shop to fulfil the objectives of the management. In this way, the production plan provides the system interface between the production control system and the shop floor. A means therefore, of assessing the effects of control policy combinations on both the production plan and the subsequent attempts of the shop to work to it, was considered to be of prime importance.

Production plan quality manifests itself through the smoothness, period load and average order size of the suggested orders profile. Clarke(1988) also supports the view that WIP orders contain essential information about the policy combinations used at the MPS and MRP level. He argues for example, that minimum stock and pan size policies may result in 'lumpier' profiles since demand is aggregated. In addition to this the average demand per period may also be influenced. Clarke(1988) expands the idea of WIP orders forming the fundamental interface between the production control system and the manufacturing facility by arguing that;

> "Many of the issues relating to AMT(Advanced Manufacturing Technology) operational policy design are seen to be inextricably linked to the policy decisions at the production control and MPS level. Therefore, the holistic approach to policy design proposed for the MRP element must be extended to the manufacturing facility."

The above suggests that in order to establish the overall effect of given control policies at the upstream level in the production control system, the MRP outputs (WIP order) should be used to drive a model which adequately represents the real manufacturing facility. This model could also be used for investigating the dynamic response of the system and identifing the significance of its constituent parts.

3.2 Selection of the modelling philosophy

Following a detailed literature survey, two possible modelling philosophies were identified as being feasible approaches to addressing the problem outlined. These were namely, computer simulation and finite capacity scheduling. Whilst a more detailed analysis of computer simulation is given in the following chapter, the two approaches are compared here for completeness. Reference has been made to a number of different finite capacity scheduling packages during the literature search. These included the Micross Manufacturing System, W.A.S.P, Forth Shift and W Squared.

Both philosophies are capable of representing the manufacturing facility, however their approaches to modelling are very different. Computer simulation provides an opportunity to mimic the process of a system in a dynamic manner. A finite scheduler, although it imitates the way in which the system elements interact with one another, can only yield static information.

Finite capacity scheduling offers some advantages over computer simulation. A scheduling package is comparatively easy to implement. The development time required for learning and model building is relatively short due to the packaged modular form. The model attempts to yield an optimised solution which would be feasible under the constraints imposed. The packages easily accommodate real system features such as varying operator skills, independent machine capabilities, parallel processing workcentres and multiple shifts.

A finite capacity scheduler is a specialist software package which is specifically set up to schedule and sequence orders on an operation by operation basis. It is normally then capable of projecting forward in time to produce work to lists, load profiles and customer delivery schedules. The main requirement of a model as proposed above would be for it to adequately transform MRP generated production plans(planned orders) to shop performance information, since this is a well accepted means of assessing the overall performance of the production system. This modelling technique works by loading WIP orders which are due for release in the manufacturing facility. In doing so the utilisation of limited resource (i.e. labour, machines etc.) is represented. This is synonymous to the real system where, capacity is dependent on the total set time which in turn is dependent on product mix. Furthermore, the job interaction between workcentres is reflected by the queued jobs at each process. Finite scheduling is however, limited in its ability to measure the performance of a given manufacturing scenario under the influence of selected control policies. For example, the technique offers no facility for incorporating stochastic elements such as statistically distributed machine process and set up times, thus limiting the models use to deterministic studies only. Moreover, finite scheduling is not designed for interactive experimentation. As a result system control policies cannot be readily modified to reflect different operating situations. This inflexibility also denies the analyst the facility to extensively evaluate the interactive nature of alternative policy combinations.

It has been argued that finite capacity scheduling, although extremely useful in production system studies, is limited in its applicability to the current situation. An alternative technique which has received much publicity in recent times and could offer a means for conducting experimentation in the context of this research is computer based simulation. The discussion in chapter 2 also suggests that maximum benefit would be approached if any modifications to the system elements are evaluated using a modelling technique capable of representing the total production system. This approach will enable the interactions both within and between the system elements to be realistically appraised.

The following section describes this modelling philosophy and discusses its application in the light of current and traditional views in the manufacturing environment. The main discussion centres around its usefulness as the basis for a total production system model for system performance orientated experiments.

4 Computer Simulation in the Manufacturing Environment

4.1 Background to Computer Simulation

Simulation was the first conscious attempt to imitate a sequence of happenings in time so that various policy decisions and actions could be tested and their effect on the system evaluated. Much material has been written about its application, but in the main proposal appraisal has remained with the analyst and divorced from the decision makers, who in the manufacturing environment represent the management. Since both parties invariably have differing viewpoints on the nature of the problem and the objectives of the study, there have been misunderstandings between them. This is further aggravated when the resulting model produces solutions which are contrary to management intuition. Acceptability to management is therefore a pre-requisite for successful implementation of a simulation model.

Digital simulation techniques began to emerge around the late 50's. Conway(1959) produced some of the earliest papers on their useful applications. However Shubik(1960) must be credited with recognising simulation as a potentially invaluable tool for industry and the firm. The first models were very simplistic in their construction and were greatly restricted by the amount of computer power available at that time. Hurrion(1977) produced what can be considered as a milestone in simulation techniques. He investigated the job-shop scheduling problem using visual interactive simulation methods. In his conclusion he wrote :

"It becomes realistic for a manager/decision-maker to explore the implications of different decisions or strategies with the aid of a real time model."

The rapid increase in computer power was welcomed by the now enthusiastic simulation model analysts. This paralleled the equally increasing dynamic nature of the manufacturing environment. Simulation sought to solve the complexities of increased production volume, changes in product mix, batch sizing policies, stocking policies, costing policies, scheduling rules and shop floor configuration. Its potential use was limited only by the analyst's imagination.

4.2 The simulation concept

Various definitions for computer simulation exist. Shubik(1960) states that:

"A simulation of a system or organism is the operation of a model or simulator which is a representation of the system or organism. The model is amenable to manipulations which would be impossible, too expensive or impractical to perform on the entity it portrays."

Shannon(1975) defines it as :

"A computer technique which seeks to mimic the sequence of happenings of a real system through time without actually using the real system".

In both definitions the system is referred to as a collection of items from a specific sector of reality that is the object of the study or interest.

Simulation model building is a complex process and is more of an art than a science. The construction of 'mental' models is inherent in the human thought process, but is fraught with defects. This is because the unaided mind is not able to relate all the complex factors in a system, and mentally trace their interactions through time.

A computer simulation model is an unambiguous statement of the way in which various components of the system interact to produce the behaviour of the system. Computer simulation models can therefore assist us in overcoming the inherent weakness of our mental models by allowing the analyst to describe a system in very precise terms.

The behaviour of the system is represented by a set of abstract relationships. These relationships are then manipulated in the computer where various parameters are tried and their total effectiveness evaluated.

Computer simulation modelling is ideally suited to multi-line, multi-product shop environments because it represents a dynamic situation, it handles feedback to aid corrective action, data is presented to the user serially and not just at the beginning, it is a fast evaluation tool and it increases understanding and appreciation of the problem. Furthermore, it allows the analyst to investigate dynamic problems where it may be impossible or too costly to observe them using the real system. For example, computer simulated test flight experiments were exhaustively conducted before the first manned spacecraft (Apollo 11) could leave the earth's atmosphere and land safely on the moon in July 1969.

Visual representation features and sophisticated graphical techniques are now becoming more and more part of simulation modelling, both at the model design stage and during model use. This makes computer simulation a powerful communication tool, providing a common basis for problem awareness and insights to the user.

Despite all this, computer simulation is no panacea. It will not necessarily give definitive answers, nor will it provide an optimum solution automatically. Shannon(1975) explains that :

> "Simulation modelling is probably not a search for absolute truth or correctness but rather a succession of theories that will progressively approach the truth."

From this it is clear that simulation makes no specific attempt to isolate the relationships between any particular variables. Instead, it mimics and observes the way in which all variables change with time. Therefore, if a change to the model yields subsequently improved performance during the simulation, there is a good chance that replicating the same change in the real system will also yield a benefit. Computer simulation modelling, like other modelling techniques, relies totally on the analyst's ability to build and structure the model competently. This is so that not only do the model elements closely mimic those in the real system, but also that the way in which the model interacts and performs reflects the behaviour of the real system. Incorrect assumptions and any misconceptions regarding the system of interest can lead to invalid and therefore useless models. So although the results of a particular simulation study may appear to be meaningful they could be totally misrepresentative.

The analyst is responsible for deciding the appropriate type of performance criteria. Moreover, it is only through conducting a series of experiments and observing the subsequent behaviour that he can decide when the model is moving toward a proposal that will meet specific requirements. The long development time associated with model and experimental design can be a significant factor at the computer simulation proposal stage. It is not uncommon for some studies to take many years before meaningful results are reported.

4.3 Model classification

Simulation models may be classified under a number of headings. Mize(1979) suggests the following :

i) Deterministic vs. Stochastic

ii) Static vs. Dynamic

iii) Equal vs. Nonequal Time Increments

iv) Continuous vs. Discrete vs. Combined

The classification of a model is determined more by the modelling approach adopted and the assumptions made rather than by the type of system being modelled. In addition, the range of model languages is usually determined, and to some extent limited, by the model classification.

A model is considered to be deterministic if all its elements behave in a completely predictable manner. This assumes that it is possible to represent the system using values which can be determined before the simulation run commences. In practice however, the uncertainty (or the stochastic nature) of variables as is seen in the variance between job process times, machine set-up times, order arrival rate and machine breakdown rate is predominantly evident. This stochastic behaviour can be represented in a model by statistical distributions generated by random number sequences. A deterministic representation of a stochastic system can be developed using expected values in place of the distributions. The latter approach makes the assumption that the variation between the actual data value and the expected data value is small enough to justify a deterministic approach.

Static simulation models include techniques such as; PERT, CPA finite scheduling and Planning boards. They all present to the user a 'snap shot' of variable interaction at a given time. Their derived solution remains valid until some basic structural change occurs in the system. The dynamic simulation model gives a continuous evaluation of each 'snap shot' of time.

There are two well known techniques for handling time in a simulation model. The 'equal time increments' method uses a predetermined time increment which allows the clock to move with regular time advances. This is beneficial when the variance of the time between state changes is small. In the more common situation where the time between state changes is highly variable it is more appropriate for the simulation clock to have a variable time mechanism which advances by moving to the next state of change of the system.

The manner in which the variables are perceived to change values determines whether a simulation will be modelled as discrete or continuous. When all the dependent variables change continuously with time, then the model is termed continuous. This technique describes a system in terms of Flows, Rates, Levels and Delays. For example the effect of tool load on the speed of a job spinning in an NC lathe can be investigated using continuous simulation since both variable values are changed continuously with time.

Dependent variables whose changes occur as finite stages are classified as studies which produce discrete simulation models. It is relatively easy to describe manufacturing processes in terms of instantaneous or discrete state changes. For example the specific point in time at which 'begin processing a job' might occur is a convenient switch between an idle machine which is available for processing and the same machine processing a job.

Pidd(1986) suggests that discrete simulation models may be further divided into four widely used modelling approaches :

a) EVENT based

b) ACTIVITY based

c) PROCESS based

d) 3-PHASE APPROACH

Each of these discrete modelling approaches embodies a distinctive world view which attempts to trace and observe the interactions of all the model elements. All the above approaches produce simulation programs with a 3-level hierarchy, as follows:

- Level 1..... EXECUTIVE (control programs);

- Level 2..... OPERATIONS;

- Level 3..... DETAILED ROUTINES.

The executive is responsible for sequencing the operations which occur as the simulation proceeds(i.e. this controls level 2). Level 1 primarily determines when the next event is due and ensures the operations occur at that time. The second level is the set of statements (program code) describing the operations that make up the model. These are explicit instructions about the entity interactions. The third and final level is the set of routines used by the second level to model the detail of the system. These include routines for producing reports, collecting statistics and changing the visual clock digits.

The event based approach to computer simulation is designed so that the executive completes a continual three phase cycle until the run is over. Figure 4.1 illustrates the event based simulation flow and shows how the model proceeds at each of the three levels, through time. There are three major stages using this approach, namely ;

1). TIME SCAN

- Determines the time of the next event

- Move the simulation clock to that time

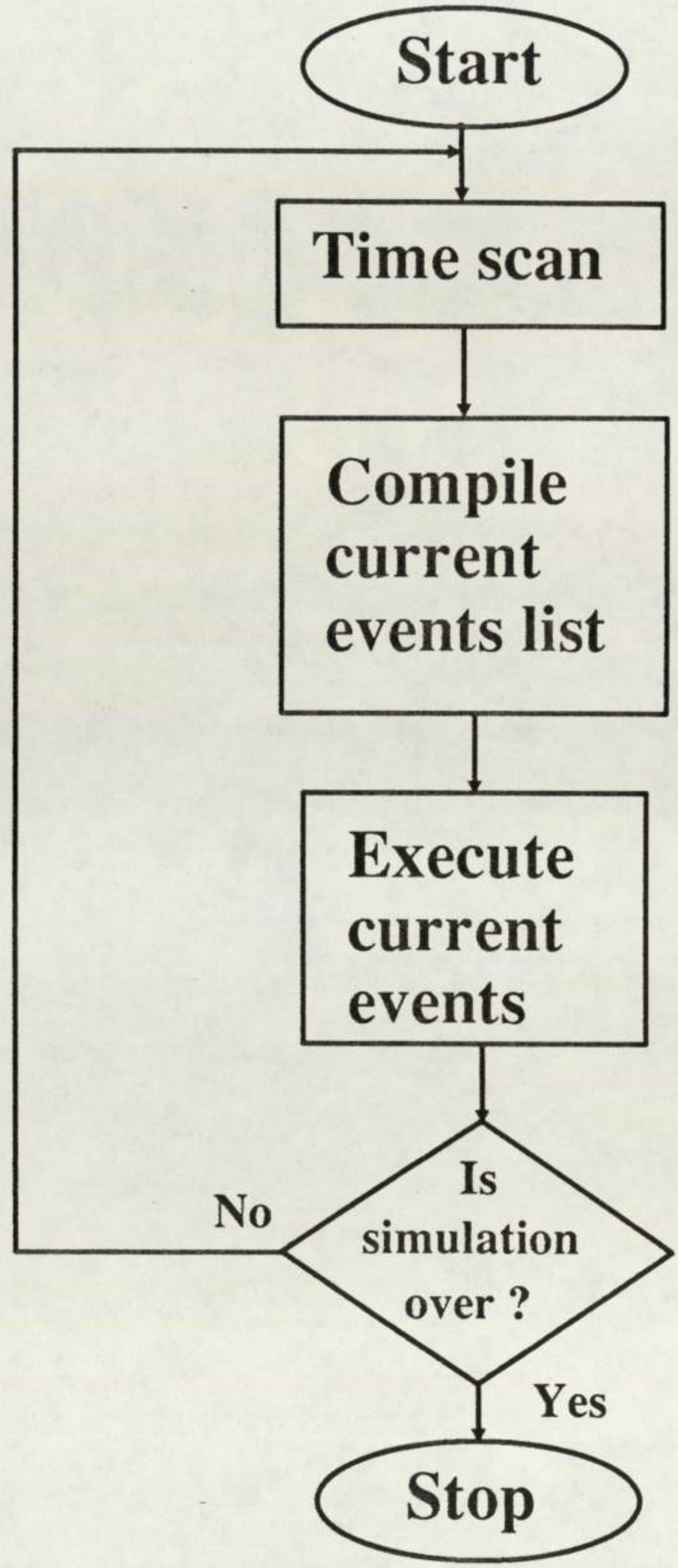


Figure 4.1 The event based executive

2). CURRENT EVENTS LIST

- This list contains all events identified as due now

3). EVENT EXECUTION

- Ensures that each event on the current events list is executed correctly.

- Once executed the event is removed from the list

This method offers the advantage of efficient processing, but unfortunately carries the penalty of potentially very large events lists and exhaustive decision statements (resulting in large programming code).

The activity based simulation approach was developed to reduce the amount of programming involved in creating a simulation model. Figure 4.2 shows that this method proceeds by continually executing two cycles until the simulation is over. During each cycle the following functions are performed;

1). TIME SCAN

- Determines the time for the next change of state

- Move the simulation clock to that time

2). ACTIVITY SCAN

- Each activity is attempted in turn until no more actions are possible

The activity scan always attempts every activity in turn, even though the conditions in the simulation may mean that certain activities can not possibly be executed. Consequently, this process becomes less efficient with increase in model

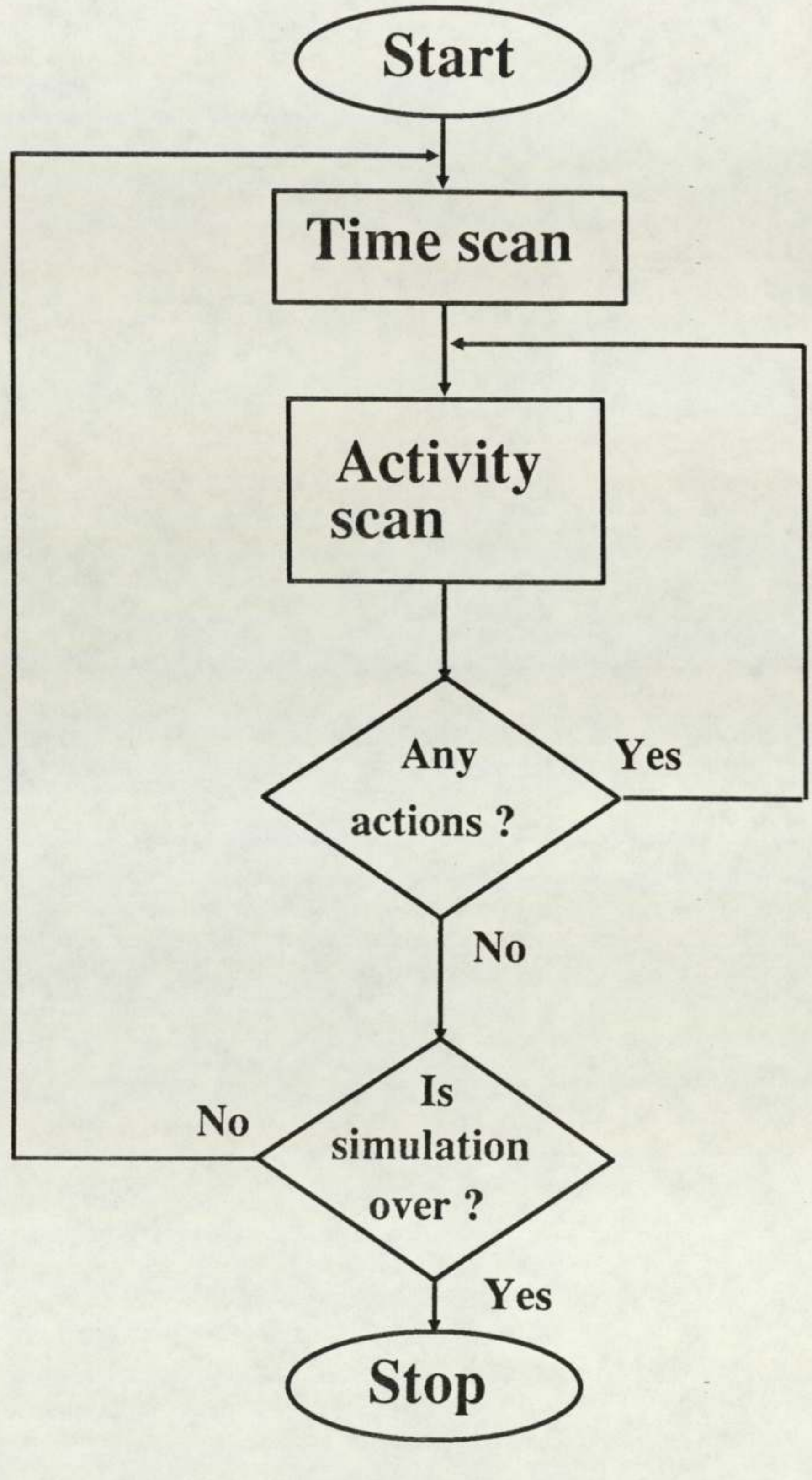


Figure 4.2 The activity based executive

complexity. The contrast is seen with the event based approach which only attempts the events in the current events list.

The process based approach to computer simulation modelling traces the activity (and hence interaction with other entities) of each entity as it progresses through the system model. Each separate temporary entity has its own process which stops and starts as the simulation proceeds. The more complicated flow of this method is shown in figure 4.3. As the simulation proceeds the executive completes two cycles until the simulation is over. The first cycle has two stages and the final cycle has one stage. The function of each stage is as follows;

1). FUTURE EVENTS SCAN

- Select entities with earliest re-activation time

- Move simulation clock to that time

2). MOVE RECORDS

- Move events associated with these entities from future events list to the current events list

3). CURRENT EVENT SCAN

- Attempt to move each entity through its process until no further movement is possible

- Repeat until no more entities can move

Pidd(1986) argues that in addition to the necessity of a very complicated executive the code using the process based method is very difficult to write.

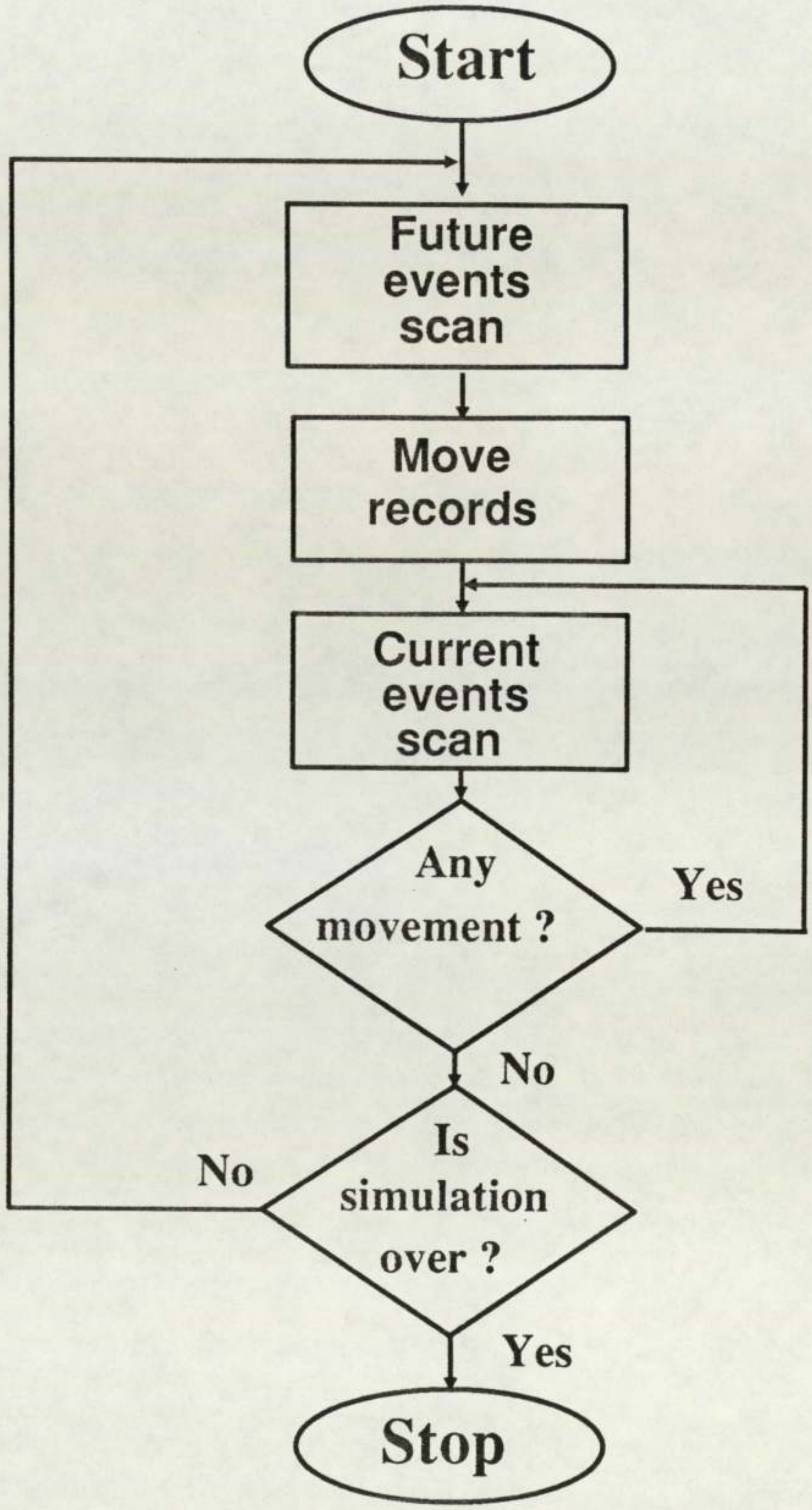


Figure 4.3 The process based executive

The final approach to simulation modelling known as the three phase technique succeeds in combining the simplicity of the activity approach with the efficient execution of the event approach. Two types of events are defined;

'B' Events (bound events)

- These are executed directly by the executive program whenever their scheduled time is reached.

'C' Events (conditional events)

- The execution of these events are dependent on either the co-operation of different classes of entity or the satisfaction of specific conditions within the simulation.

The three phases are shown in figure 4.4 and defined as follows;

A Phase (TIME SCAN)

- Determines when the next event is due and decides which 'B' events are due to occur.

B Phase (BOUND EVENT CALLS)

- Executes all those 'B' events identified as being due now

C Phase (CONDITIONAL EVENT CALLS)

- Attempts each of the 'C' events in turn and only executes those whose conditions are met

- Repeat 'C' scan until no more activity is possible

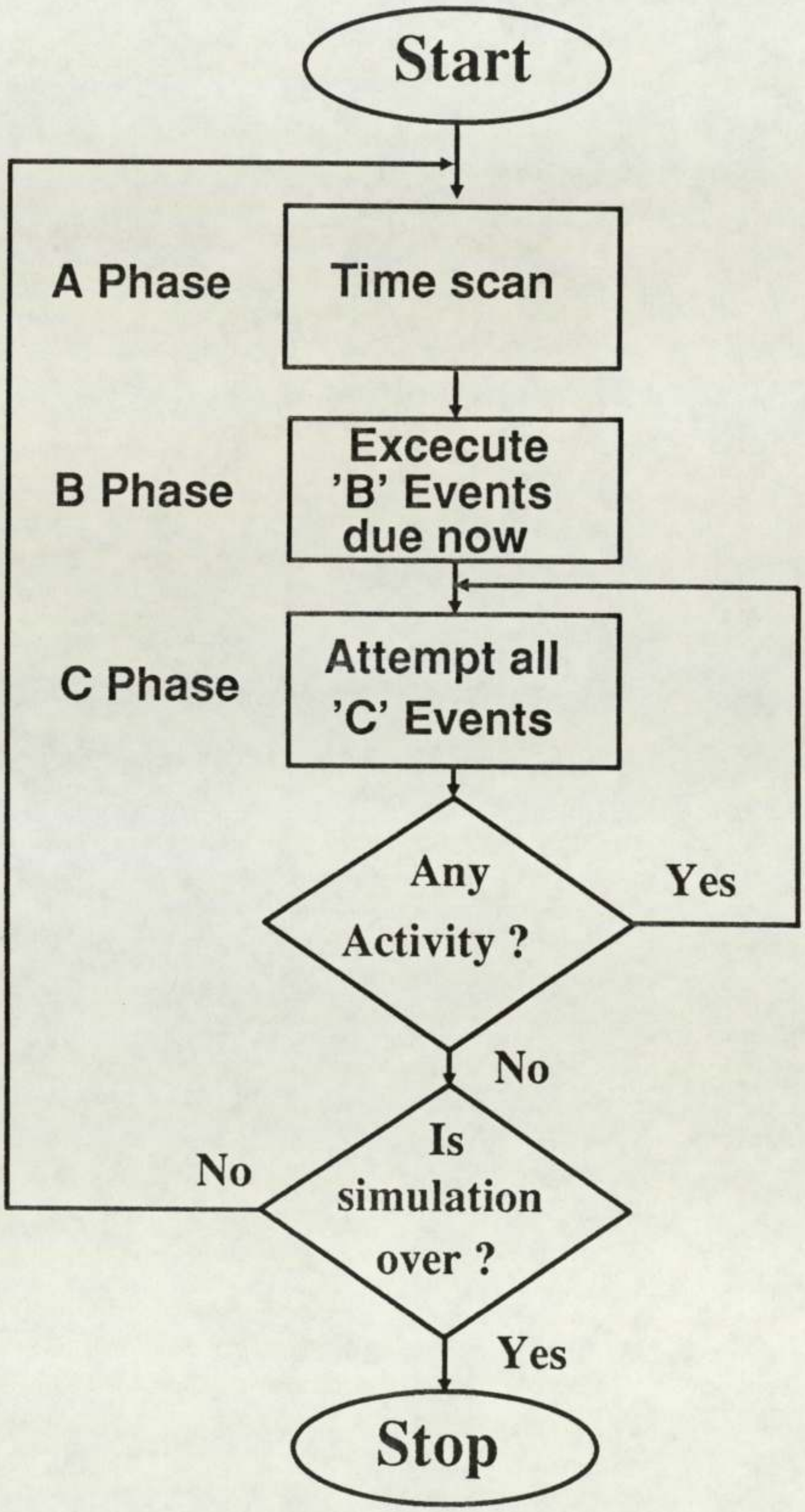


Figure 4.4 The three phase executive

A detailed account of the three phase approach can be found in Pidd(1986) and Tocher(1963).

4.4 Simulation in the production system

The literature reports on a wide variety of simulation studies within the manufacturing environment. Early simulation studies were in the design of the production system with particular emphasis on the manufacturing process. Considerations such as shop layout, product mix and company policy were included in the analysis.

These production system models normally had three distinct functions; the Market Operation (whose job it was to generate orders with unique specification parameters), the Scheduling Operation (which organised the orders into efficient schedule programs) and the Production Operation (which was responsible for converting the orders into tangible products in order to satisfy customer demand). Pegels(1969) reports on a simulation model that was designed to represent a complex industrial system which manufactured corrugated board in the form of flat sheet. The model was made up of an empirically based order generating process, the corrugator scheduling operation and the corrugated board manufacturing process. The main purpose of this model was to design an optimal production process by assessing its behaviour on a long term basis. For example, one of the experiments performed investigated the likely consequences of imposing a minimum order quantity policy. Another experiment sought to evaluate the effect of machine set-up times on the performance of the process.

Planning the production process on a long term basis was essentially seen as an important aspect of production process design. In addition, an equally important use of computer simulation was recognised for the evaluation of short and intermediate term production process problems. Pegels(1976) in another paper describes a simulation model which was mainly used to give advanced warning of bottlenecking machines so that management could take corrective action to avoid the inevitable delays. In addition, Pegels states, "The simulation study may also be used to evaluate the following :

i) the effect of the queues in the various work centres due to changes in the batch size, overtime rules, and the various methods of calculating lead times;

ii) the number of days a part is late from the due date. This information can be used to load the part much earlier so that assembled models could be shipped at the proper time or alternatively the sequence of operations on the part can be altered so that it will be finished earlier;

iii) the work on queues due to altering the structure of the various work centres. Altering the structure involves the redistribution of the specific duties, by an expansion or reduction of service, in any given work centre."

The commercial importance of computer simulation manifested itself in the number of complete packages which became available. These packages were designed to study a wide range of industrial applications whilst at the same time being user friendly. The paper by Phillips(1977) demonstrates the capabilities of a well known simulation package called GEMS (Generalised Manufacturing Simulator). This is a Fortran based analysis program which was developed to study assembly line or job shop manufacturing environments. GEMS was used to study product flow rates, manufacturing capabilities and queuing phenomena. GERT (Graphical evaluation and review technique) , developed by Pritsker(1966), is another complete simulation package which employs a similar philosophy to GEMS in terms of construction and operation. It is a procedure for modelling stochastic decision networks.

Generalised simulation models consist of a validated model of a particular type of system, which the user adapts and re-validates to his individual needs. This is achieved by the use of input data, which is in contrast to other models which require a certain amount of programming. These simulators are therefore referred to as 'data driven'. More advanced users can incorporate patches of code into the model to allow special features to be handled. A generalised simulator is usually written using a general purpose simulation language such as GASP(Pritsker 1974), or ECSL(Clementson 1982). The simulator is at a higher level than the simulation language used and thus facilitates a more 'user friendly' environment. As a consequence, some of the flexibility and features of the original language, even within the constraints of the model are lost. This is the unavoidable trade-off to allow the simulator to maintain ease of use and relatively quick model building without the excessively adverse effects upon run-times.

Dedicated simulation packages represent a further class of computer model. They normally take the form of interactive, predictive models, which are capable of storing the current status of the production system. The response of the existing system can then be predicted for changing demands on the production process. An interactive online simulator called PASS(Predictive Adaptive Simulation

System) is described by Tabata(1977). This package is used to enhance the prediction and evaluation functions of the production process. In contrast to the conventional simulators, PASS has the following features;

i) An interactive online simulator for daily production planning and control.

ii)A package incorporating machine communication functions.

After future problems have been predicted the model will go on to suggest precautionary measures which may include changes in; due dates, job priorities and production capacity.

A prototype package called OOPS (Online Ongoing Production Simulator) was first announced at the AIIE annual conference held in Atlanta, Georgia in Spring 1980. Rogers(1980) discusses the primary purpose of this package as being an aid to the production scheduling and dispatching functions in production planning. The simulation is continually informed of what is already in the system as well as the consequences of producing any other items. Given planned loading and resource limitation, OOPS will determine the most likely state of the system in the future. The visual display of the model provides a direct communications vehicle between dispatches, schedulers and the production managers.

Many of the more modern production systems are organised using a Material Resource Planing (MRP II) system. This planning system considers the bill of materials (BOM) for each product to be manufactured, current inventory status and the Master Production Schedule (MPS). This information, along with product lead times is used to generate suggested purchase and shop manufacturing orders needed to fulfil the company objectives. Orders in the real world situation continually change due to demand or customer uncertainty. In each successive period the MPS is updated by the production scheduler to show new or improved information. Although research in the field of production control has spanned two decades, only in the past few years has it been appreciated that a manufacturing system is fundamentally dependent on its production control system.

The traditional view of simulation modelling in the manufacturing environment was to consider it as an independent tool for investigating production system problems. The idea of incorporating the simulation model as part of the actual production system for control purposes has only recently been implementable.

A simulation based production control system was presented at the first international conference, 'Simulation in Manufacturing' held in Stratford-upon-Avon, UK in March 1985. The model was designed as an interactive and rapid manufacturing simulator to form an enhancement to the existing MRPII system. The package was used to determine optimum scheduling for any given period and predict machine loadings, bottlenecks, likely delays and finish times. This information was used to control the flow of batches on a day to day basis. Figure 4.5 shows the input requirements of the simulation model and also the model predictions at any given time in the simulation run. The flow diagram in figure 4.6 illustrates how the simulation model interfaces with the existing production system set-up. The model itself is directly linked to the shop floor via computer terminals where the operator keys in the details of the job he has just completed. The package outlined is content to treat the symptoms of production control practice without seeking to investigate the possible causes.

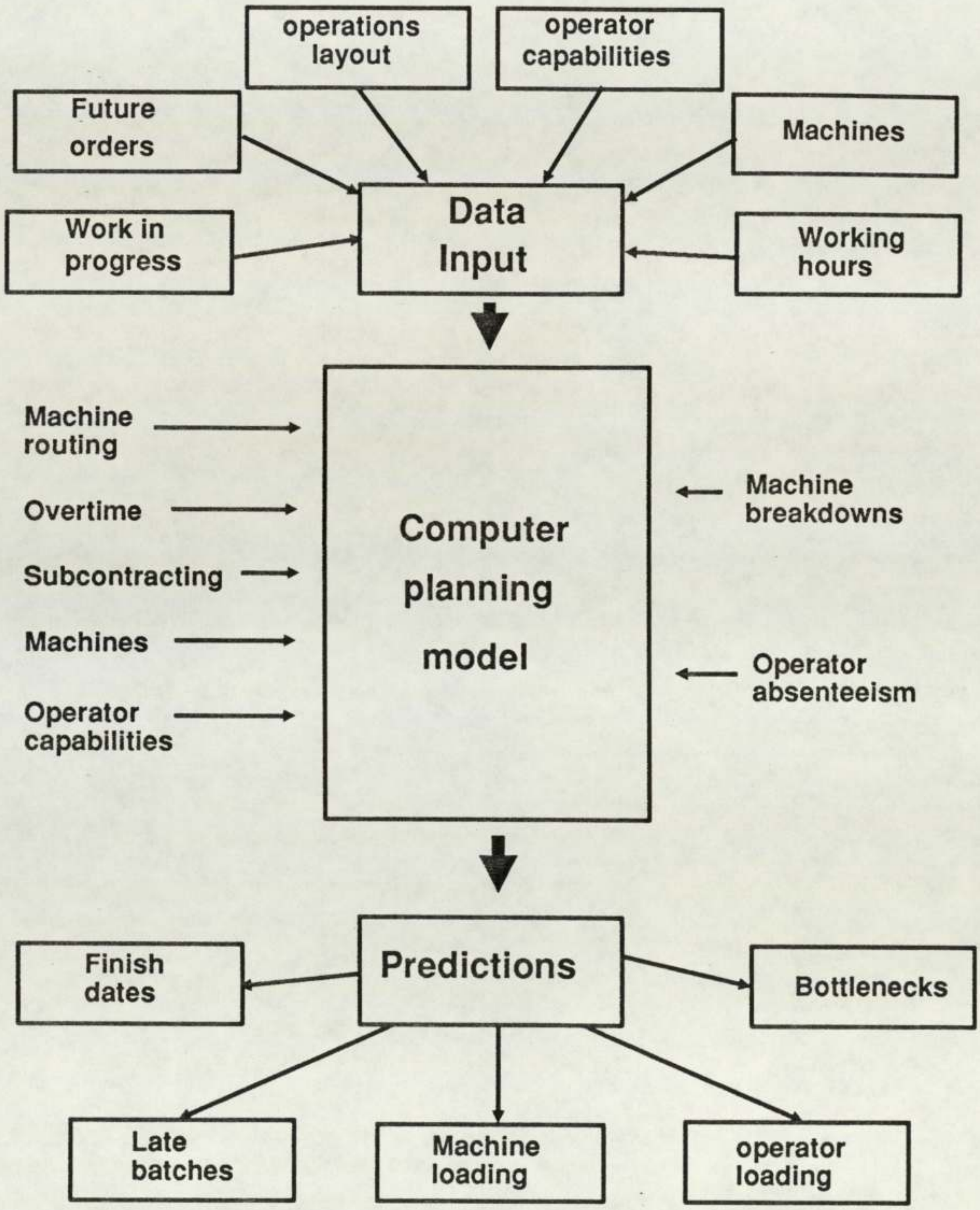


Figure 4.5 Schematic diagram of an interactive computer model after Spooner(1985,p72)

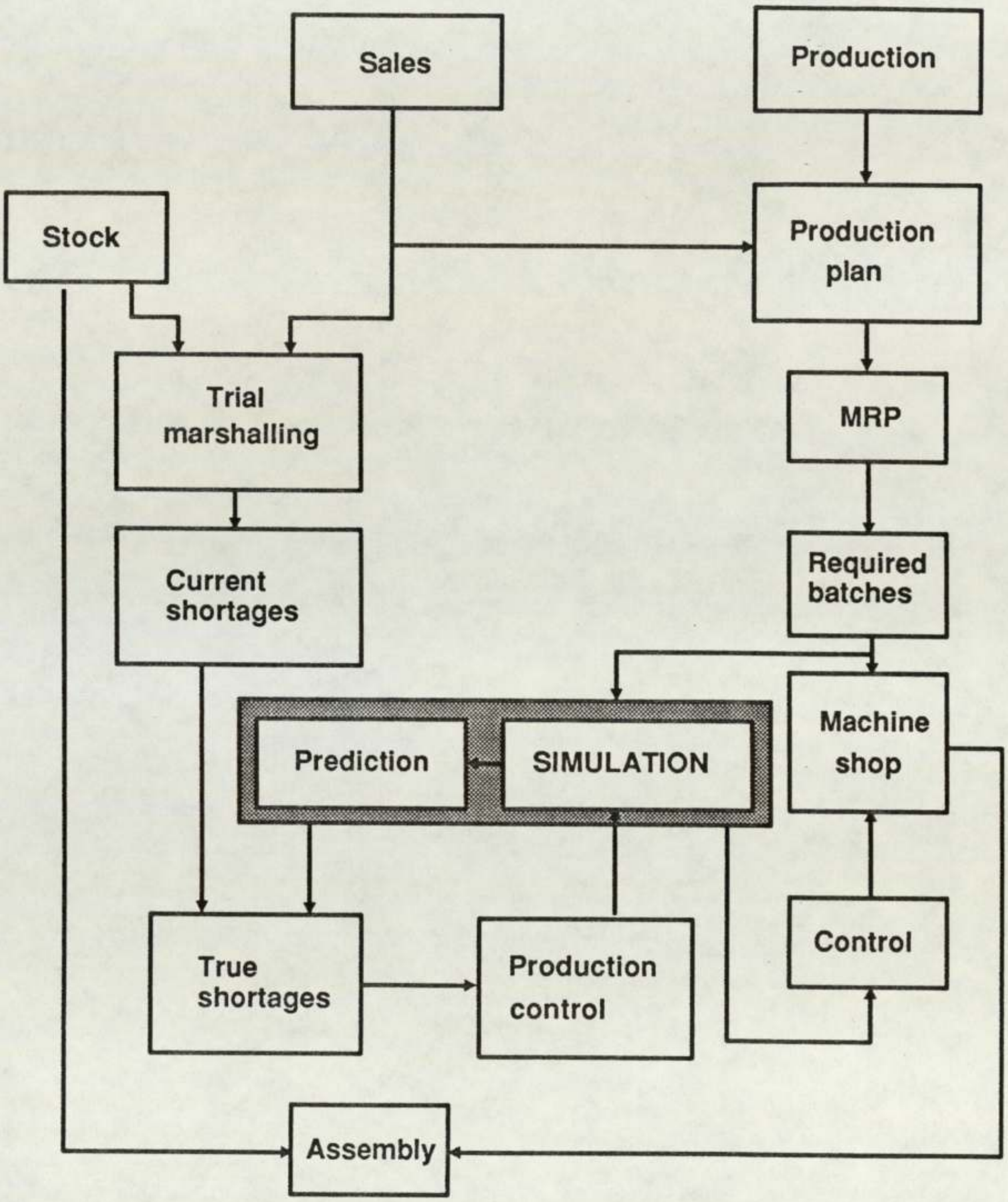


Figure 4.6 Flow diagram of a total simulation based production system after Spooner (1985,p73)

Spooner(1985,p65-73) gives a more detailed account of the model objectives and features.

The effect of uncertainty in the short term, such as machine break downs, operator performance and day to day shop scheduling, can be handled very well by interactive, predictive models. In recent years there has been growing interest in research into the effect of longer term MPS uncertainty on issues relating to the performance of the business. The importance and implications of uncertainty are recognised and discussed by Berry and Whybark(1975). Two sources of uncertainty are identified by Whybark and Williams(1976), namely demand and supply uncertainty, which they further separate into two types: quantity and timing uncertainty.

De Bodt and Van Wassenhove(1983) present a simulation study of a firm which uses MRP in a dynamic environment with considerable uncertainty. Changes were made to the MPS which reflected the actual level of uncertainty. Orders were then recalculated by MRP and the process repeated. The aim of the study was to evaluate lot-sizing and safety stock policy decisions. De Bodt and Van Wassenhove concluded that lot-sizing decisions are fundamentally influenced by uncertainty.

The effect of uncertainty on MRP system nervousness is discussed by Blackburn et al(1985). The simulation experiments, unlike many MRP studies, were carried out under dynamic or rolling schedule conditions. This was to avoid the fundamental problem of horizon sensitive systems. The considerable increase in computing power demanded by these experiments meant that research was restricted to the single level case (i.e. to avoid the inclusion of complicated interactions). The complicated series of control decisions which determine what, how many, and when orders are to be launched on to the shop floor, have traditionally forced the majority of analysts to concentrate on modelling separate sections of the system in great detail while making questionable assumptions about the rest of the system. This has been recognised in the work done on MRP implementation by Melnyk and Gonzalez(1985), lead times by Kanet(1982) and system nervousness by Carlson et al(1979). A discussion of the resulting implications have been documented in the thesis by Clarke(1988).

Minifie and Heard(1985) have also researched this issue and state that :

> "Previous research projects have established the effects of a limited number of performance measures in highly simplistic environments. In general, these experiments were carried out in simulated environments barely rich enough in realism to permit the manipulation of the experimental variables. Other policy variables were effectively held constant in that multiple values for them were not allowed. Consequently, some powerful interactions between various policy variables may have been overlooked. The conclusions of previous research are also open to question with respect to the richness of simulated environment, the set of policy variables studied, and the performance measures used".

Minifie investigated these effects with a model which used a fully functioning MRP model to drive a factory simulator. In discussing the results, she concluded;

> "The presence of highly significant interactions casts doubts on previous research conclusions based on much more limited environments".

4.5 Common assumptions made in production system modelling

In a literature search the author has cited a significant number of production system simulation studies which have been based on simplistic assumptions. The following is not intended to be an exhaustive report of the findings in the literature. However, some of the most significant aspects of simulation model assumptions are discussed. The relevant studies tend to fall into one of two categories namely;

i) distribution order driven manufacturing simulators. See Blackburn et al(1984), Arumugam(1985) and Pegels(1976);

ii) MRP order driven manufacturing simulators. See Nandakumar (1985), Bott(1981) and Hoo-Gon et al(1984).

In a distribution order driven manufacturing simulator, orders are generated using specific standard or non-standard statistical distributions. Other distributions are then used to determine the order attributes such as batch type, batch size, start date, due date, set-up and operation times, percentage scrap and sometimes even the job route. This information is used as input to the manufacturing system simulator. Order progression is simulated through each work centre. Some simulation models emulate realistic queuing disciplines where there is competition for available resource. When a job is completed it is put into a finished goods store ready for dispatch. Many of these models are characterised by a very simplistic order generation model which drives a highly sophisticated shop floor simulator.

The second type of model found in the survey was that in which the manufacturing simulator was driven by the suggested orders of the existing MRP II system. The order generation takes into account the current inventory status, real product structures, real product information (including set-up times, operation times, start date, due date and batching policies) and a real MPS. The manufacturing simulator however, uses standard or non-standard distributions to determine product flow time, based on an estimation for queue lengths, machine break downs and transportation times etc. This results in a highly sophisticated order generation system driving a simplistic shop floor simulator.

With such assumptions, analysts have in the past spent a large proportion of their studies actually trying to justify them. Many of the models rely on numerous replications and confidence tests to give what are considered to be validated results.

Blackburn et al(1982) developed a distribution order driven simulation model to test the effectiveness of proposed modifications to various lot-sizing techniques (i.e. Wagner- Whitin, Silver Meal etc.). The studies were conducted using a fixed MPS. It was later shown by De bodt et al(1983) that the results of lot-sizing studies conducted without modelling variability in the MPS cannot be valid. As Von Horn(1971) propounded ;

> "A simulation model with untested, untestable or refuted assumptions is at least disturbing."

Another example involved the misuse of lead times. Traditionally, manufacturing lead times have been based on the experience of the company in question. The MRP pioneers, particularly Orlicky(1975), have also suggested that reasonable lead time allowances are more than sufficient to generate successful MRP runs. However, Kanet(1982) showed the distinction between lead time and flow time and that whilst the former was an independent assessment of the time required to complete a product, flow time (the actual time taken to complete the product) was dependent on the dynamic nature of the system. In a study to determine an optimum job shop priority rule, Arumugam(1985) produced a simulation model which was driven by randomly generated parameters. One of these was product flow time. The fact that this was used as an input to, and not as a consequence of, his model indicates a serious misunderstanding of the difference between lead time and flow time, and hence a dangerously invalid assumption.

Pegels(1976) presents a distribution order driven simulation model to investigate the performance criteria of an existing machine shop. He selected 17 representative products from approximately 250 real products. This assumption yields a number of possible implications :

- the distribution of set-up times for the 17 products may be completely different to that of the 250 products resulting in invalid set-up costs.

- the sequence in which the 17 representative products flow through the shop may not reflect that of the real parts in the real system in terms of capacity.

- it is highly unlikely that any relationship between the product mix of the system and that of the model will be found. Since each of these areas form an important part of the model validation stage it would be very difficult for the model results to bear a measurable relationship to those obtained in the real world.

Nandakumar(1985) uses an MRP order driven manufacturing simulator to compare the performance of various batch sizing techniques in a multi-product, multilevel environment. The manufacturing simulator is represented simply by the stock flow set-up of a particular firm. This means that inter-relationships between queuing products are not modelled. By neglecting this important aspect,the effects of different batching rules cannot be fully evaluated since they are closely related to product mix.

4.6 The need for realism

Leading practitioners have strongly urged the importance of building 'sufficient' realism into a simulation model. Fisherman and Kiviat(1967) divide the process of evaluation into three categories :

1) **Verification**, to ensure that the model behaves as the analyst intended.

2) **Validation**, to test agreement between the behaviour of the model and that of the real system.

3) **Problem analysis**, which deals with the analysis and interpretation of the data generated by the experiments.

Shannon(1975) in his book, 'System simulation' states :

"Unless our modelling efforts are to be pure exercises in science fiction, the need to show that a model's output does indeed bare some meaningful relationship to what can be expected as behaviour from the real world remains one of the most crucial aspects of simulation study."

A host of other authors also air the same opinion, namely Rapaport(1953), Popper(1959), Black(1962), Churchman(1957), Conway et al(1959), Herman(1967), Mihram(1971) and Horn(1971). It follows from the above that any simulation studies done on the design of a production system should incorporate a 'total' model approach which will investigate both the model parameters and their inter-relations with one another. The literature survey also shows that the simplistic assumptions made in building a partial model not only jeopardises confidence one has in its conclusions but also artificially increases the validation phase.

5 The total system modelling approach

5.1 General discussion

The need was discussed in chapter 4 for an approach to modelling the production system in a manufacturing environment, which included sufficient realism, from which valid conclusions about the real system could be drawn. The real world is however, extremely complicated. If attempts are made to include too many features of reality in the production system model, then the analyst soon finds that he is engulfed by complicated equations containing unknown parameters and functions. In contrast, models that contain a limited number of realistic features, demonstrate predictions which may not suit the absolute requirements of the analyst. In addition, there are a number of practical considerations associated with model design which need to be reconciled if the necessary compromise is to be achieved. These include;

i) ease of model construction;

ii) specific problem solutions sought from the model;

iii)financial limitations;

iv)time limitations;

v) the inclusion of relevant elemental description of the system at appropriate areas in the model.

It is therefore pertinent that the model analyst chooses those factors that are significant to the system under study. Successful selection of the factors to include in a system model usually implies a preliminary anticipation of the types of system behaviour that could be important. Many factors receiving attention in normal company operations may have little effect, while important interactions may depend on factors that have been given little consideration. A careful and perceptive exploration of the options, influences on decisions, available data, beliefs and doubts is necessary to ascertain the model boundaries. These must in turn be examined against an understanding of how the dynamic interactions within the system may be created. On reviewing the current literature the author has cited a number of studies which have incorporated a total modelling view to the specification and development of their models. Each of the studies are by their very nature different. There is however, an element of commonality between them that considers the approach as developing a production system model which realistically represents the major relevant system factors, but does not include the overhead of impractical complexity.

Bridge(1987) extends this approach to the actual design of the manufacturing system by advocating the use of a comprehensive modelling strategy as well as the normal design procedure. In this way the on going design of the manufacturing system is based on a decision support system. The design methodology provides a structural approach to the identification of relevant manufacturing system elements. This is achieved by analysing the model at four levels namely; the factory, the department, the work centre and the work station. Each level respectively increases in detail to reflect the way in which the appropriate system elements interact.

Pegels(1969), in a study to optimally design a production process recognised the need to incorporate the three major elements of a production system namely, the market place, production control and the production process. Not withstanding some of the assumptions made, the approach was in itself sound and contributed much to the advancement of modelling strategy. Pegels's decision to use continuous simulation was influenced by the complexity of the proposed model and the inadequacy of current analytical or mathematical techniques. This simulation philosophy accommodates the aggregate modelling of each major element. Forrester(1971) again exploited the features of continuous simulation by developing a modelling approach for simulating not only the production system but also its influencing environment. Forrester's Industrial Dynamics was designed to investigate the information-feedback characteristics of industrial systems. He further argued that;

> "Industrial Dynamics provides a single framework for integrating the functional areas of management, marketing, production, accounting, research and development, and capital investment. It is a quantitative and experimental approach for relating the organisational structure and corporate policy to industrial growth and stability."

His total modelling approach to system evaluation has inspired a host of other analysts including Coyle(1977), Love(1980), and Goodman(1974).

The inclusion of customer demand uncertainty constitutes the representation of a major production system element. Chapter 4 discussed the importance of this model element. In recent times studies have been done under rolling schedule or uncertain MPS conditions. De Bodt et al(1982) and Minifie et al(1985) have both conducted separate studies to identify its effect. Both studies conclude that uncertainty is highly significant and should be considered in future research work under the influence of a total modelling approach. Minifie's modelling approach in particular utilises a fully functioning MRP system which is used to drive a comprehensive manufacturing simulator, known as FACTORY (a description of FACTORY can be found in Melnyk (1980). The model included three levels of parts (purchased, manufactured and finished) and included demand uncertainty in the MPS (rolling schedule). A multi-variate analysis of variance (MANOVA) technique was used to establish sensitivity of rescheduling to the policy interactions between and within the various levels.

The above discussion re-affirms the importance of using a total modelling approach for management decision support. In addition, the use of such an approach is gaining wider acceptance in the manufacturing environment. In view of these points the following section discusses the practical implications of adopting a total modelling approach for system control policy evaluation.

5.2 Alternative modelling approaches

The possible methods of creating a total production system model are summarised below ;

1) A single model which simulates the production control system driving the manufacturing system.

2) A model of the production control system driving an independent model of the manufacturing system.

3) The actual production control system driving a model of the manufacturing system.

To be complete, a fourth category would include a model of the production control system driving the actual manufacturing system. This later approach however, would clearly be highly undesirable since experimental expediency is of prime importance and the resulting disruption to the real system would be unacceptable.

It was initially suggested that continuous simulation would provide an appropriate basis to represent the whole production system as one contiguous model. Forrester(1971) demonstrated this approach in his work on system simulation. Using this approach the major model elements are represented by logical decision gates which respond to variable inputs and control the flow of information in the system. The use of variable rates and levels in this way allows very complicated systems to be modelled without the overhead of extensive coding. Love(1980) reports on a typical continuous simulation model which combined whole system appraisal with minimal program coding. In addition, interfaces between the specific model elements representing the production control system and the manufacturing system would be fairly straight forward.

The resulting model could be classified as a general representation of the production system, since the detailed internal interactions are largely replaced by simplified equations. For example, the rate at which customer orders are generated may depend on the levels of actual stock, target stock, outstanding customer orders, and target level of customer orders. A change in any one of these levels would cause a corresponding change in customer order frequency. Continuous simulation clearly operates at an aggregate level by concentrating on the rates of change of populations of entities. Therefore, this technique is well suited to analysing the changes in these rates. At the performance monitoring level, it would be possible to evaluate those factors associated with work in progress levels, process utilisation levels, customer satisfaction(i.e. average lateness, stock-out frequency etc.) and machine queue lengths. Independent entity interaction such as two jobs competing for limited processing resource cannot, however, be assessed separately. The implications of this are most severe when an analysis of discrete operations is required. For example, continuous simulation techniques cannot be used to assess the success or otherwise of shop floor scheduling policies, which is a fundamental requirement in this particular study (see chapter 2). This limitation alone justifies the rejection of this technique for one which can analyse the incremental operations of system elements.

In contrast to the continuous simulation technique, discrete event models concentrate on the state changes and interactions of individual entities. Consequently, simulation analysts are able to simulate at levels of elemental detail not found in continuous simulation. This makes it possible to both evaluate and analyse internal model interactions as well as assessing overall performance. The level of detail required by the manufacturing system model should, however, be such that its purpose is not obscured by complicated interaction. A quantitative assessment of individual batch, machine and labour element is both achievable, and in line with the purpose of the current study.

The level of detail required in the production control system should be such that the generated production plan realistically represents the control policy combinations used to process the customer demands. This is again achievable using discrete simulation modelling techniques. With this method however, complex models are generated which are often cumbersome in use and give only a limited insight into the modelled system. Muth(1977) describes the use of discrete linear control theory for representing the production control function. This technique highlights the structure of the production system as well as being capable of deep and detailed analysis. The complexity however, of the resulting model using either technique, would tend to approach that of the real production control system. It follows therefore, that a possible approach to representing the production system would be to use the real production control system as the order generation element for the manufacturing system model. This suggestion, whilst offering an innovative solution for modelling the production system, also implies a much reduced validation phase as a consequence of using real system elements. The feasibility, of such an approach depends on an ability to achieve the following;

i) the use of an independent test database. Apart from causing a great deal of disruption to the users of the live system, the utilisation of the live database would be both impractical and uneconomical since it is likely to include parts which are either obsolete or are not currently being used;

ii) a reduction in the data processing time associated with MRP in order to increase the scope of the necessary experimental analysis;

iii) an efficient and reliable interface between the real production control system and the manufacturing system model. This would need to emulate the complicated interactions of both the production managers and planners as well as the production scheduler.

Since the real production control system requires a number of modifications in order for it to fulfil the total model approach objectives, the model can be classified with those suggested in 2). at the beginning of section 5.2.

This chapter has discussed the practical and conceptual implications of a modelling approach which seeks to represent the function of a production control system such that it bears some meaningful relationship to what can be expected in real life. It was argued that the highest level of representation is achieved by using real elements of the production control system to drive a comprehensive manufacturing simulation model. The remainder of the work discusses an industrial case study which was used to test this approach at the highest level.

6 Industrial Case Study - Fulcrum Communications Plc.

6.1 Introduction

Fulcrum Communications Plc (FCL), a wholly owned subsidiary of British Telecom plc (BT) is the recently formed manufacturing division. This appointment in April 1985 heralded a new era for FCL moving from a BT support organisation, which simply reacted to demands from BT, to an independently operating company. This independence has given FCL the opportunity to explore new marketing objectives and subsequently has resulted in an expansion of their portfolio in what is rapidly developing industry. Working in association with the BT Research laboratories and other units, BT Fulcrum (the company trading name) provides a comprehensive design, manufacturing, supply and product support service.

The decision to form an independent company immediately highlighted a number of operational problems. Perhaps the most significant of all was their extremely limited relevant operational experience base. At conception, FCL was also faced with a completely new product range which included the production of highly sophisticated printed circuit boards (PCB's) and hi- tech electronic telecommunication systems. In addition to this, the strategic change in direction meant that FCL were also faced with completely new production volumes.

In line with this operational challenge, FCL responded by making major investments into a new manufacturing facility designed with the latest equipment for telecommunication system production and a Material Resource Planning system (MRP II) for business control functions. In addition to this a thorough and complete training/educational program was planned which was designed to coincide with the implementation of the new production system.

6.2 A business investment at FCL

The manufacturing facility at FCL had already been designed and implemented prior to the start of this study. It is referred to as 'Modern Manufacture' and consists of mainly semi-automatic equipment in the production department and jigs and work benches in the assembly area. The machines in the production department are primarily employed to manufacture printed circuit boards(PCB's). The main functions include component preform, insertion of components into the PCB's, flow soldering and PCB testing. A special semi-automatic machine is used for sequencing components on to a bandeliered strip so that the insertion machines select the correct component for a specified position on the board. The remainder of the production department consists of manual labour to insert awkward components, perform manual reworking operations and mask the PCB's prior to flow solder.

The assembly area is used to build sub-assemblies which are used to make up configured customer requirements and finished items. Testing of the finished systems is conducted using automatic test equipment(ATE). The tests cover not only functionality but also accuracy and reliability. The flexibility of the machining and assembly processes in the manufacturing facility provides the scope for numerous configurations and control options.

Information about operation details are generated by Motion Time and Measurement(MTM) techniques in conjunction with a Computer Aided Process

Planning and Estimating System (CAPPES). These are used to complete the Operational Process Chart (OPC). Each part which is manufactured or assembled on the shop floor carries with it an OPC which tells the operator how many processes are still required to produce the part, what the next process is and how long it is expected to take.

The manufacturing facility has been designed so that it can receive suggested orders for PCB's and systems from the production control system. Once received jobs are launched and processed according to the instructions given from the production control system and the operational details of the OPC. The production activity takes a variable period of time and is primarily dependent on capacity, load, lot size and job priority.

The latest addition to the FCL initiative was the implementation of an MRPII system. The selected package was the Borroughs (now Unisys) 'Total Manufacturing System' (TMS). TMS is made up of a number of mostly self contained user modules which have been designed to aid in the control of various aspect of the business. The modules allow a great degree of business decision flexibility, by providing a large range of parameter settings. At the heart of TMS lies the MRP module which is responsible for the generation of works orders against actual and forecast demand for finished customer goods. These orders are offset from their due date by estimates of lead time. TMS also includes modules for inventory control, Work In Progress (WIP) functions, parts definition and financial assistance.

The current TMS system at FCL runs on a Boroughs A9 mainframe computer. A test version of TMS is also available on the smaller Boroughs B5900 mainframe.

The policy at FCL was to test and modify TMS modules on the B5900 before releasing them to the 'live' A9 system. The production control outputs include suggested purchase and manufacturing order reports, job and work centre status reports and order simulation features.

6.3 Major issues at FCL

In the light of the investment made by FCL, together with the lack of sufficient historical data, a number of major issues were raised. Some of the more important factors relating to these issues are summarised below.

Lead times at FCL were being determined using a simple gross value distinguishing only between printed circuit boards and assembly of completed PCB's. Lead time values of fifteen and twenty five days were given respectively. These values are used in TMS to offset the suggested orders of components for parent items. This results in raw material being ordered too early for some products and too late for others. In addition, a study by Kanet(1982) has shown that overestimates of lead time can increase the propensity for customer order change in the MPS. One of the problems faced by FCL is to determine realistic lead times with the view to 'firming' up the MPS. Consequently, it was essential that FCL knew to what extent lead time depended on the shop capacity load and product mix. This was further complicated by the fact that the shop characteristics were in themselves dependent on the lead times given.

The products under consideration, although essentially very similar in appearance had highly variable operation routings. This diversity of routes was one of the main reasons for the difficulties experienced when assessing shop floor work centre loading. Since the shop floor capacity at FCL was sensitive to the volume and types of batches flowing, it was argued that careful control over product mix would lead to subsequent control over shop load. In addition, this control was also deemed useful in relieving bottlenecked work centres and thereby increasing batch throughput. It was therefore argued that FCL needed a measure to gauge how sensitive the manufacturing facility was to product mix, if shop load characteristics were to be successfully analysed. An essential feature of this measure would be to show to what extent product mix could be modified without disturbing customer demand.

The machines in the manufacturing facility were not designed to perform multi-processing tasks. Furthermore, the ratio of set to process time on each machine and the volume of customer orders was such that the likelihood of queues being formed prior to job processing was potentially high. This combination in the manufacturing facility lead, on a number of occasions, to bottlenecking work centres, which Goldratt(1981) argues greatly influence the available capacity of the shop floor. The task of locating the resource constraints was severely hampered due to the high variability of product mix and the unpredictable effects of resource management. A method therefore, of identifying resource constraints and reducing their adverse effects was considered to be essential.

A possible method of increasing shop floor performance is to schedule work flow using priority rules. Priority rules attempt to resolve the inevitable conflict which occurs whenever more than one job queues at a given workcentre with limited resource. There are numerous priority rules in use today which use different criteria for flowing jobs through a manufacturing system. Studies, including

Aggarwal(1973)et al and Conway(1964), have shown that no priority rule is superior to any other for all measures of performance. This suggests that more than one rule may be required in a given manufacturing system. FCL needed to first ascertain whether or not priority rules would be appropriate on the shop floor and if so which ones would be suitable for their purpose.

The quantity in which items are ordered has a pronounced effect on the lumpiness in the production plan which reflects on to the shop floor loading characteristics. Recent studies, including R.St John(1984) and Aucamp(1984) have also shown that the relationship between batch size and machine set-up costs is highly significant. Fewer set-ups are required as the batch size increases, but WIP costs increase as a result. Previous researchers have generally looked at a subset of the issues related to batch sizing. Included are rescheduling, the invalidity of rough cut capacity planning and flow time. Therefore a 'total' system model approach was required to determine what effect batch sizing policy had on the performance of the manufacturing facility , and what policy would yield a satisfactory increase in shop floor performance.

Recent studies by Wemmerlov(1985), Kropp(1983) and Blackburn(1985) have shown that variability in the MPS due to demand uncertainty, batch sizing techniques and rolling schedule effects has an adverse effect on the performance of the manufacturing unit. Successful integration of TMS and the manufacturing facility therefore required that this effect be kept to a minimum. Kaimann(1969) in connection with lot-sizing studies, suggested the coefficient of variation(CV) as a general measure of the resulting degree of lumpiness in the MPS. It was argued that this would give FCL an effective indicator that would help to control the influence that variability had on the shop floor performance.

Each of the issues outlined above needed to be reconciled if successful integration of TMS and the manufacturing facility was to be achieved. Recent studies have reported a 90% dissatisfaction amongst MRP users. Clarke(1988) discusses a number of possible reasons for this unacceptable situation and emphasises the importance of policy and parameter design.

FCL recognised the significance of the operational policies and design problems raised by the integration of the two systems. In appreciation of these problems, FCL commissioned Aston University to conduct a two year project to evaluate the alternative parameters and policies which would have an effect on and were necessary for successful implementation of their production system. From the outset a need was perceived for an experimental facility which would minimise disruption to the real system. The resulting model would allow various operational and strategic management decisions to be made. In addition, the model would have the capability of analysing how well the manufacturing system performed under controlled policy combinations. Successive policy combinations would then use the results of the previous experiments with the view to improve the shop floor performance. Furthermore, it was recognised that a 'total' system model approach (as discussed in chapter 5) would offer significant benefits towards evaluating the possible interactive effects of TMS and the manufacturing set-up at FCL.

6.4 Specification for the experimental model

In order to use the 'total' modelling approach for the creation of the proposed experimental facility at FCL, in line with the objectives discussed in chapter 5 a number of system specifications are required. These specifications are discussed below.

6.4.1 The system parts model database

The real system parts database contains relevant details about all the parts being modelled. This information includes the following ;

engineering data;

i) a unique part number;

ii) a short part description;

iii) the stock location and bin number;

iv) product group;

bill of materials data;

i) a unique part number;

ii) directly related sub-components;

iii) the usage of each sub-component;

iv) effectivity dates for each sub-component;

production schedule data;

i) a unique part number;

ii) order quantity;

iii) period concerned;

stock policy and inventory data;

i) a unique part number;

ii) minimum order quantity;

iii) lead time;

iv) order policy;

v) pan size;

vi) time bucket policy;

vii)minimum batch size.

The information held about each part is used at given times during the model run to satisfy specific data requirements or to decide the direction of progress in the model when a decision needs to be made. FCL's parts database resides on an A9 Borroughs mainframe computer. TMS also uses the A9 computer to perform the production control functions that actually drives the manufacturing system. Using the actual A9 system database for modelling purposes would not be possible since this would result in disruption in the real production system. The literature search suggested however, that for a model of this type to be representative of the real system it would need to include the real product range. An alternative to using the A9 database would have been to replicate it and fabricate a test database, possibly on another computer. This test database should hold information about the real system parts such that the following conditions are met ;

i) Realistic representation of FCL's product range. This may result in a limited model product range such that the model is not over complicated or caused to produce results slowly due to long experimental runs. The validation phase however, should, with a high level of confidence, suggest that the model is representative in this respect.

ii) Realistic representation of product mix.

iii) Replication of product structure for all modelled items.

6.4.2 The production control model

The production control system at FCL (namely TMS) runs on the A9 mainframe computer and interacts directly with the system parts database. Chapter 5 discussed the advantages of using a real MRPII system to perform the production control function in the model. In order to realistically represent FCL's TMS, the production control model should include the following functions and features ;

i) Capability of interacting with a parts data base as specified in section 6.4.1.

ii) Ability to reschedule a model of the manufacturing facility at FCL based on changes in the MPS. These changes (caused by uncertainty of customer orders) are then reflected in the manufacturing shop's production plan. Consequently this would make possible the execution of rolling schedule experiments (i.e. periods are added to the end and subtracted from the beginning of the MPS between successive regenerations, thus representing customer variability and uncertainty in the MPS).

iii) A time phased production order generation mechanism capable of driving a model of the manufacturing facility at FCL. TMS uses the MRP process to perform this function. For each suggested order generated the manufacturing system model would require ; part type, quantity ordered, launch date and due date.

iv) A stock or inventory maintenance facility in order to keep a record of stock movements and levels of all the parts being modelled.

v) A comprehensive suite of policy and parameter settings which would allow the major issues discussed earlier to be investigated.

vi) A realistic representation of the market place for whom FCL would reasonably expect to manufacture its products.

6.4.3 The manufacturing system model

The purpose of this part of the model is to represent the work flow from work centre to work centre over time and analyse how the real manufacturing facility at FCL responds and is affected by the various policy and parameter combinations discussed earlier. Specifications for this proposed model include the following ;

i) Capability to fully interact with a parts data base as specified in section 6.4.1.

ii) Ability to realistically represent work flow and labour interactions between work centres. This is essential since the majority of the job time in the real system is spent queuing at work centres with limited capacity.

iii) Each work centre and as far as possible each machine or process should be represented individually. This would mean that utilisation and performance data could be easily collected and analysed for each element in the manufacturing system. Consequently, the dynamic capacity of each work centre could be determined which would yield much information about the effect of product mix and set times on shop performance. Also, with individual work centres, definite input and output job queues can be easily identified for policies where queue discipline is important. Furthermore, variable routes can be investigated and analysed in more detail. Finally the identification of bottleneck work centres would be greatly eased by the representation of individual work centres.

iv) The ability to vary the number of operators and setters at a given work centre at any time during an experiment. This provides the model with the realistic resource constraints on labour and allows detailed analysis of possible labour requirements.

v) The ability to generate detailed statistical data for analysis of the manufacturing system performance under varying conditions and policy combinations. The collected data from the model should include information about the following ;

- work in progress ;

- job flow times ;

- work centre utilisation;

- job queue length;

- customer satisfaction (job lateness, stock out data etc.);

- stock levels.

The following section discusses possible methods of realising the specifications mentioned above for the proposed FCL experimental facility.

7 Analysis of the FCL model options

7.1 Model feasibility study

Chapters 5 and 6 have discussed the approach and specification for the FCL model and have identified the model elements which must be included if realistic appraisal is to be achieved. The following discussion assesses the feasible options for the proposed production system model. This model will be configured to represent the major elements of the real production system which, as chapter 6 argues, are;

i) the market place;

ii) the production control system;

iii)the manufacturing system and

iv)system information transfer.

This assumes no financial (i.e. cash flow) constraints on the ordering policy for the purposes of the proposed model.

7.2 Market place model options

Of the four system elements mentioned above, the market place is perhaps the most unpredictable. It is responsible for producing customer demands. Each customer demand will eventually generate other demands for each of the components that make up the finished item. Consequently modelling the market place poses a great problem to the analyst. Traditionally, model analysts have used statistical distributions based on random number samples to represent the MPS (which reflects the demands of the market on the company) in studies of this nature. The procedures developed by McLaren(1977), Wemmerlov(1982) and Blackburn et al(1979) are examples of generating requirements. McLaren's method was based on sampling from a uniform distribution whereas Wemmerlov used samples from a truncated normal distribution. Blackburn combines these two methods and used a compound distribution. Each of these three procedures aimed to provide the following information about the market place for each period in the MPS against each product ;

i) whether or not there was a customer requirement;

ii) the requirement type;

iii) the requirement quantity;

The requirements in any MPS generally vary from period to period due to uneven customer demand profiles. This variation has given rise to the term 'Lumpy demand' (sometimes called demand asperity). In general, lumpiness refers to the difference between requirements from period to period ; the greater the difference the lumpier the demand. Kaimann(1969), in connection with his lot-sizing research suggested that the 'coefficient of variance' (CV) be used as a general measure of the degree of lumpiness. CV, in this context, is defined as "the ratio of the standard deviation of requirements per period to the average requirements per period." Both McLaren and Wemmerlov have used CV as an experimental factor in rolling schedule research. Furthermore, in both cases CV was found to be a statistically significant factor.

Another form of variation which Blackburn reported as one of the main causes of instability or nervousness in the MPS was demand uncertainty. Changes in demand lead to order modification. This may cause prior ordering decisions to be subject to revision and, when altered can trigger a sequence of order changes throughout the system. Blackburn in his research altered the previously forecast demands in the MPS by an amount chosen at random from a normal distribution with mean of zero and standard deviation of ten. An alternative to using statistical distribution would be to analyse FCL's market place over a suitable period of time, recording all the demands that were actually created along with any changes to their quantity during the course of the analysis. This data would be used to represent the effect of the market place on the production system by performing rolling schedule experiments. Demand lumpiness and uncertainty would both be characteristics of the data collected.

This method of representing the market place used past data and consequently evaluated what would have happened in the production system had a certain set of policy decisions been made. Discussions with FCL management indicated a preference to this latter approach since real demands would be created which take into account the actual changes in the market place. The added advantage of a reduced validation phase further justified this approach to that suggested in the literature survey.

7.3 Production control model options

There are a limited number of alternative methods of building a model to represent the function of TMS. A possible solution would be to use one of the many

MRPII packages which were currently available on a micro computer. These packages not only had the capacity to process many individual component parts but also provided information at speeds comparable with mainframes at a fraction of the cost. In addition their flexibility and compatibility with other micro based packages made them a sound investment for future modifications to the model. It was argued however, that using a production control system other than TMS would necessitate a sizeable validation phase in order to ascertain how close the model was to the real system. This would result in a large number of experiments, testing every aspect of the model and comparing its performance with that of TMS, for compatibility. Further discussions concluded that an alternative to using a separate production control system would be to use a test version of TMS running on the Borroughs B5900. This, it was argued, would reduce the validation phase to a minimum. In addition this option offered a number of important advantages. It would be possible to tailor the test version to suit the practical and experimental requirements of the model whilst at the same time imitating the function of TMS. Each part number had already been defined in the live system and the information would only need to be replicated in the test database. Furthermore, the B5900 was totally independent of the A9, therefore experiments running on the test database would have no effect on the live system. The B5900 would however, represent an operational resource on the data processing department at FCL and as such access would be limited, depending on the number of system users and available computer time.

Using TMS to model itself however, would result in long processing times, which are comparable with those of the real system. In modelling terms this presented a real problem since one of the objectives of a study of this kind was to produce speedy results. Previous MRP runs executed at FCL revealed that the processing time of TMS was a function of the number of parts actually being processed. Given the limitations of the B5900 in terms of processing speed it was suggested that a possible solution would be to reduce the data base. This step would undoubtedly raise a number of questions relating to model validity and whether or not the objectives of the total modelling approach were being violated. If, it was argued, the assumption can be made that there is no major difference between the real system and a model of it with a reduced database then this approach must be considered as a feasible option for the purposes of the FCL model.

7.4 Manufacturing system model options

The manufacturing system model was to be built using one of the many available discrete simulation packages. Any formal computer language has many characteristics. This must both satisfy the requirements of the model builder as well as the end user. A number of different simulation systems were evaluated in the light of the specific requirements of FCL. A summary of their main features and limitations is given below.

7.4.1 Simulation language options

SIMAN, supplied by Rapid Data is a general purpose Fortran based simulation language. Features of the system include extensive statistical functions and a memory allocation facility which is useful for testing large models. Whilst supporting both continuous and discrete event logic, it is primarily intended to be used for process or transaction based discrete models. The package does not allow animated graphical representation of the model which would severely limit the way in which the results are presented. In addition, the verification phase would be made more difficult.

ECSL, produced by CLE.COM Ltd. is a very powerful programming language for general purpose simulation. It embodies a complete language (rather than a subroutine library that must be used with a host language like fortran). The many features include ; a host driven graphical display, sampling routines, random number streams, automated program documentation and excellent debugging facilities. An optional program generator called CAPS (Computer Aided Programming for Simulation) is also available.

The system can be run on a 16 bit CP/M or MSDOS computer, but the capacity of the system was limited for the size of model required by FCL. In addition to this the execution speed was slow because *ECSL* is an interpreted rather than a compiled language. The method adopted to drive the graphics (largely handled by the host computer) further reduced the processing speed.

PE Information Systems Ltd. offered a visual interactive simulation system called *HOCUS*, which builds models directly from an inputed activity cycle diagram. Consequently no conventional programming was required. This simulation concept allows models to be developed rapidly and easily understood. There were also special optional Fortran subroutines available from PE that were designed to increase model flexibility and efficiency. The main limitation of this package (at the time of investigation) was the lack of graphical representation.

FORSSIGHT is supplied by Istel but was originally developed for British Steel. It is a visual interactive simulation language. The facilities included extensive statistical analysis, simulation of continuous movement of items on a display and the control of the display time scales to real time. In addition, report generating facilities were available with histograms and graphs etc. Its many features included high resolution graphics, slow down/speed up facility and a display recall facility. At the time of investigation however, Istel were planning to phase out *FORSSIGHT*.

SEEWHY was the first visual interactive simulation system which fully integrated the logic of a simulation model with the graphical representation of the real world. In line with this, its supplier(Istel) claimed that it was well suited to the simulation of production systems. Many advanced modules are available for the collection of statistics, attribute setting and entity definition. In addition the package included a program generator, EXPRESS, whose flexibility was more than adequate for the requirements of the FCL model. The cost of the package was however, substantially higher than its rival simulation language, *OPTIK*.

OPTIK was written and supplied by Insight International and offered the most effective cost/benefit solution to FCL's simulation requirements. The package provided a general purpose visual interactive simulation system. Like *SEEWHY*, *OPTIK* is a Fortran based package. The user written routines are used to manipulate the numerous *OPTIK* libraries. *OPTIK* is a modular package designed around *OPTIK 1* (Visual Interactive Utilities) and *OPTIK 2* (Relational Database). These two modules form the foundation for the simulation module, namely *OPTIK 11*. The package can be used to build models of extremely high complexity and would more than satisfy the requirements of FCL in this respect.

Extensive statistical and reporting facilities are provided for the presentation of the results. The system also provides numerous interactive facilities which can either be utilised during the execution of the program or as an extension to the operating system. These facilities are particularly useful during the debugging phase of the model development. The powerful graphics capability with multi screen output and interactive functions for on screen data input have been designed for simple yet effective presentation purposes.

OPTIK 11 has been developed using the 3 Phase approach to simulation modelling as proposed by Tocher (described in section 4.3) and as such produces models which run highly efficiently. Although the system was not inherently user friendly the subroutine names and argument structures were fairly straight forward.

7.4.2 Model internal options

In designing the program routines for the FCL model the factors that must be included should arise directly from the questions that are to be answered. For this reason, the range of options were confined to those aspects relevant to the performance orientated statistics which are necessary for the appropriate evaluation of policy combinations.

The representation of any work centre and its associated processes can be considered at a number of different levels of complexity.

At the lowest feasible level, a 'black box' representation which simply simulates the input and output characteristics would provide meaningful data about work centre utilisation. It would not however, be able to assess the efficiency of individual processes. This could be reconciled by elevating to the next level of complexity and representing the work centres with 'black box' processes.

Further sophistication is obtained by including process details. The time it takes to perform a particular process is made up of individual time elements. Primarily, time is required to set up the machine for a particular job. Then a certain amount of time is taken to test a single part on the newly set machine. Finally, when the operator is satisfied that the machine is operational, each part is individually processed.

The level at which the work centre is modelled will be influenced not only by the required integrity of performance data, but also by the level of detail at which jobs are modelled. Jobs can either be modelled at batch level or component level. Jobs in the real system spend most of the time as process batches (this is not necessarily the same as the order quantity) as they progress through the manufacturing system, being only separated at processes which operate on one part at a time. If, to replicate this, components are modelled separately then there would be a requirement for thousands of transient entities in order to trace their individual progress through the system model.

The number of entities required could be substantially reduced by representing the jobs as complete batches. Each batch would however, need to be associated with the relevant data so that its current position and state could be easily identified. This data would need to include information about; batch size, the number of components processed, the current work centre at which the components are being processed and the current operation number. This data would also be essential for the gathering of statistical information on which the performance criteria should be based.

A further issue relating to the storage of information and data for the efficient flow of jobs through the system model is raised here. A single data base could be used as the model 'look up' table. This would provide an easy means of locating required data for jobs at specific times during the course of the simulation run. Due to the immensity of data however, this method is associated with slow accessing speeds and also slow data analysis which progressively worsens as the access frequency increases. A variation of this method would be to label the batch entities with a number of specifically defined attributes so that the relevant data (as suggested above) stays with the batch as it proceeds through the system model. This option would relieve the processing associated with the 'look up' table so that it would only need to contain fixed data such as process times, set times, route details and product structure.

Modelling each machine process as an individually coded routine would necessitate the generation of a lengthy and tedious simulation program. A method of modelling the individual characteristics of each machine without the overhead of program code replication would be pertinent here. Investigating the real system showed that each machine process exhibited similarities in job processing characteristics. These characteristics can be outlined as follows;

i) Identification of a batch of work to be processed from the work centre input queue.

ii) A period of time allotted to setting up the process given that the required resources are available.

iii) The sequential processing of each part in turn until the process batch is complete.

iv) Movement of the completed batch of work to the work centre output queue.

The excessive use of program code could be avoided by exploiting the identical characteristics of all processes and writing general machine process routines. These routines would provide all the necessary process elements, but would only take on the role of a particular machine process when called to do so during the course of the simulation. This method of machine process representation would also reduce the simulation run time.

7.5 Information transfer options

In a study of this kind, system information such as job launch data, part process times, operation details and experimental parameters etc. are used extensively to drive the model. The resulting data transfer can either be achieved manually or automatically through software control. Manual transfer would be possible but would limit the total number of experiments and thus accrued benefit to FCL. Additionally, experimentation time would be restricted to normal working hours.

Automated transfer would ensure that experiments are confined to periods outside normal working hours thereby minimising the impact on peak computer operations. In addition, since model run speed was of prime importance, the time taken for this valuable information to be transferred between the various parts of the model would be kept to a minimum. If the whole FCL model was resident on a single computer then the task would be one of efficiently programming the parameter structures for ease of data transfer.

For the case where both TMS and the shop model run on separate micros there would be a requirement for communication hardware and, in some cases software. The communications package would be configured so that one of the machines (the host) emulates the protocol of the other, i.e. the host converts machine parameter settings so that they are both compatible. The package would also perform house keeping functions such as data error detection and hand shaking.

If, however the production control system model and the manufacturing system model are both resident on separate machine types (i.e. the Burroughs mainframe and the TDI Pinnacle microcomputer respectively) then the problem of data communication must be considered very carefully. A number of alternative approaches were available to achieve data communication between the Borroughs B5900 and the TDI Pinnacle. These options took into account the existing machines at FCL and included ;

i) TDI<> APRICOT<> BORROUGHS

- Required a program on the Apricot

ii) TDI<>IBM AT<>BORROUGHS

- Required an IBM AT screen emulator

iii) TDI<>BURROUGHS

- Required a TDI screen emulator

In addition to the above requirements each option needed a dedicated software program on the Borroughs mainframe and the provision of a B5900 upload facility.

The communication task also becomes more complicated when this approach is considered for two main reasons ;

a) Different machine character coding systems. The Pinnacle uses ASCII whereas the B5900 conforms to the EBSIDIC standard and uses the Unisys proprietary twisted pair terminal lines.

b) The B5900 communicates with its peripherals using a poll/select protocol system. In this way the computer systematically scans all of its output channels and responds to any messages sent by any other connected machine. The Pinnacle was not designed to accept a polling signal from the B5900. Communication hardware systems were available which were designed to form an asynchronous link with poll select and protocol conversion.

The required communication software would be used to determine the way in which data is transferred, and as a user interface to input data transfer requirements.

Options i) and ii) were considered to be potentially the most cost effective to implement. Their common drawback however, was in the need to use a slave translator (Apricot or AT) which, although available, would increase experimental run time considerably. Furthermore control of the slave could limit the scope of future experiments.

Option iii) was considered to be potentially the highest cost solution. However, direct Pinnacle to B5900 communication afforded a number of operational advantages, the greatest of which was increased experimental capacity. In addition this configuration could well be a prerequisite for future 'What if' and 'Predictive order launching' models.

8 The FCL Production System Model

8.1 Introduction

The model developed for this research has been designed with a fully functioning MRPII system and a comprehensive manufacturing simulator in keeping with the total system modelling approach discussed in chapters 5, 6 and 7. The following discussion details how this has been achieved both in method and concept, and highlights the major decisions made during the course of the project.

It was agreed that a model which used the Borroughs B5900 mainframe to represent TMS and the TDI Pinnacle super-micro to simulate the manufacturing system would offer the best solution for the requirements of FCL. Consequently the use of two dissimilar machines dictated the employment of a comprehensive communication package consisting of both standard hardware, namely the P1000 protocol converter, and specially written interface software.

In order to make the model a feasible option it was clear from the start that a sizeable database reduction would be necessary to minimise model run processing time(as specified in chapters 6 and 7). At the same time it was important to maintain a sufficient portion of the database so as not to invalidate the models interpretation of the effect of the parts in the real system. The model elements that could be affected include the distribution of set-up times, the sequence in which jobs flow through the shop and the shop product mix. This prompted a detailed analysis of FCL's past market.

In determining the parts to be modelled the decision was made at this time to use information from the real market place, which had been collected over a period of three months(September 1986 to November 1986). Clarke(1988) documents the three month MPS as seen by the production scheduler at the beginning of the horizon in 1986. As each week passed the changes to the MPS were recorded, to be later used as data for the part of the model that would represent the actual uncertainty in the MPS. This exercise initially reduced the database to 38 end level items, since these were the only ones that were MPS'd over the whole 3 month experimental period.

A further database reduction was made by analysing the total demand for each end item over the experimental period and only modelling those products whose effect on the system were substantial. Table 8.1 shows the PCB demand as a percentage for each item. The analysis showed that 21 of the live end items(indicated with an M) represented about 93% of FCL's total demand in terms of PCB's.

Purchased components represent a large percentage of the database and consequently account for much of the data processing resource. For the purpose of the model, all purchased components were ignored on the basis that FCL were striving to achieve 100% fluidity on the shop floor (i.e. consistent availability of sub-components whenever required to build parts). Supporting discussions argued that if purchased components were always available then their effect on the system performance of the shop could be considered as minimal, and therefore irrelevant in this study. This assumption dramatically reduced the model database and in

SYSTEM TYPE / PARTNUMBER	No. PCB's	PART DEM'D	PCB DEM'D	% PCB'S AGAINST TOTAL	
SYS 1					
AEQP-V/DATA-102	8	400	3200	8.43	M
3HAA-00030AAL 01	17	345	5865	15.45	M
SYS 2					
ADUM-PLANNING-01	1	1030	1030	2.71	
3HEB-00739AAD 01	1	1030	1030	2.71	M
3HEB-00741AAF 01	1	1030	1030	2.71	M
SYS 3					
3HEB-00753AAT 01	1	2200	2200	5.79	M
3HEB-00756AAW 01	1	100	100	0.26	M
3HKA-00153AAN 01	2	1700	3400	8.95	M
SYS 4					
3HEB-00738AAC 01	1	2000	2000	5.27	M
SYS 5					
ADUM-AHS-102	8	24	192	0.51	M
SYS 6					
3HER-00001AAJ 02	8	8	64	0.17	
3HER-00001ABK 02	10	56	560	1.47	M
SYS 7					
ADUM-EDGELEY-103	45	11	495	1.30	M
3HEF-00099AAH	5	600	3000	7.90	M
SYS 8					
ADUM-ECTE-102	11	20	220	0.58	M
3HEM-00029AAZ	4	36	144	0.38	M
3HEF-00122AAT 01	3	14	42	0.11	M
SYS 9					
3HEF-00094AAP 02	17	284	4828	12.72	M
ADUM-GATEWAY-103	17	45	765	2.01	

SYSTEM TYPE / PARTNUMBER	No. PCB's	PART DEM'D	PCB DEM'D	% PCB'S AGAINST TOTAL	
SYS 10					
ADUM-QUERC-102	42	47	1974	5.20	M
3HEF-00067AAD 02	5	900	4500	11.85	M
SYS 11					
ADUM-QUADMUX-101	10	20	200	0.53	M
SYS 12					
ADUM-SCOUR-102	11	5	55	0.14	
SYS 13					
3HEF-00043AAP	7	80	56	0.15	M
3HEF-00044AAQ	6	80	480	1.26	M
AEQP-SSTV-105	-	-	-	-	
AEQP-SSTV-106	-	-	-	-	
SYS 14					
3HEA-00033AAP 01	4	4	16	0.04	
3HEA-00033ABP 01	5	5	25	0.07	
3HEA-00033ACQ 01	5	5	25	0.07	
3HEA-00033AFT 01	5	15	75	0.2	
3HEA-00033AGU 01	5	11	55	0.15	
3HEA-00034AAV 01	4	2	8	0.02	
3HEA-00034ABW 01	6	16	96	0.25	
3HEA-00034ACX 01	6	2	12	0.03	
3HEA-00034ADY 01	4	15	60	0.16	
3HEA-00034AEZ 01	6	28	168	0.44	
3HEA-00034AFA 01	6	-	-	-	
	Total PCB demand		37970		

M = Modeled products

Table 8.1 Analysis of expected PCB demand for all live FCL products addition reduced the potential model processing time to the extent that thorough experimentation then became feasible.

On completion of the analysis it was found that a representative database of 240 parts (97 assemblies/sub- assemblies and 143 PCB's) would be satisfactory for a study of this nature. Appendix I. shows the list of modelled part numbers. The first 20 characters of the TMS model numbers are duplicated from the live database part numbers. The final 4 characters represent the model reference number which was used to distinguish between parts during simulated manufacture.

8.2 Data Gathering

Having determined the actual parts to be modelled and defining them in the database as specified in chapter 7, it was then necessary to record the relevant model processing information for each part as follows;

i) Process time

ii) Process set-up time

iii) Part operation details

iv) Product structure or BOM

From the stand point of time and resource, collecting and analysing input data of this kind is a major task. Regardless of the sophistication of the model, data that is inaccurately collected, inappropriately analysed or not representative of the environment, will inevitably lead to model outputs which may be misleading and possibly damaging or costly when used for policy design or decision making. For this reason each item of data was carefully collected and tested.

The shop floor operators are measured against standard hours at each work centre process. These standard hours had previously been implemented using synthetic MTM times and were representative of the actual time it takes to process a particular job at a given work centre, due mainly to automated machines. Original discussions suggested that it would be necessary to apply a statistical distribution to each of the process times based on histograms of the data. Further discussions lead to the conclusion that although the facility existed to investigate the distribution of process times, it would not be utilised for our current purposes. This assumption was accepted on the basis that the effects of process time variation would diminish as the simulation progressed. Interactions between product mix, order timing and resource constraints, it was argued, would have a far greater influence on system performance. The model process times were subsequently implemented using the actual standard hour data. The A9 database contained the MTM process times for each part being modelled. This data was duplicated on to the test database.

At the time of writing, process set-up times were not being recorded on the A9 database. Since the model required set-up time information a method of estimating this data was necessary. Working closely with the shop manager, an exercise which involved categorising each

Work Centre	Description	No. of M/C's
1	Dual in-line-package. DIP insertion(CNC).	1
2	Axial component insertion(CNC).	1
3	Radical component insertion(CNC).	1
4	Hand insertion & mask.	5
5	Semi automatic insertion with cut and clinch(NC).	6
6	Semi automatic insertion without cut and clinch(NC).	4
7	Inspection 1.	2
8	Flow solder(NC).	2
9	Hand insertion and finalise.	10
10	Inspection 2.	3
11	Automatic test equipment (ATE)	4

Table 8.2 Work centres used for the manufacture of PCB's

	SET TIMES PER PCB TYPE		
WORK CENTRE	Small	Medium	Large
1	40	55	70
2	35	45	55
3	40	55	70
4	5	10	15
5	57	91	125
6	57	91	125
7	5	5	5
8	10	15	20
9	6	8	12
10	5	5	5
11	5	10	15

Table 8.3 Operation set times per work centre (mins.)

modelled PCB as small, medium or large was conducted (assemblies and sub- assemblies do not require a set-up time at FCL). These groupings were based on physical PCB size and number of components per board. This information formed the basis of the work station set-time data for the model which was estimated by personnel who were familiar with the processes in question. The work centres included in the analysis are shown in Table 8.2 along with the model numbers used to distinguish them. As an example, a PCB categorised as large may require an estimated set-up time of 50 minutes whereas a small PCB may only need 10 minutes at the same workcentre. These estimated times are stored in the test database against each appropriate part at its respective operations. Table 8.3 lists the set times allocated for each work centre against PCB category.

Details relating to the route for each part were again available on the A9 database. This data was duplicated on to the B5900. The route for each part is included in Appendices II. and III. Operations involving the processing of purchased components only, such as pre-forming and component sequencing were not included on the B5900, since purchased parts were not being modelled.

The product structure for each of the modelled items was resident in the BOM module of TMS on the A9. The multilevel BOM duplication onto the B5900 was necessary to PCB level only.

8.3 The Production Control System Model Element

A full description of the TMS model element is given in Appendix VI. The following discussion relates to the particular approach adopted.

TMS is an integrated information system comprising(in its original form) of 10 major modules and 8 supportive modules. In addition 2 associated modules with full interfacing are also available.Figure 8.1 shows a schematic representation of the functional links between each of the modules. For the purpose of the FCL model it was found that only the following 5 modules of TMS would be necessary to represent the production control system;

i) Engineering Data Control (EDC)

ii) Materials Requirement Planning (MRP)

iii) Stock Control (STK)

iv) System Support Facility (SSF)

v) Work in Progress (WIP)

The omissions were made since these modules influenced factors which were outside of the model boundary. Their inclusion was also unjustified since it would only constitute a further overhead.

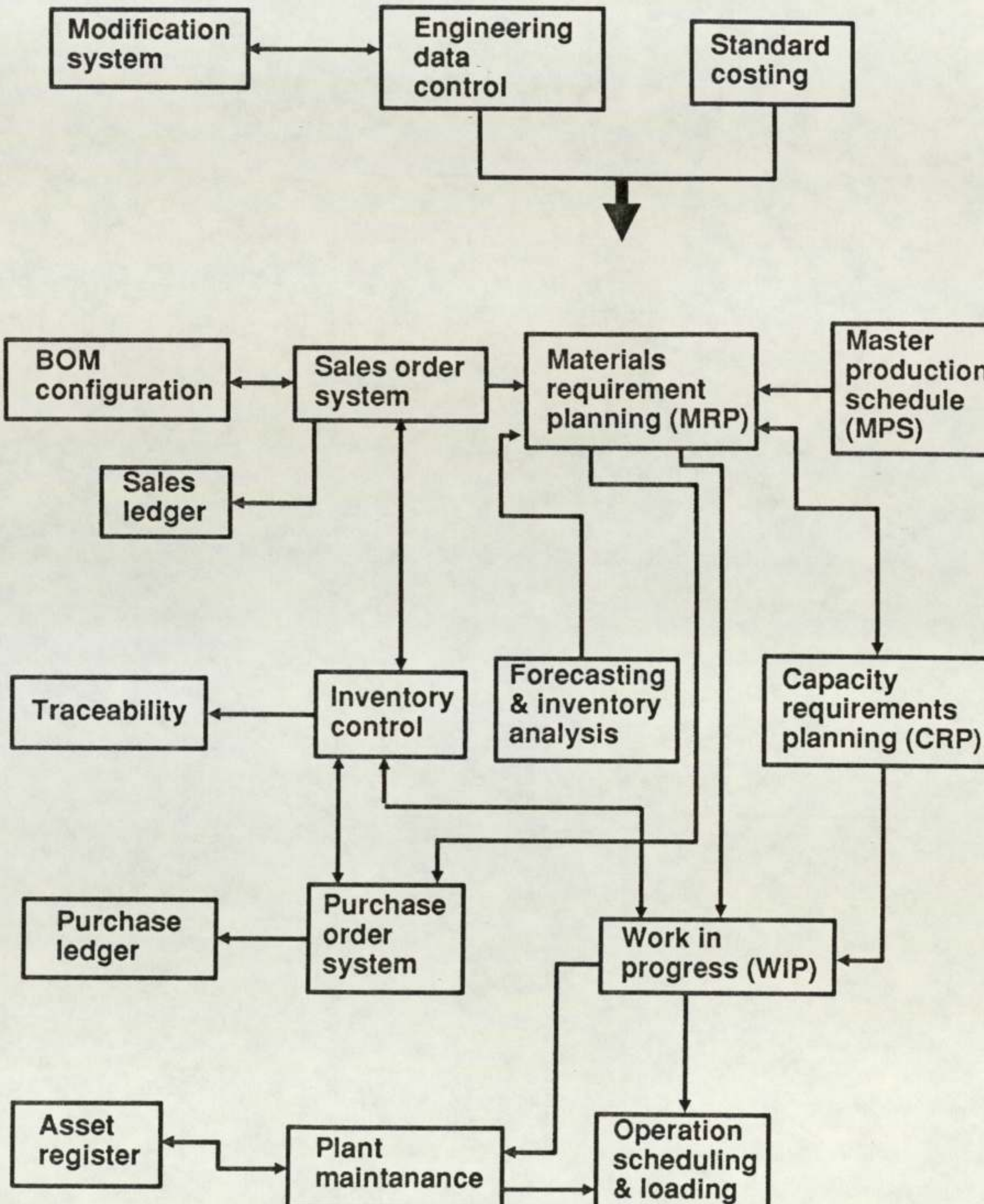


Figure 8.1 Schematic diagram of the TMS module links

The WIP module was utilised the least since most of its functions represented what was happening in the manufacturing facility and were therefore already modelled in the manufacturing simulator. The WIP module actually performed five functions ;

i) creation of a production plan by manipulating the MRP output ;

ii) release simulation - Simulates the potential component shortage situation if a certain plan is launched into production (pre- allocation) ;

iii) release - Actual release of a production plan for the current week. This may optionally :

allocate component stocks,

allocate jigs/tools as relevant,

produce kit list documents,

produce operation tickets ,

iv) monitor shop floor activity - feedback of activity bookings ;

v) control expediting - provide shortage control and monitor status/progress of batches.

Functions i), ii) and iv) were fully provided by the manufacturing model and were therefore not required on the TMS model. The task of releasing actual orders was also performed in the shop model; however, the allocation of component stock, due to its complexity was calculated using the facility available in the WIP module. The WIP expediting facility was also used for similar reasons.

The planning of requirements is one of the most complicated areas to define in the production control process. The regenerative MRP module simplifies and standardises the planning process and provides accurate requirement calculations. The primary objective of the MRP module is to plan procurement activities for the manufactured items in FCL. The STK module was also implemented to provide an interface for updating and maintaining the necessary planning parameters for each part.

8.4 The Manufacturing System model element

A full description of the shop model element is given in Appendix VI. The following discussion relates to the particular approach adopted.

The purpose of the manufacturing system model is to simulate the work flow from work centre to work centre over time with the view to studying how the manufacturing facility at FCL (namely Modern Manufacture) is affected by the various policy and parameter combinations discussed in chapter 6.

The event based simulation system, OPTIK was chosen to represent the manufacturing system at FCL. The detailed factory simulation model resides on a TDI Pinnacle which is a high speed super microcomputer, using a 1Mb cpu and a 20 Mb hard disc. It runs a Motorola MC68000 chip at a speed of 12 MHz and at the time of purchase was considered to be one of the fastest microcomputers in the world.

The simulation process is monitored on an Intercolor microcomputer screen. A graphical representation of both the PCB manufacturing department and the assembly area have been included.

The assembly area, although modelled in less detail than the manufacturing department serves as an important model requirement. It was primarily used to consume the finished PCB's from the manufacturing department. In addition, sub-component availability was required before an assembly order could be started. Graphically, the assembly area has been emulated using rows of coloured bars which represent the assembly areas and coloured squares on the bars to represent assembly jobs.

Machines in the PCB manufacturing department are represented by appropriately coloured squares. The colours are used to indicate the status of the machine at any time during the simulation run. Yellow is used to indicate that a particular machine is being set up for the next job in the queue. Whilst in this state the machine cannot be used for any other purposes. When a machine begins processing a job its colour changes from yellow to green. A machine idle state is represented by the colour red. Input and output queues are represented as yellow and magenta respectively. The size and content of a work centre queue at any given time, although not shown on the screen, can be determined interactively with the use of special OPTIK routines. This information is used to locate WIP. The amount of current WIP is shown as a digital value at the bottom of the screen. Stores representation is divided into three separate areas. The orders waiting for launch are shown as a list of job numbers to the left of the screen where as the completed PCB's and finished products stores are shown as lists of job numbers to the middle and right of the screen respectively.

The simulation model itself is capable of including the whole product range with their individual routes, operation times and set-up times. The information is stored as specially written OPTIK data packets which can be accessed at any time from the Pinnacle database. In addition each of the work centres and machine processes are modelled separately. The machine elements were designed to be effective at the level where process utilisation could be easily measured. Each work centre has its own unique mode of operation, including setter/operator restrictions and available processing resource. The model divides the time that a job spends at any work centre into four separate time elements. These time elements and their corresponding influences are summarised below;

i) **Queue time**. Each job joins the appropriate work centre input queue which corresponds to the next operation number on the route. The time a job spends in any given queue is primarily determined by the characteristics of the other jobs preceding it. These characteristics include set time, operation time, batch size and job priority. Furthermore, these queues are allowed to increase without restriction so that the resulting average queue lengths can be assessed and used as a basis for identifying bottleneck work centres.

ii) **Work centre set time**. When all required resources are available for a job to be processed then the appropriate work centre is set up by a setter with the correct skill characteristics. The set up time for each job (regardless of batch size) against each work centre is stored in the pinnacle data base.

iii) **Machine process time**. The model assumes that a work centre can only be set up if the appropriate operator is available as well as the appropriate setter. Therefore, as soon as the setting operation is complete, the machine process can begin. The machine element also assumes that each part can only be processed individually. Therefore, the process time is the product of the component operation time (stored in the pinnacle database) and the process batch size.

iv) **Transport time**. When a job has been processed at a particular machine, it is immediately moved to the respective work centre output queue whilst it awaits transport to the next work centre or stores location. The time a job waits in the output queue simulates the transport time, since when it elapses, the job is immediately moved to its next location as prescribed by the route details, which are stored in the pinnacle database. The same average transport time has been used for each of the parts in any work centre output queue. This average value is implemented at the start of a simulation run and is used as an experimental parameter to investigate the effect of different average transport times on the performance of the system model.

Owing to the similarities of job processing characteristics in the real system, the model has been written using general event routines. This method allows the simulation of individual processes without the overhead of excessive program code. A modified activity cycle diagram (ACD) showing the way in which the events and activities of FCL have been linked together in the manufacturing model is shown in Figure 8.2. The shop model comprises two independent simulation cycles which are both driven by suggested orders from the TMS model. The first cycle (on

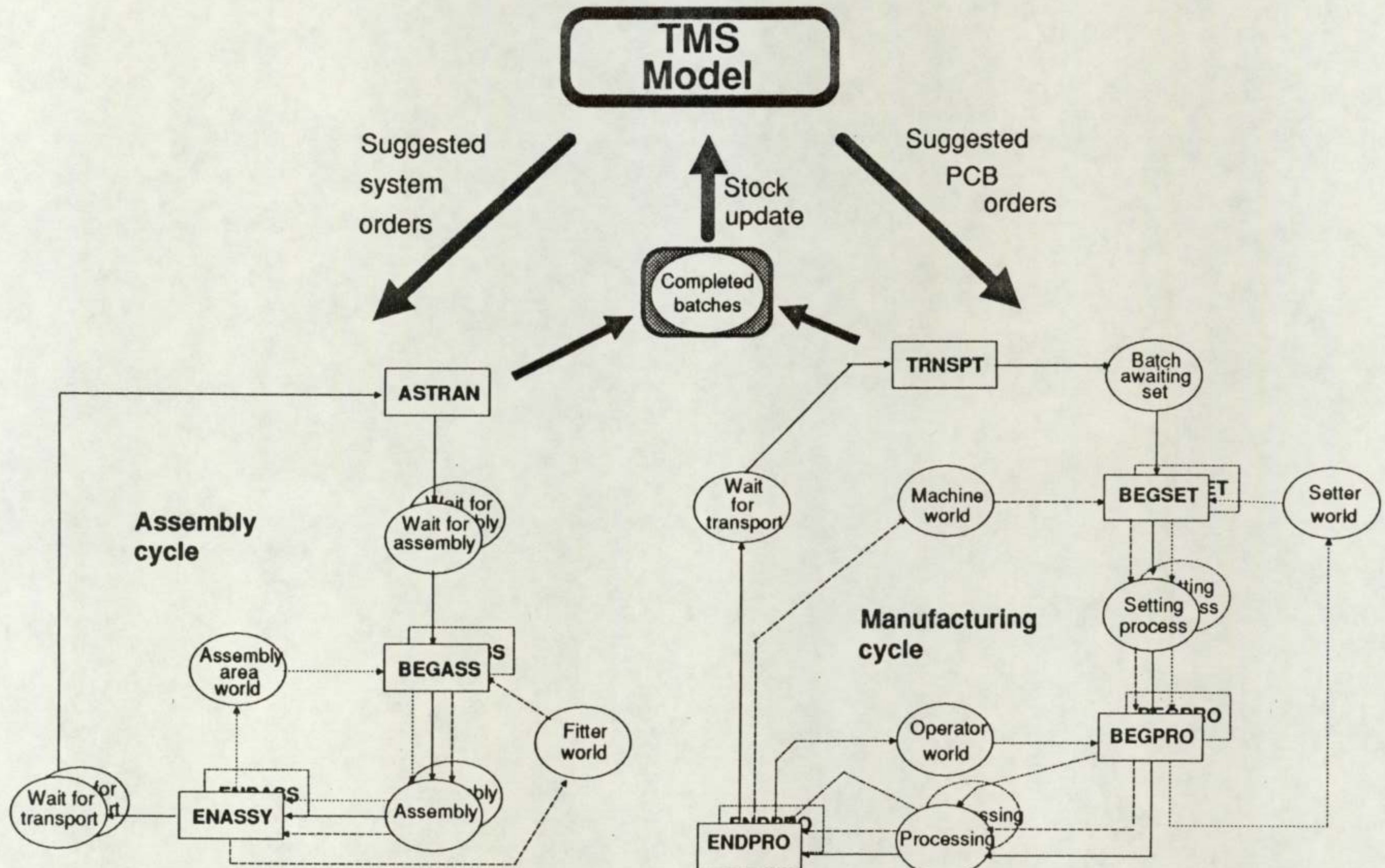


Figure 8.2(a) The modified activity cycle diagram for the simulation of manufacture and assembly

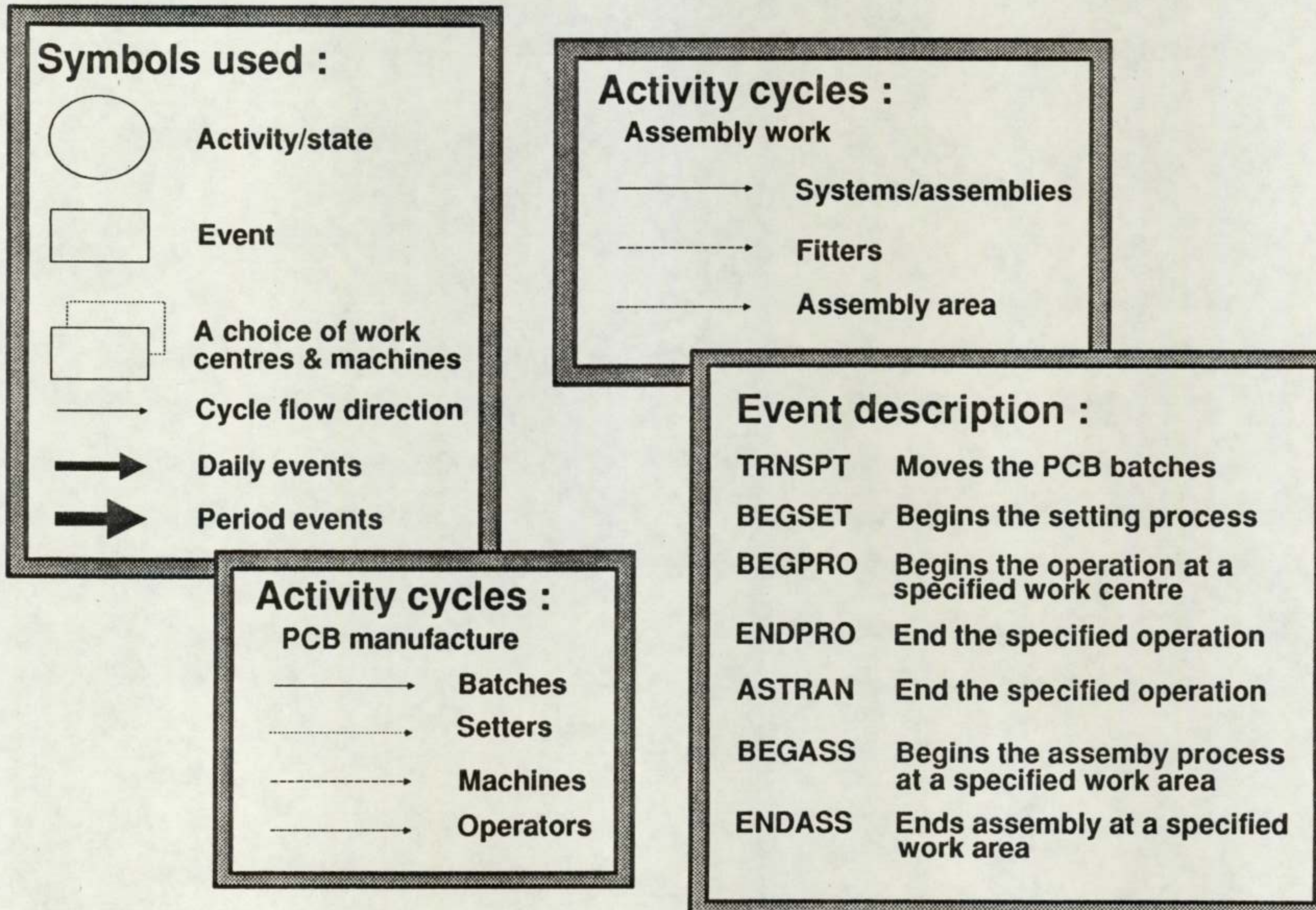


Figure 8.2(b) The key to figure 8.2(a) .

the right of the diagram) represents PCB manufacture, which involves the insertion and fixing of electronic components into a specially designed board. The other cycle represents the assembly processes for individual or combinations of PCB's. These cycles are linked by the completed batches store. This store holds all completed jobs, whether PCB's, assemblies or finished items. Each part is dispatched from the stores to satisfy either an assembly or a customer requirement. In addition the completed batches information is used to update the stock control module of the TMS model prior to the next MRP run.

The Optik 11 software is used to translate this ACD into computer program code (see Appendix V). The resulting simulation model is based on a number of specially written event routines which use entities and lists to represent the various elements in the real system. Jobs are released and allowed to proceed through the manufacturing facility as complete process batch entities. As a consequence of this, each batch entity is associated with relevant information and data so that its graphical location and current state can be identified at any time. This is represented by data attributes and includes the following information;

i) part type;

ii) order number;

iii) batch number to distinguish between split batches;

iv) transfer batch size;

v) launch date;

vi) current operation number;

vii) current workcentre number;

viii) due date.

Machines are also represented by entities which need to be associated with current states and job types being processed. Machine entities therefore store information in the following attribute headings ;

i) workcentre number;

ii) machine number;

iii) batch type for which the machine is currently set;

iv) operation number;

v) display number.(for graphical representation)

Operator and setter entities only need to know whether or not their particular skill resources have been exhausted at any particular time during the simulation run. Therefore, they have been modelled using the following attributes;

setter entity with particular skill;

i) maximum number of setters in this setter world;

ii) current number of setters in this setter world;

operator entity with particular skill;

i) maximum number of operators in this operator world;

ii) current number of operators in this operator world.

Relevant information about fixed data is stored in the form of a database 'look up' table. The actual data used for the FCL model is listed in Appendices I., II. and III. The following lists the data requirements for the manufacturing model;

i) PCB process data;

ii) assembly process data;

iii) assembly pick list data;

iv) model/TMS partnumber translator.

When a number of entities are grouped together they form lists. The lists are used to represent workcentre queues for batches and resource worlds for machines, operators and setters.

Each model event has a unique subroutine whose execution in the model is dictated by one of following event categories;

i) conditional Event;

ii) bound Event;

The reader is referred at this point, to section 4.3 for a detailed description of i) and ii). The manufacturing model comprises 2 conditional events and 8 bound events. The following discussion details these events and describes how they contribute to the working of the FCL model. In the discussion the author assumes the word 'machine' to refer to any manufacturing activity performed in the manufacturing facility.

8.4.1 Conditional Event Description

SUBROUTINE BEGSET

Determines whether or not a setting process can begin at a particular machine by continually testing the following four conditions for each of the workcentres ;

i) check for waiting jobs in the workcentre queue;

ii) check for availability of a suitable machine to process the job;

iii) check for availability of setter resource with correct skill;

iv) check for availability of operator resource with correct skill.

If all the conditions are satisfied then the BEGSET routine can be executed. Appropriate changes are made to the machine, setter and operator attributes to reflect the fact that they are now tied up with a specific job. The appropriate machine attributes are also changed to show which job is currently being processed and what operation number this constitutes. It was found that bottleneck work centres were particularly prominent in the PCB manufacturing area. Therefore in the model, the machine input queue is organised so that fewer set-ups are performed if the machine is a bottleneck. This bottleneck sequencing concept can be visualised by considering Figure 8.3. The black box in Figure 8.3a represents a machine which is currently processing a job, C555, for example. When the job is complete, under normal operation, the next part type for which the machine needs to be set would be A111. If however, the machine becomes a bottleneck and no more jobs of the same type are at the front of the queue, then a bottleneck sequencing algorithm controls the flow of work through this machine. At the completion of each job the algorithm searches for any other jobs of the same type which may be in the queue. If a job is found of the same type then this one is processed next, thereby saving a machine set-up and consequently increasing the dynamic capacity of that particular workcentre. The algorithm has no effect if there are no jobs of the same type in the queue.

This decision process is illustrated in figure 8.3b. Under the normal mode of operation, job B333 would follow A111 however, in the bottleneck sequencing mode A111 becomes the next job to be processed.

Once the next job for processing has been determined, the Pinnacle database is accessed in order to retrieve the set time allocated to that part number at the current machining operation. This data is used to schedule the time at which the setting process will end and the subsequent machining operation begin, since a job at this stage does not need to wait in a queue. The event scheduled for the end of the set time is BEGPRO.

The order in which workcentres are chosen to be tested for the possible execution of a setting process is fundamentally important ; were the workcentres to be always selected in a given order then those workcentres nearer the end of the list would not get a fair chance at utilising the limited operator and setter resource. In an attempt to reduce this bias, a random number generator is used to change the order in which these workcentres are selected. This order change procedure is performed each time the BEGSET routine executes successfully. This new order is used for subsequent BEGSET executions.

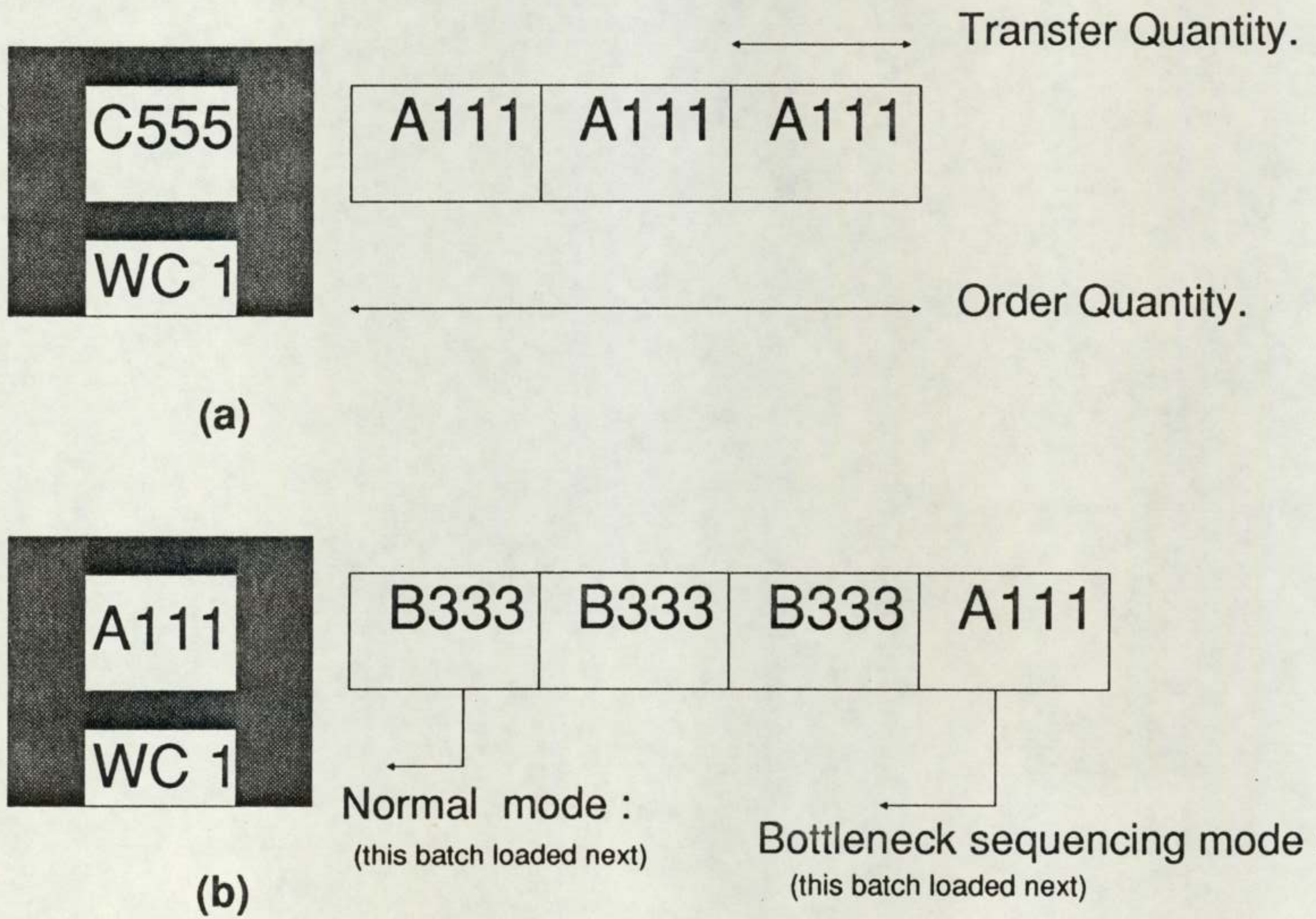


Figure 8.3 The bottelneck sequencing concept

The final step in this routine is to modify the graphical representation of the machine to indicate that it is currently being set up. The colour adopted for this purpose is yellow.

SUBROUTINE BEGASY

Determines whether or not an assembly process can begin at a particular work area by continually testing the following three conditions for each of the workcentres;

i) check for waiting assembly jobs in the workcentre queue;

ii) check for availability of a suitable work area to assemble the job;

iii) check for availability of operator resource with correct skill.

Notice that since there are no machines or processes to be organised in the assembly department of FCL, a setter resource is not required.

If all the conditions are satisfied then the BEGASY routine can be executed. Appropriate changes are made to the working area and operator attributes to reflect the fact that they are now tied up with a specific job (i.e. the next one in the queue). The appropriate work area attributes are also changed to show which job is currently being assembled and what operation number this constitutes. Once the next job for assembly has been determined, the Pinnacle database is accessed in order to retrieve the assembly time allocated to that part number at the current work area. The total assembly time is calculated by multiplying the part assembly time with the transfer batch size. This data is used to schedule the time at which the assembly process will end. The event scheduled for the end of the assembly time is ENASSY. A random number generator is again used to change the order in which these workcentres are selected for testing. This order change procedure is performed each time the BEGASY routine executes successfully. This new order is used for subsequent BEGASY executions. A job assembly process is graphically represented in the model's assembly area by a block of blue on a cyan background.

8.4.2 Bound Event Description

SUBROUTINE TRNSPT

This routine is responsible for all PCB job transportations between stores and workcentre and between each workcentre. The routine is initially scheduled for the time when each job is to be launched in to the system (this is also the method used to start off the simulation run). Subsequent scheduling of TRNSPT is done by the bound event, ENDPRO.

The batch entity to be transported is determined by interrogating information carried over from the particular event which scheduled the current TRNSPT. This information holds details about the previous workcentre number, part type and next operation number. The next workcentre number is retrieved from the Pinnacle database (PCB route details) using the part type and the next operation number for precise location of the data. The appropriate batch entity attribute is reset to this new workcentre for subsequent route information.

Based on this data, TRNSPT then ascertains whether or not the previous operation was the last one. If the previous workcentre indicates that this was the last operation, the batch is moved to the completed parts stores. If however, the job requires more operations, then the batch is transferred to the next workcentre queue and waits there until it can be processed. This will occur when all the conditions of BEGSET are satisfied for this particular job.

SUBROUTINE BEGPRO

The BEGPRO routine is responsible for representing the machine process for all PCB jobs. The batch entity to be processed is determined by interrogating information carried over from the scheduling event BEGSET. This information includes details about the next workcentre, and the machine which has just been set up for the current part type.

Appropriate changes are made to the machine, setter and operator attributes to reflect the fact that the current machine is in the state of processing a specific job.

The part process time is retrieved from the Pinnacle database (operation times) using the part type and the next operation number for precise location of the data. The total processing time is calculated by multiplying the part process time with the transfer batch size. This data is used to schedule the time at which the machine process will end. The event scheduled for the end of the assembly time is ENDPRO.

The final step in this routine is to modify the graphical representation of the machine to indicate that it is currently processing a job. The colour adopted for this purpose is green.

SUBROUTINE ENDPRO

At the completion of each process (ie when the simulated time reaches that of the total process time determined in BEGPRO) the event ENDPRO is executed.

Appropriate changes are made to the batch, machine and operator attributes to reflect the fact that the current machine is in the state of being idle and make resource available for future jobs.

The routine then schedules the event TRNSPT for a time in the future(PCB transport time) dictated by a variable parameter determined at the commencement of the simulation run. The final step in this routine is to modify the graphical representation of the machine to indicate that it is currently in an idle state waiting for the next job. The colour adopted for this purpose is red.

SUBROUTINE ENASSY

At the completion of each assembly operation (ie when the simulated time reaches that of the total assembly time determined in BEGASY) the event ENASSY is executed.

Appropriate changes are made to the batch, work area and operator attributes to reflect the fact that the current work is in the state of being idle and make resource available for future assembly jobs.

The routine finally schedules the event ASTRAN for a time in the future(assembly transport time) dictated by a variable parameter determined at the commencement of the simulation run.

SUBROUTINE ASTRAN

This routine is responsible for all assembly transportations between stores and work area and between each work area. The routine is initially scheduled for the time when each job is to be launched in to the system (this is also the method used to start off the simulation run). Subsequent scheduling of ASTRAN is done by the bound event, ENASSY.

The batch entity to be transported is determined by interrogating information carried over from the particular event which scheduled the current ASTRAN. This information holds details about the previous workcentre number, part type and next operation number. The next workcentre number is retrieved from the Pinnacle database (assembly route details) using the part type and the next operation number for precise location of the data. The appropriate batch entity attribute is reset to this new workcentre for subsequent route information.

Based on this data, ASTRAN then ascertains whether or not the previous operation was the last one. If the previous workcentre indicates that this was the last operation, the batch is moved to the completed part stores. If however, the job requires more operations, then the batch is transferred to the next workcentre queue and waits there until it can be processed. This will occur when all the conditions of BEGASY are satisfied for this particular job.

8.5 The total FCL model

The bound and conditional events described in section 8.4 are used to simulate the actual manufacturing and assembly processes in the manufacturing facility. During the simulation run however, it is required that at specific times, access be made to the TMS model in order to interface the two and make one total model. This has been achieved in the manufacturing model by the use of a specially written routine called ENDDAY, which the author has termed a procedural bound event and in the TMS model by a number of job procedures called Work Flows.

8.5.1 The manufacturing model's view

The event ENDDAY in the shop model determines the activity of the model between simulated days by controlling the execution of a number of subroutines. The procedural flow of each of these various subroutines is shown in figure 8.4. The following describes this event and it's associated routines.

SUBROUTINE ENDDAY

At the end of each simulated day this event is scheduled. Its primary purpose is to carry out all the appropriate procedures which occur at this time, namely;

i) the updating of all the appropriate stock levels and writing each modification to the disc on the Pinnacle using the routine called WIPCLS;

ii) the collection of all stock level statistics;

iii) the reinitialization of all idle batch entities for future use in the model when creating new orders.

At this point ENDDAY checks if this is the end of the period as well as the end of the day. If the check is confirmed then the subroutine ENDPER is called. If however, the check results in a negative response, (ie end of day only), then the subroutine STTDAY is called. The later subroutine performs those activities which occur at the start of the day, ie ;

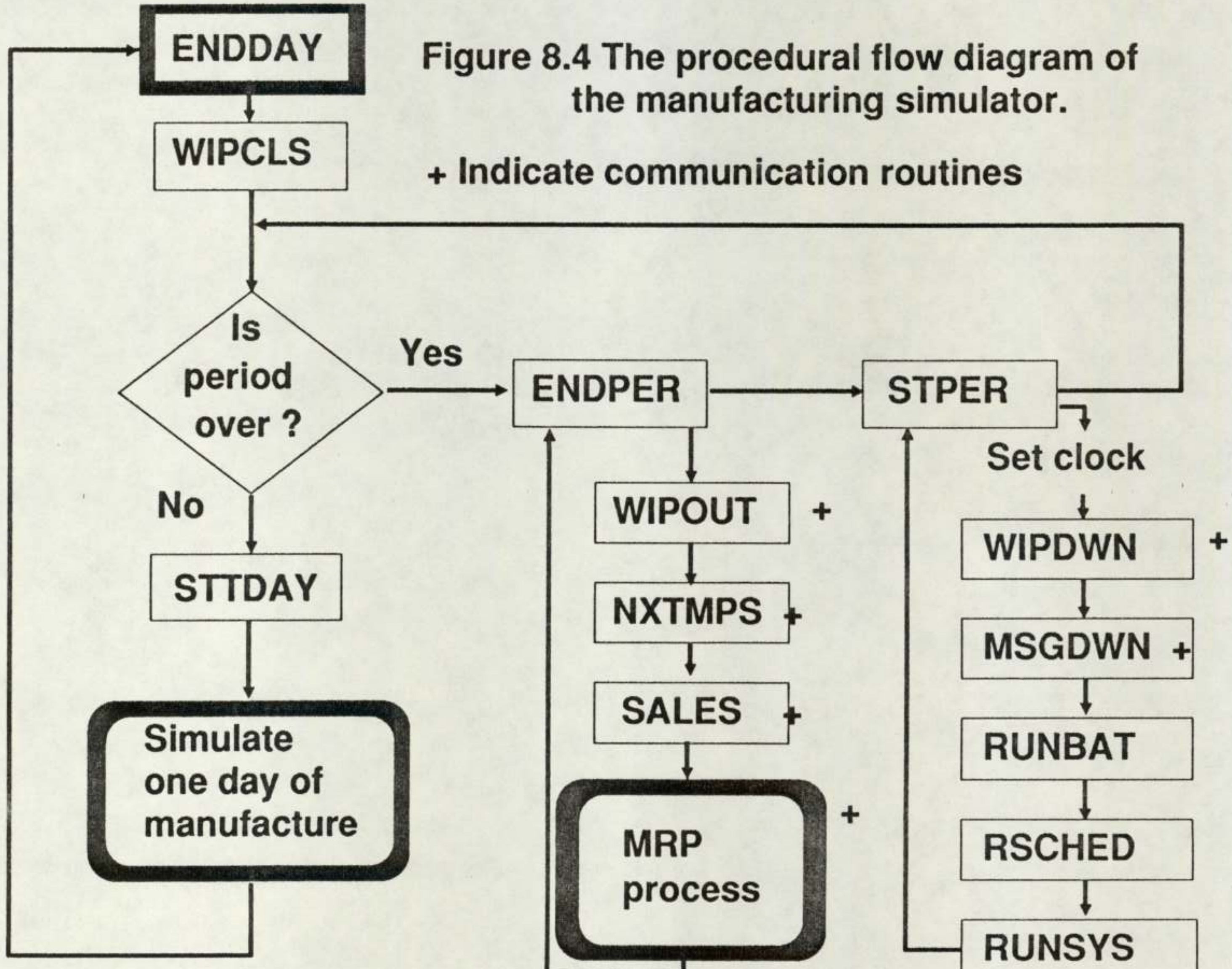


Figure 8.4 The procedural flow diagram of the manufacturing simulator.

+ Indicate communication routines

i) increment the day counter, and

ii) the launch of all assembly batches which are due for now and have sufficient component stocks.

SUBROUTINE ENDPER

Activities that occur at the end of each period are included in the subroutine ENDPER. This routine begins by reading all the completed stock information from the Pinnacle and uploading the appropriate part type with the completed quantity to the TMS model database using the subroutine, WIPOUT. Each experimental period has a separate MPS change file associated with it. Each file is made up of; part type, period concerned and new MPS quantity. After successful upload of the stock data, the MPS is modified in the TMS model by uploading the appropriate file which contains all the changes which will have occurred during the period and up to the point of the next MRP run. This upload is achieved by invoking the subroutine, NXTMPS.

Any sales which are due to occur in each period are held in individual files. Each file holds the part types and quantities that are sold in each respective experimental period and are uploaded to the TMS model in the subroutine, SALES. This information is also used to reduce finished stock levels in the MM model as well as the TMS model.

SUBROUTINE STPER

The subroutine, STPER is called at the completion of an MRP run on the TMS model. All activities that occur at the commencement of each period are included in subroutine STPER. Before the communication routines are invoked the following tasks must be performed ;

i) increment the period counter to the next period;

ii) determine the number of days in the next period using the appropriate shop calendar. In this way the model takes into account public holidays and FCL shut downs;

iii) schedule the event ENDDAY to occur in 480 minutes from now.

All the suggested orders and action messages are downloaded from TMS and written to specific periodic files using subroutines, WIPDWN and MSGDWN respectively.

All suggested orders are converted into planned orders and used to make up the manufacturing model's production plans for the appropriate departments. The subroutine, RUNBAT generates the production plan for the manufacturing department, whereas RUNSYS is used to generate the assembly production plan. Both plans are made up of batch entities.

These production plans are organised into launch date order and the appropriate scheduling of the events TRNSPT and ASTRAN for each planned order is done. Again all transportation times are dictated by a variable parameter which is determined at the beginning of each experimental run.

The action messages downloaded from TMS are used to delay and expedite suggested orders which have not yet been launched in the MM model. This rescheduling mechanism is incorporated in the subroutine, RSCHED.

The schematic flow diagram in figure 8.5 shows the total FCL model run cycle from the view point of the manufacturing simulator.

At specific points in time during the simulation run the bound event, RECORD is scheduled. This routine is responsible for collecting the statistical observations of the model. Each observation is time weighted in order to minimise any bias caused by early transient behaviour. The regularity of the event RECORD is determined by a variable parameter which is entered at the beginning of the experimental run. The following model elements are observed ;

i) batch size;

ii) shop floor WIP;

iii) job flow time;

iv) due date accuracy;

v) workcentre utilisation;

vi) workcentre input queue;

vii)inventory stock.

The final routine to be executed in any experimental run is the bound event, ENDED. It is scheduled to occur at a specified time in the future by a variable parameter which is entered at the beginning of the experimental run. The routine performs statistical analysis of all the observed data collected by RECORD. The minimum, mean and maximum values of each data item are then written to a file

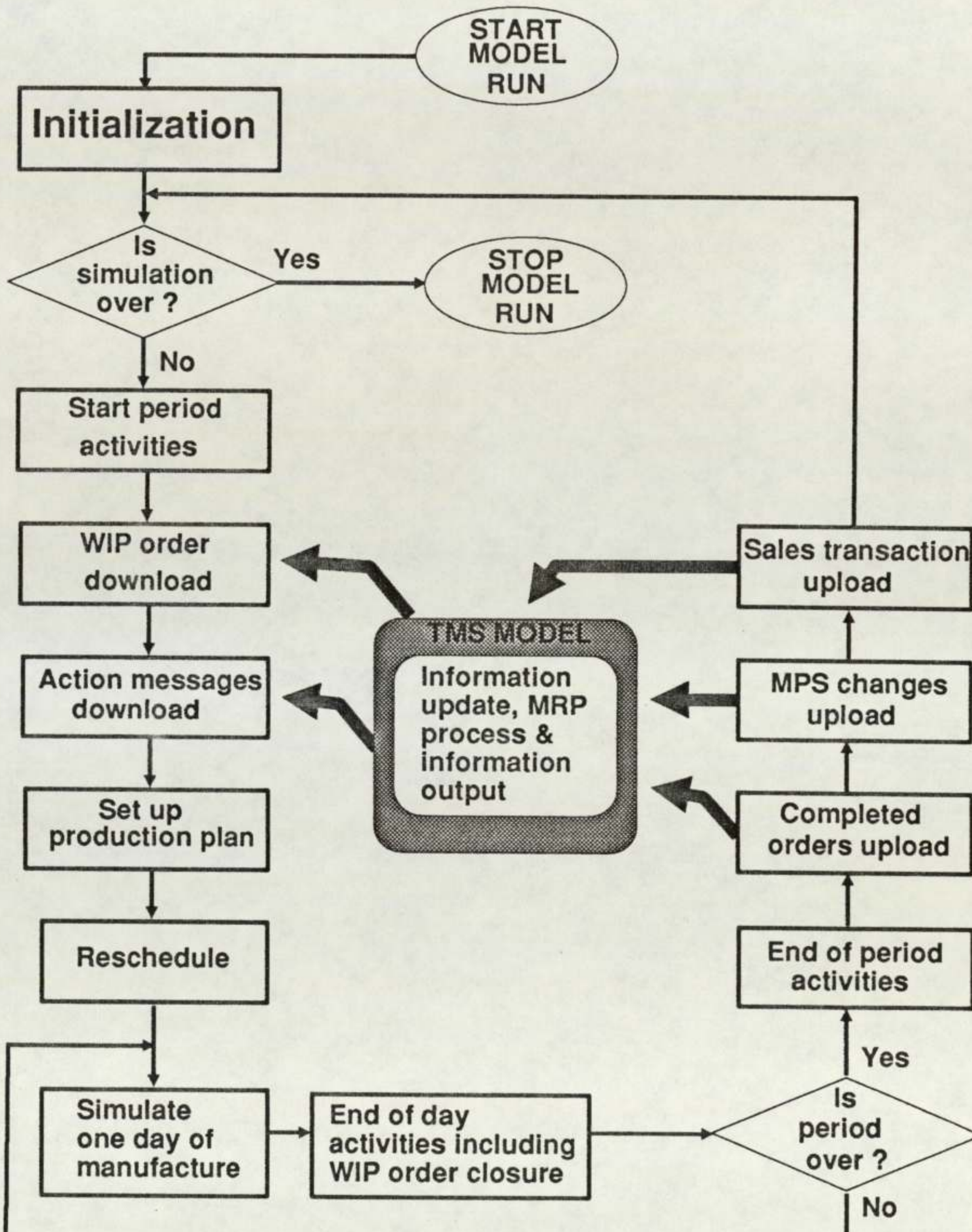


Figure 8.5 Schematic diagram of the manufacturing simulator processes including interaction with the TMS model.

on the Pinnacle and used as the basis for measuring the performance of the FCL model under various policy and parameter combinations.

8.5.2 The TMS model's view

The flow of the TMS model is controlled using 3 work flow routines. A description of each of these routines follows.

RUN/ASTONSTK

This routine must be run prior to the start of any experiment. It updates the TMS data base with the selected stock policies for each part. All stocking policies are previously entered into a file on the Pinnacle and uploaded to TMS using the model utility programs (see appendix VI). The data used to update the stock control module, for each part in the TMS model is as follows ;

i) minimum order quantity;

ii) lead time;

iii) order policy;

iv) pan size;

v) EOQ/Time bucket;

vi) minimum batch size.

RUN/ASTONMPS

This routine must be run prior to the start of any experiment. It updates the TMS data base with the initial MPS demands for each top level part. The file containing this data is originally created on the Pinnacle and uploaded to TMS using the model utility programs (see appendix VI). The production schedule is created in the TMS model by using the following data ;

i) part type;

ii) period number;

iii) quantity.

RUN/ASTONMRP

This routine forms the main section of the TMS model and as such controls all the activities necessary to generate suggested orders and action messages for the simulation model of the manufacturing facility. After updating the experimental counter the MRP process is run. The requirements for a valid MRP run are ;

i) the MRP parameter file containing all the necessary information about the current run. The parameters include ;

- MRP run start date;
- MRP run cut off date;
- MRP run mode (Regenerative or Net Change);

ii) a valid MPS;

iii) valid stock policies;

iv) a complete BOM for each part;

v) uniquely defined part numbers.

The outputs from the MRP process are used to download the appropriate information to the simulation model. The suggested orders for each part is defined by the following data ;

i) part type;

ii) order quantity;

iii) start date;

iv) due date.

Each action message uses the following parameters to download its information ;

i) part type;

ii) action message (expedite or delay);

iii) original start date;

iv) original due date;

v) new start date;

vi) new due date.

At this point, the work flow waits until the simulation model has completed a whole week of simulated manufacture, using the information downloaded from the TMS model as well as the data already resident on the Pinnacle database. The work flow automatically continues when the stock control module is updated with the completed stock information from the manufacturing model.

Any parts which may have been sold during the week or any changes which may have occurred in the MPS are then uploaded from the Pinnacle to the TMS model.

The production schedule is then rolled over by one week to the next period. If at this point the experimental counter has not exceeded the number of simulated periods required, then the MRP process is repeated until such time that it does.

The schematic flow diagram in figure 8.6 shows the total FCL model run cycle from the view point of the TMS model.

8.6 The communication interface between the TMS and shop model

The FCL model employs both hardware and software to achieve the task of interfacing the B5900 with the TDI Pinnacle. The hardware set up of the model is shown in figure 8.7. The P1000 protocol converter is used to link the two machines and allow the Pinnacle system to emulate a standard Borroughs terminal, which has two way communication between it's self and the B5900. The P1000 does this by providing a poll/select to RS232 asynchronous protocol conversion to any device with RS232 interface. A schematic diagram of the total FCL model links are shown in figure 8.8.

The communication software routines for the FCL model can be considered as existing at four different levels.

Level 1 - Routines that address the physical RS232 port at the back of the Pinnacle microcomputer.

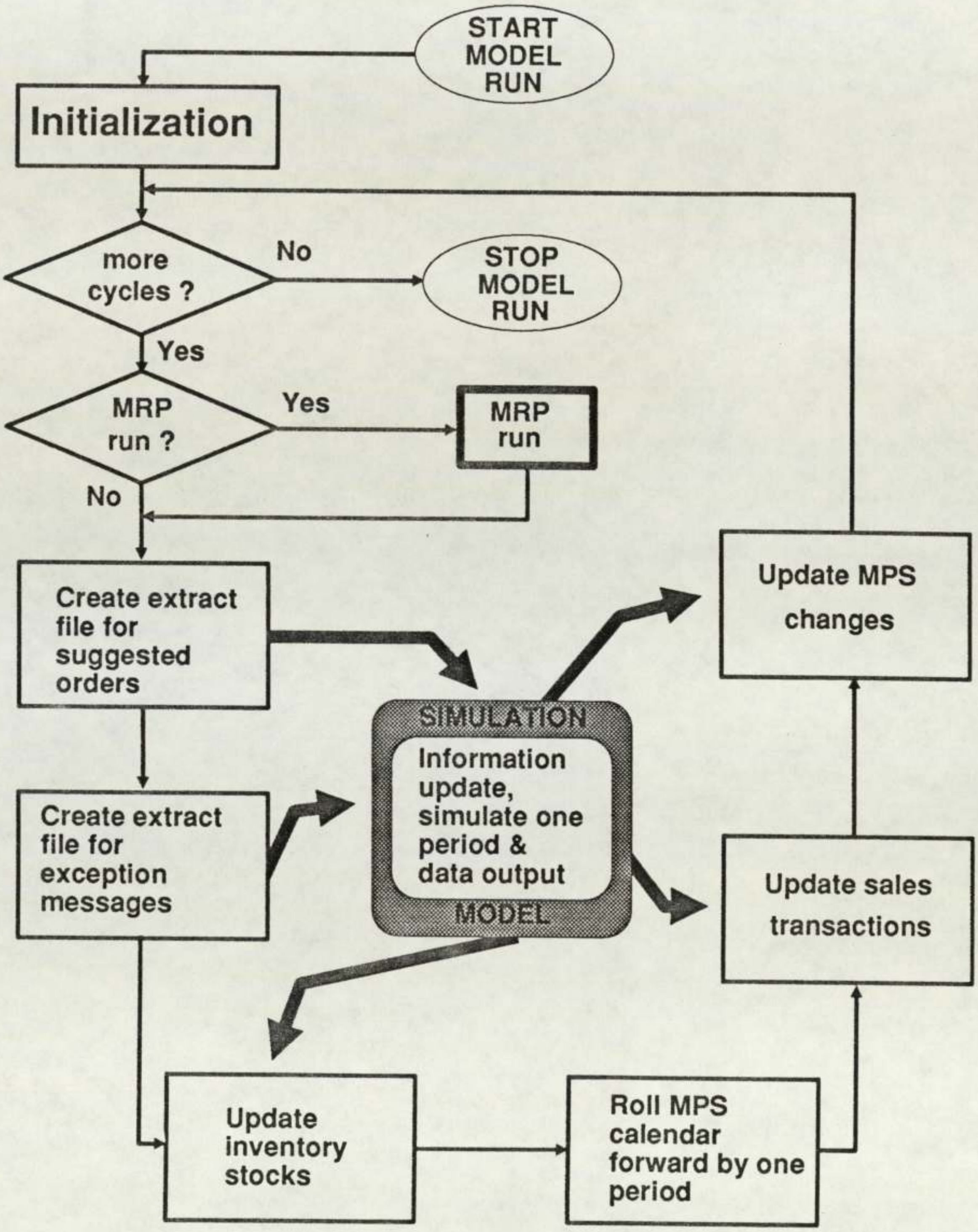


Figure 8.6 Schematic diagram of the TMS model processes including interaction with the manufacturing simulator.

Level 2 - Routines that control the sending and receiving of information to and from the RS232 port.

Level 3 - Routines that will determine how the model responds to given messages which are received from the port.

Level 4 - Routines that control the way in which the information is used to influence the FCL model.

The schematic diagram in figure 8.9 shows how the routines at each level are linked to one another programatically. At level 2 there are two separate routines ; one for transmitting information, called SENDTX and one for receiving information, called GETTXT.

SENDTX allows a character string of up to 255 characters to be sent to the communication port (port 6) of the Pinnacle microcomputer. The text to be sent is held in a character variable string. For the purpose of transmitting this variable, the characters are converted to there corresponding ASCII values. The level 1 routine, TALK01 is then called to individually write each of the ASCII values to the port until an end of text character (a null character) has been detected.

GETTXT allows a character string of up to 255 characters to be received at port 6 of the Pinnacle microcomputer. The message is received at the port in the form of ASCII values by calling the level 1 routine, TALK00. This routine also checks for spurious characters which do not make up any part of the message. When an end of text character is received control is returned to the level 2 routine, GETTXT, for character conversion. The received text is held in a character variable.

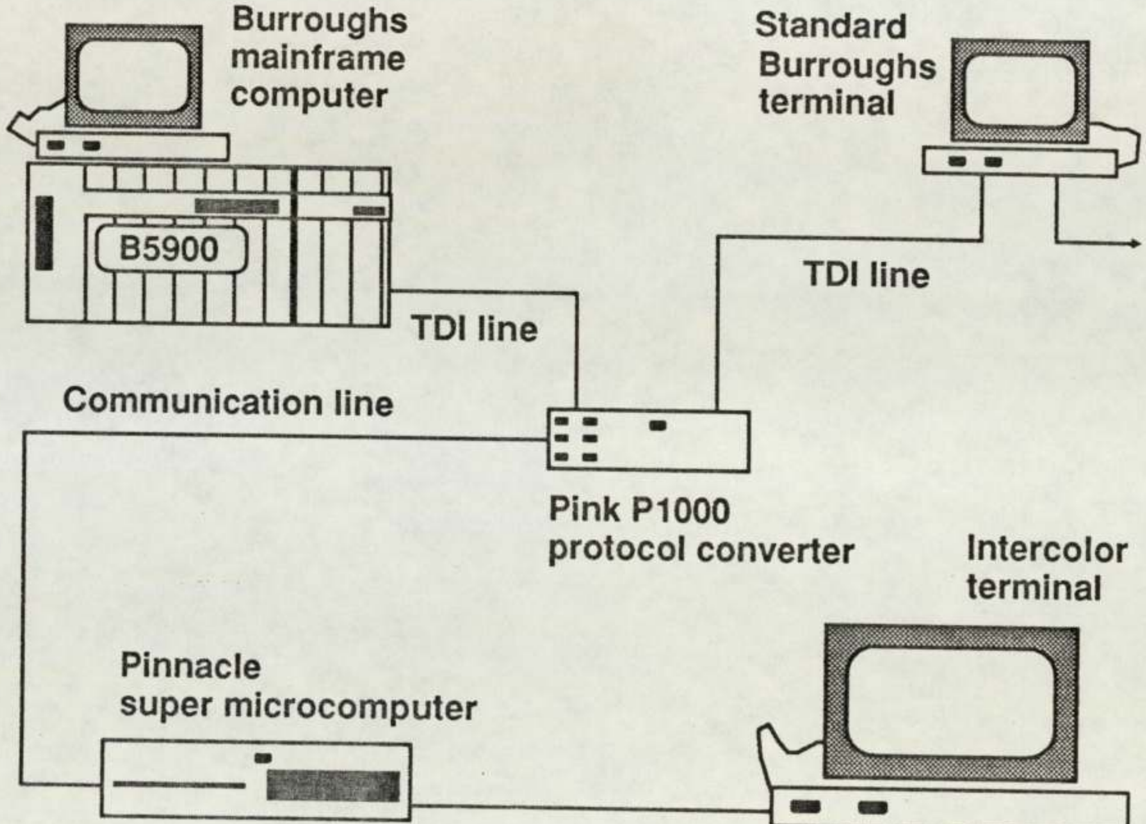


Figure 8.7 Hardware configuration for FCL model

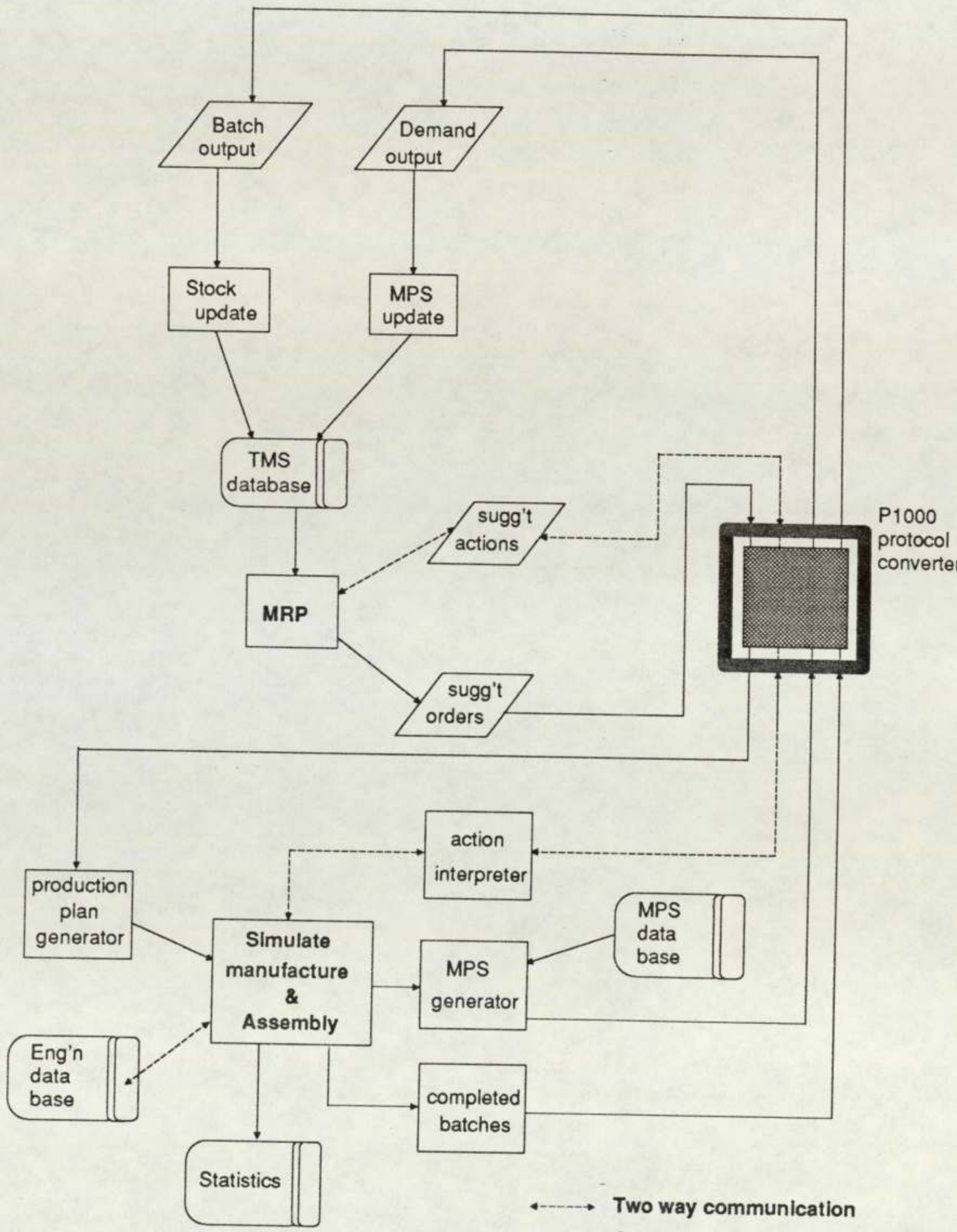


Figure 8.8 A schematic diagram of the 'total' model

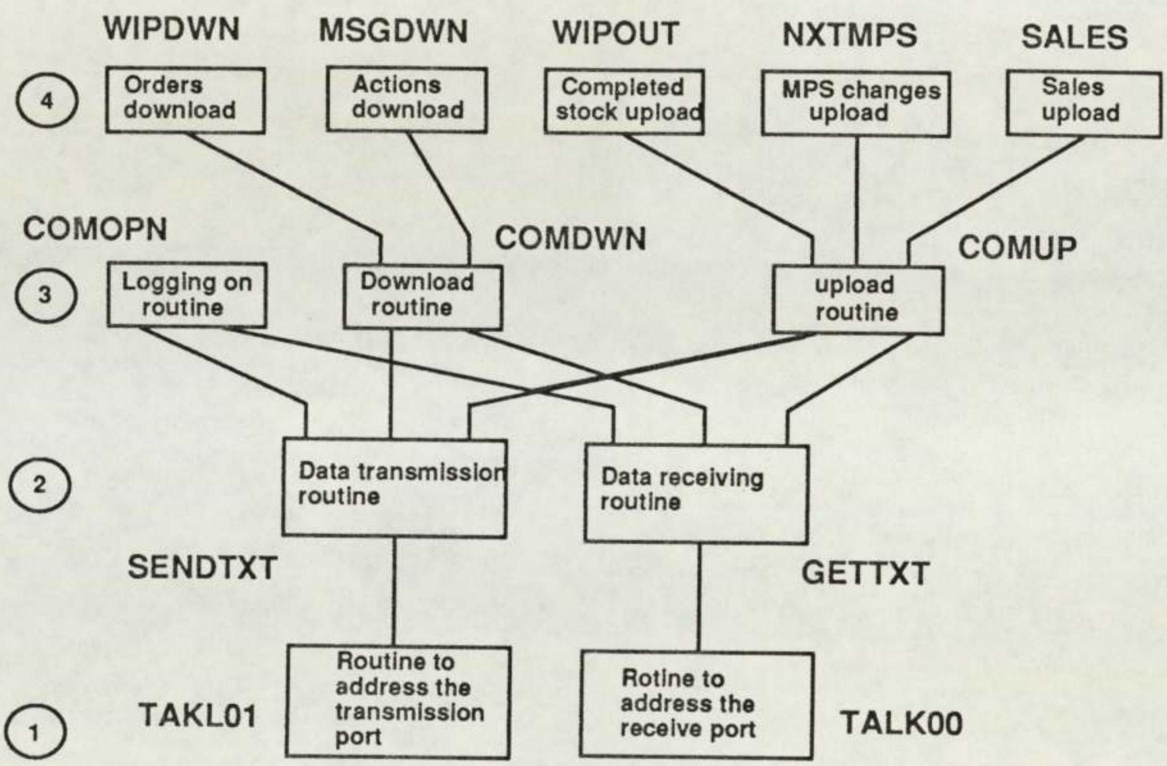


Figure 8.9 Hierarchy of communication software levels for for FCL model

All routines at level 3 have access to level 1 and 2 routines. COMOPN is used to initiate a session on the B5900. The routine determines what part of the logging in procedure is being sent or received and responds accordingly. COMOPN has no level 4 link since the logging in procedure is standard and is instead called directly with fixed parameters.

Two routines exist at level 3 for the purpose of actual model data transmission. The first is COMDWN which controls the activities associated with all download programs (ie data flow from B5900 to Pinnacle). These include ;

i) calling the level 2 routine, SENDTX with the appropriate command to allow the B5900 to download information to the Pinnacle;

ii) using outgoing text to determine how to read incoming text;

iii) calling the level 2 routine, GETTXT to return the information which is being downloaded.

The routine, COMUP controls the activities associated with all upload programs (ie data flow from Pinnacle to B5900). These include ;

i) calling the level 2 routine, SENDTX with the appropriate command or data to be uploaded from the Pinnacle to the B5900;

ii) using outgoing text to determine how to read incoming text;

iii) calling the level 2 routine, GETTXT to return the 'ready for more data' prompt from the B5900.

At level 4 there are two download routines which use COMDWN and three upload routines which use COMUP. The routine WIPDWN is responsible for receiving the suggested orders from TMS and writing the relevant information to a file on the Pinnacle database. MSGDWN receives the action messages from TMS and again writes the relevant information to the appropriate file on the Pinnacle. Completed stock data is sent to the B5900 using the routine WIPOUT. Sales and MPS change data is uploaded by the level 4 routines SALES and NXTMPS respectively.

For a more detailed description of these routines the reader is referred to the program documentation in appendices V and VI.

9 Validation of The FCL Model

9.1 General Discussion

The verification and validation of a simulation model is perhaps the most important and certainly the most difficult task to perform during simulation model development. Neither are isolated processes and as such should not be considered only at the completion of the program writing stage. Instead the process of validating a model is ongoing.

Fisher and Kiviat(1967) have conveniently distinguished between verification and validation as follows ;

i) **Verification** - the process of ensuring that the model behaves in the way the modeller intends.

ii) **Validation** - A comparison of the behaviour of the verified model with that of the real system.

Verification of each of the major FCL model elements (i.e. the market place, the production control system, the manufacturing system and the information transfer technique) have been greatly eased by virtue of the design methods used. The first two elements use the real system to analyse how they behave under controlled conditions. This means that the verification phase for these two elements can be virtually eliminated. The third model element has been developed using the OPTIK simulation package which incorporates a comprehensive trace facility for model code diagnostics as well as animated graphics to show the modeller how the different processes and model entities interact.

The final model element emulates the role of the production scheduler and the production controller. The results of each decision making process are well documented as job summaries and can be analysed at any time during the experimental run.

The second aspect, validation, does however, raise a number of important philosophical questions regarding the value of true correspondence between the model and the real life system. In the purest sense, Naylor and Finger(1967) suggest that validation is the test of truth for a simulation model. The word 'true' is infinitely subjective and consequently Naylor and Finger continue by saying ;

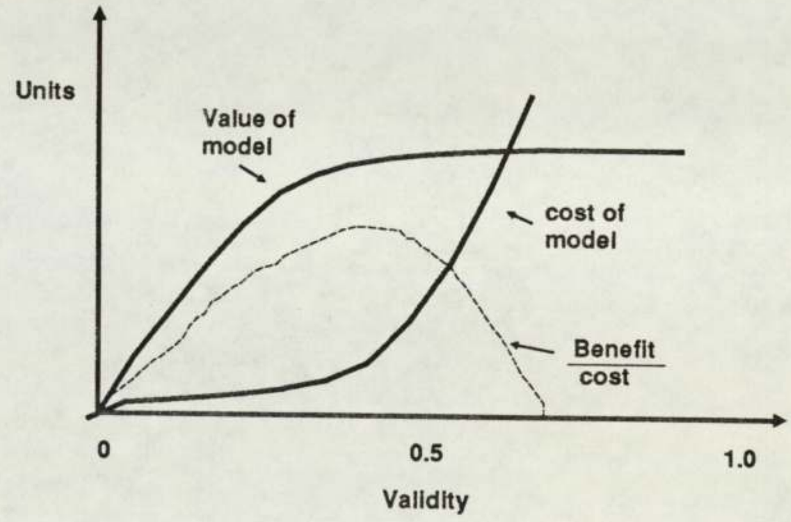


Figure 9.1 Anshoff and Hayes validity curves

"...to prove that a model is 'true' implies (1) that we have established a set of criteria for differentiating between those models which are 'true' and those which are 'not true,' and (2) that we have the ability to apply these criteria to any given model. In view of the difficulty which arises in attempting to agree upon a set of criteria for establishing when a model is validated, Popper(1959) has suggested that we concentrate on the degree of conformation of a model rather than whether or not the model has been validated."

This view confirms the widely accepted means of evaluating a model as a representation of reality. Anshoff and Hayes(1972) further suggest that as the degree of conformation of the model increases, its associated development cost also increase. At the same time its value to the model user will also increase, but most likely at a decreasing rate. Figure 9.1 shows the way in which Anshoff and Hayes depicted this relationship. It should be noted that, if '0' represents absolute invalidity and '1' represents absolute validity of the model, then the 'benefit to cost' ratio normally peaks far short of '1'. The process of actually validating a simulation model has been the subject of much debate among simulation program analysts and practitioners. The arguments differ between the need for the validation to be objective and the need for the analyst to make constructive and intelligent use of his subjective beliefs. There are three major methodological positions concerning this problem which are aptly referred to as the rationalist's view, the empirisist's view, and the pragmatist's view.

The rationalist propounds that a model is made up of logical deductions from a series of basic truths. He further adds that these truths are unquestionable and as such are not open to objective analysis because they are apparently obvious. In contrast, the empiricist believes that all model elements must be verified before their inclusion is accepted. In relating this view to management science, Reichenbch(1951) argues that ;

> "Verifiability is a necessary constituent of the theory of meaning. A sentence the truth of which cannot be determined from possible observations is meaningless."

The third view on how model validation should be conducted is one which is not concerned with the internal structure of the model. The pragmatist asserts that as long as the model fulfils the purpose for which it was built then it is a valid model. His conclusions are made on output-input transformations only.

The controversy is usually over matters of emphasis and not necessarily one view against another. In general the analyst needs to be objective, but the way he makes progress is by following up subjective insights.

In an attempt to globally define the validation process Hermman(1967) proposed the following five validity criteria or approaches ;

i) **Internal validity** - using model replication, and holding model inputs constant, one determines whether the variance of the response is too large.

ii) **Face validity** - using subjective opinions regarding the surface or initial impression of the model's realism.

iii) **Variable parameter validity** - primarily "sensitivity testing" in order to ascertain whether the effects of changes in the model's variables are compatible with comparable alterations in the modelled system.

iv) **Event validity** - comparisons of "predictions" (responses) of the model with past (recorded) history of the actual system.

v) **Hypothesis validity** - examination of connections between system elements, so as to determine whether the model reproduces these relationships.

The aim of any simulation model should be to create the same problems and behaviour characteristics as the process or system being modelled. It is therefore essential that a model is not only created for a specific purpose but that its validity be evaluated only in terms of that purpose. The purpose of the FCL model is to provide management support in the selection and design of control policies and parameters with the view to increasing the efficiency of PCB manufacture.

During the validation of the FCL model, Herman's criteria was used as far as possible for the purpose intended by the model. The use of this technique was however, limited in this work because of the lack of relevant historical data at FCL. Despite this, validation of the model was proved to be acceptable based more on subjective rather than objective analysis.

As discussed earlier in chapter 7 the total FCL model consists of four major sub-models, these respectively represent the market place, the production control system, the manufacturing system and the information transfer system. The validation process for the FCL model has been considered at three levels. First the evaluation of each of the sub-models along with their associated elements, secondly the way in which these sub-models interact together and finally the evaluation of the whole FCL model. The following sections discuss the application of Herman's criteria to the FCL model.

9.2 Validity of the market place model

In creating the market place model the marketing department, planning department and shop floor personnel were consulted. They each played an essential role in verifying that the collected MPS data from the past 12 months of their business operations was relevant to our study. In the light of these discussions and with due regard for the advice given it was agreed that a 3 month horizon of all of FCL's market place activities would be suitable for the purpose of the market place model. This time period was chosen to represent the market demands between the months of September and November 1986, since at that time the data was considered to be more reliable and the data recording techniques were more consistent. A number of verification checks were made on the data by the research team to eliminate both typographical errors and any misconceptions between our own ideas and the original intentions of the production schedulers when the data was recorded.

A requirement of the experimental phase was to evaluate the effect of policy combinations against different market places. The consistency of FCL's current market place activities presented a problem in this respect. The appropriate FCL personnel were again asked to verify a number of artificially induced amendments to the original market model in order to form the basis of alternative and feasible market places. The first amendment involved splitting up the original demands in order to spread out the required quantities over longer time periods. This, it was argued, would produce a smoother and consequently more even production schedule. In reality, it was argued, this smoother demand profile would be experienced by applying an appropriate order intake policy at the MPS level. Another change to the demand profile was based on the well established policy (not currently at FCL) of only accepting genuine customer demands and involved the removal of all forecast demands in the original production schedule. Other verification discussions included artificially increasing each customer demand as it entered the shop model prior to launch and grouping the customer demands at MPS level based on demand frequency and load. The subjective opinion of those involved in the discussions was that these alternative schedules were valid representations of what would be expected in reality under the appropriate conditions. A detailed description of these market place models is given in Clarke.(1988).

9.3 The validity of the production control system model

The following discussion describes the way in which each of the four major TMS modules(EDC, MRP, STK and WIP) were validated for the purpose of the FCL model.

There was little difficulty in performing internal validation on the TMS database model since it was largely a replication of the live A9 database. The assumption that all purchase components could be conveniently omitted from the study was based on FCL's challenge to achieve total fluidity on the shop floor for all purchased components. In addition to this, the influences of the effects of purchased components lie outside of the manufacturing facility(i.e. the main influences come from vendors and suppliers) and would consequently warrant further study. In the light of this it was argued that the database could be adequately represented by modelling to PCB level only both in part definition and also in product structure. The MRP module, whilst being an exact replica of the real system, still had to be tested against the newly created database for validational purposes. The majority of these tests were conducted using the variable parameter validity of Herman's criteria. The predicative nature of the MRP process made the comparison between real and model responses very straight forward. The tests included ;

i) Making sure that the main inputs to the MRP process, i.e. the MPS, the BOM and the inventory stock were being used correctly.

ii) Analysing the customer demands produced, given the inputs to MRP.

iii) Analysing the netting off process of MRP by checking the resulting suggested orders report as well as the distribution of part allocations.

iv) Analysing how the MRP process responds to changes in the MPS over a number of periods, where MRP is run every period. Initially it was suspected that for every period where a change occurred, MRP would need to be run several times, corresponding to the number of levels in the offending BOM's. This would allow the effect of the change at the top level to be reflected to the lower level components by confirming the suggested orders at each stage. Tests revealed that only one MRP run was in fact necessary for each period, regardless of any changes which may have occurred. To achieve this the assumption was made that all suggested orders for the model be confirmed. This was acceptable since all of these orders would eventually be launched in the shop model unless they were adversely affected by exception messages.

v) A check for logical generation of exception messages based on changes in the MPS or overdue demands.

The MRP module includes a mechanism for representing each customer requirement in terms of quantity and time over a possible horizon of 60 weeks. This, it was argued could be used in favour of the actual MPS module(which is not included in the TMS model) since it fulfilled the necessary requirements of representing customer demands without the added overhead of the separate TMS module. This representative MPS was verified as being compliant with that of the relevant functions of the actual MPS module.

The inventory stock module for the TMS model was used to represent all the stock control policies for each part and as such, due to the purpose of the FCL model, this information was entered at the time of experimentation. It was however argued that every modelled part would assume an initial stock level of zero. The basis for this argument was reconciled by the assumption that after the transient phase of the experimental run, stocks would adjust to their natural levels. This hypothesis was tested and found to produce results in favour of the stock level argument.

The main purpose of the WIP module in the TMS model is to generate the appropriate allocations for the component stocks, based on end item requirements. Testing for this was conducted in conjunction with the MRP module and the resulting allocations were found to be of unquestionable validity.

9.4 Validity of the Manufacturing System model

The manufacturing model has been designed to simulate both the manufacture of PCB's and the assembly of sub-systems and finished products. The primary task however, was to evaluate the effect of control policy decisions on the effect of the former. Initial discussions were therefore directed at creating a model that represented PCB manufacture in great detail whilst at the same time simulating the assembly area using a less comprehensive approach. It was later agreed that there was a necessity for the assembly area representation to be modified in order to produce a valid FCL model. This was justified, since one of the performance criteria was due date accuracy, which relied on a valid appraisal of the flow of work from launch to finished goods store. It was further noted that the assembly area provided a facility for consuming manufactured PCB's in the completed batch store and was therefore instrumental in maintaining component stocks at realistic levels.

Herman's Face validity criteria was used to verify the ACD as detailed in chapter 8. This was accomplished by exhaustive discussions with the relevant FCL personnel, since this was to form the basis on which the simulation model code would be written. The discussions highlighted the function of each machine and process, the role and availability of operators and setters, the function of the stores area and the basic flow of jobs on the shop floor. The code for the modern manufacturing model was written using the verified ACD logic.

The next task was to verify that the program code produced the manufacturing model as intended by the author. A number of assumptions were made at this stage which in no way violate the validity of the FCL model or are in conflict with the total simulation model approach as discussed in chapter 5.

The process times were modelled deterministically. The justification for this is discussed in chapter 8. Apart from a much reduced verification and validation procedure, this assumption eliminated the need for experimental replication and thus afforded a more extensive analysis of policy combinations.

Breakdown of machines seldom occur in the actual manufacturing facility, largely due to reliable equipment and the policy of preventative maintenance. For this reason machine breakdowns are not accounted for in the the current model.

The facility to include this feature does however, exist and may be used by future analysts for subsequent studies.

The batch launching mechanism in the manufacturing model releases jobs exactly as specified by the MRP model. The delays that would normally be encountered in reality were deemed to be of little consequence for the model purpose. This was justified on FCL's continued commitment to minimise the problems associated with batch release. A further justification for this assumption was for the shop model to be wholly evaluated on its ability to respond to the MRP outputs under favourable conditions. The assumption is also made that all jobs are launched at the start of any given shift. In reality, jobs are released perhaps 3 or 4 times through out the day. The timing error as a consequence of this assumption was acceptably small when compared to lead time allowances and was therefore accepted as negligible for the purpose of the study.

It was argued that there was no direct relationship which existed between transport times and workcentre visitations in the real manufacturing system. Instead, these times depend on the availability of the appropriate job folder and a requirement from the next workcentre for that particular job as well as the general dexterity of the operator moving the job. In the light of this situation it was decided that the model should be designed so that the transportation time could be used as an experimental parameter rather than a model variable. In this way it would be possible to evaluate the effect of transport times on the performance of the model. Each operator and setter classifications were grouped to form 'resource worlds'. Lengthy discussions with the actual operators and setters on the shop floor along with the relevant production managers were held to ascertain the validity of this assumption. It was subsequently agreed that the labour world representation was valid. Given the above assumptions, a number of internal logic tests were conducted to ascertain that the model reflected the function of the real system. Test batches were defined so that they would sequentially flow between each workcentre to show correct queuing characteristics and machine process logic. In addition, these batches were used to test the resource worlds for operators, setters and machines. Numerous hand simulations were also run to test a limited number of parts for correct operation route, product structure, operation time and set time allocations from the Pinnacle database.

The graphical features of the OPTIK language proved to be very useful in the assessment of Face validity of the shop floor model. Throughout the model development, close contact was maintained with the production personnel who, on the basis of their experience of the real system, were able to check and comment on the reality of the model's behaviour. Agreement was reached that in the subjective opinion of those concerned, the model did show responses typical of the real system.

The remaining criteria for the manufacturing system element is discussed in relation to the FCL model in section 9.6.

9.5 Validity of the Information transfer system

The transfer of information was automated for reasons of expediency. At each stage where information was required; the model took on the role of either the production controller, the production scheduler or the production manager. The relevant people in each of these departments were asked to subjectively verify that this flow of information was typical of the decision making process in real life. They jointly confirmed this and thus completed the face validity test for the information transfer system.

9.6 Validity of the FCL model

For the validation of the total FCL model each criteria was limited to one set of control policies, namely the policies that had been in use during the period of time under investigation, and the corresponding market place. The policies at FCL were represented in the model at 3 specific levels as follows ;

i) **The MPS**

This was the historical data which represented the market place over the 3 month time period from September to November 1986.

ii) **TMS**

Stock policies for each part were used to control the generation of suggested orders. Blanket policies included ; lead time allowances of 15 days for PCB's and 20 days for assemblies, a pan and minimum batch size of 20 for all parts and a 2 week time bucket order policy for PCB's with assemblies being made to order.

iii) **Modern Manufacture**

The shop model was configured to represent the actual function of modern manufacture over the stated period. At that time there were no formal priority rules in service and transport times on average were very long. Further investigation suggested that although there were some functional changes in the real system over the period which was to be analysed, they were not considered major enough to effect the results of the experiments. The response of the model to changes in parameter values was in general consistent with the results expected from experience in the real system. The equivalent changes could not however, be fully tested for economical as well as practical reasons.

The initial tests involved variation in labour and machine availability to establish the effect of resource constraints. The aim of this was to re-enact past resource limitations, as described by FCL personnel in order to investigate the occurrences of predicted bottlenecks. The model was also run against a number of artificial shop loads(in terms of batch quantity) and the corresponding performance monitors analysed. These experiments were conducted at 80%, 85%, 90%, 95%, 100%, 105%, 110%, 115%, 120%, 125%, 130% and 135% of the original shop load. It was interesting to note that as the load increased, so to did the machine utilisation but at a continually decreasing rate. It was also found that for similar increases in load the basic machine utilisation profiles exhibited the same characteristics. An example of this characteristic profile is shown in figure 9.2. Variation in job transportation times were also shown to affect the FCL model in a similar way to that expected in the real system.

The event validity tests were conducted in order to establish variations between model predictions and past historical data. After running the FCL model with the original policies as outlined above, a strong relationship was found between the performance measures of both the model and the real system. Clarke(1988) summarises the resulting comparisons.

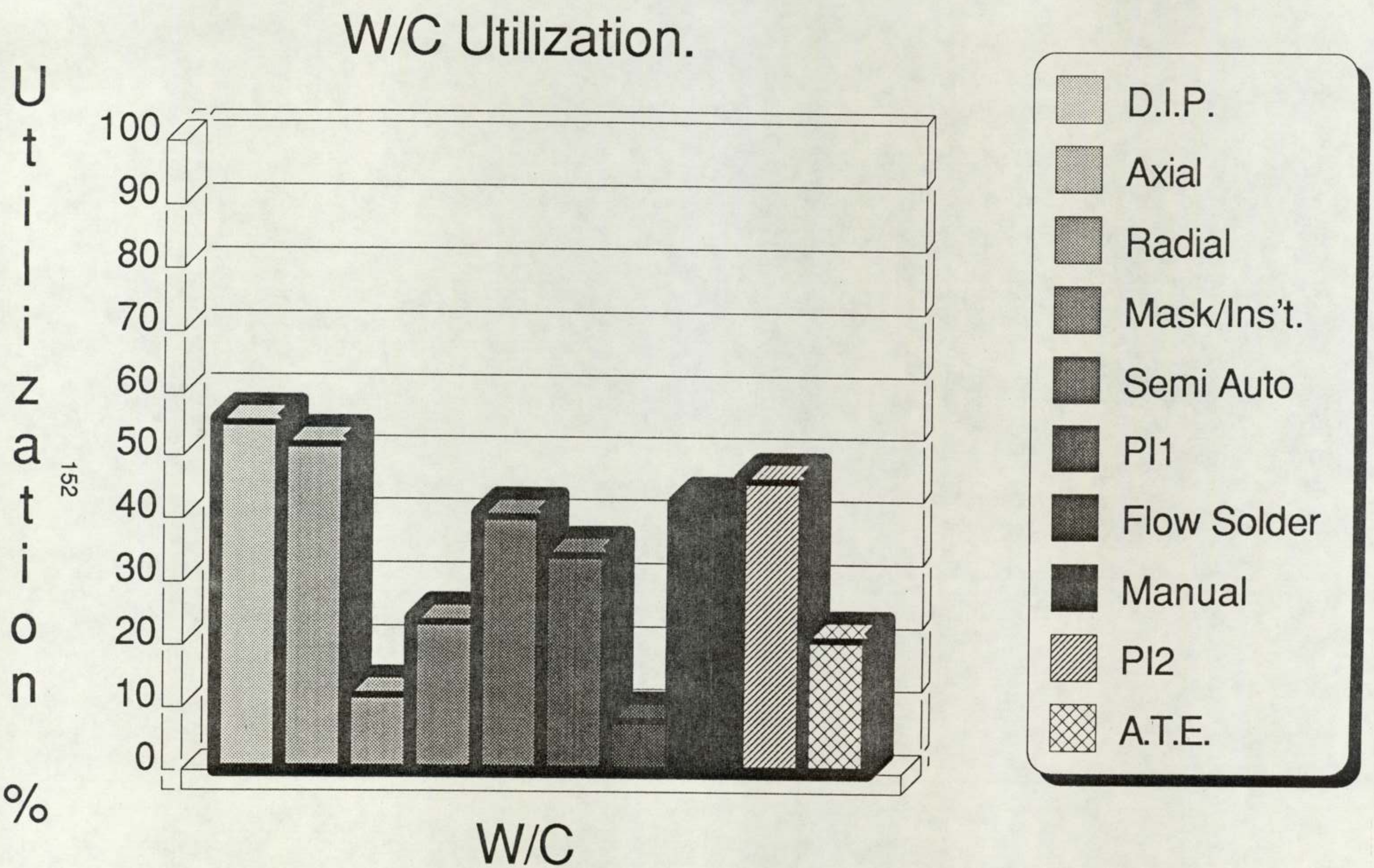


Figure 9.2 Typical load profiles

The hypothesis validity tests are primarily concerned with the elemental relationships of the model. This is discussed in chapter 8 and as such has been accepted as valid.

9.7 The validity of the modelling approach

It has been shown that the modelling approach as discussed in chapter 5 can be applied to the development of a policy evaluating facility for an industrial case study. The validity of the modelling approach therefore, has been judged on its suitability to this particular purpose. An essential test of approach validity is whether or not better systems result from investigations based on model experimentation. This is discussed with particular reference to traditional results in chapter 10. The following points concern the subjective validity of the modelling approach adopted. This is pertinent since improved managerial effectiveness invariably rests on the intuitive support they receive.

The concept of a real production control system driving a manufacturing simulator was highly instrumental in reducing the skepticism felt among the FCL management as to the credibility of the model. The inclusion of the MPS as a real input to the MRP process further served to build their confidence in its proposed capabilities.

The modelling approach was also judged by its suitability for addressing the specific questions and problems of FCL as discussed in chapter 6. It was recognised that an elaborate and accurate model could do little if it related to behavioural characteristics which were of no consequence to the success of FCL's business objectives. The management agreed that the approach incorporated sufficient information regarding the probable factors relating to the integration problems of FCL. This relates to the proper selection of model boundaries, which after much discussion were confined to the market place, production control and production supply. As with the selection of objectives, the choice of system boundary was not guided by any definitive theory that could be objectively assessed. Their application was instead based on the judgement and successful experience of those involved in the discussions.

The interactive nature of the system variables is a direct consequence of the modelling approach adopted. For example, the decision to model real products and their respective sub-components flowing through individually resourced processes was a definite requirement since the approach dictated a comprehensive performance monitor for FCL. The performance criteria requirements prompted lengthy discussions with the FCL management. The conclusion of these discussions lead to the current variable interactions in the FCL model being accepted as a valid representation of those in the real system.

10 Experimental Analysis using the FCL model

10.1 Experimental Methodology

The total modelling approach has been discussed and successfully applied to an industrial case study using FCL (see chapters 5, 8 and 9). The resulting model has been designed to specifically evaluate the various control policies achievable at different levels in the real system at FCL (see chapter 6). It is emphasised that the model in no way serves the purpose of an interactive predictive tool, but instead offers guide lines towards the managerial task of making decisions by investigating control policy options. Furthermore, the FCL model in its current form is suitable for batch run experiments only. This experimental method requires that each policy and appropriate experimental parameter be selected and entered into the model prior to any experimental run. The model provides two distinctly separate operational modes during experimental analysis (see user guide) which allow a wide range of real system attributes to be analysed and affords a more efficient usage of the of the limited computer resource. In addition, the interactive approach to experimental analysis provides a unique opportunity to ascertain the effects or otherwise of a rolling schedule on the performance of the manufacturing facility.

The author has produced a detailed user guide for the model which includes a step by step description of experimental procedure using the FCL model and its associated utility programs. In addition, a comprehensive account of the experimental methodology is given in Clarke(1988) along with a discussion of the experimental results and associated implications. The following discussion is confined to a sub set of those experiments which yielded results that were not only interesting but also different to those produced by traditional simulation research approaches.

10.2 Experimental design

Each experiment with the FCL model was conducted with variation at three separate levels, namely MPS, MRP and shop. These levels were further subdivided into several factors which related to appropriate control policies. Figure 10.1 illustrates the setting of the control policies at each of the levels.

At the MPS level, three feasible manifestations of the market place were used to provide independent inputs to the production control system model. The original MPS data at FCL formed the primary and base experimental conditions since it represented the actual market place as was seen throughout the data collection period. The second experimental factor at this level was achieved by arranging the original customer demands into groups based on the standard hour content of each MPS product as specified in Clarke(1988). The final MPS factor was again based on the grouped MPS, but with the added assumption that all MPS forecasting be removed. Clarke includes the three MPS configurations in his thesis.

The production control experimental conditions were initially categorised into four separate policy combinations based on order, stock and lead time. A linear lead time policy with a variable queue time element was introduced at this level and used to ascertain its merit against the currently used generic lead time policy. The production control level was finally reduced to two experimental conditions based on two linear lead time policies. A detailed discussion of the analysis used to arrive at these factors is given in Clarke(1988).

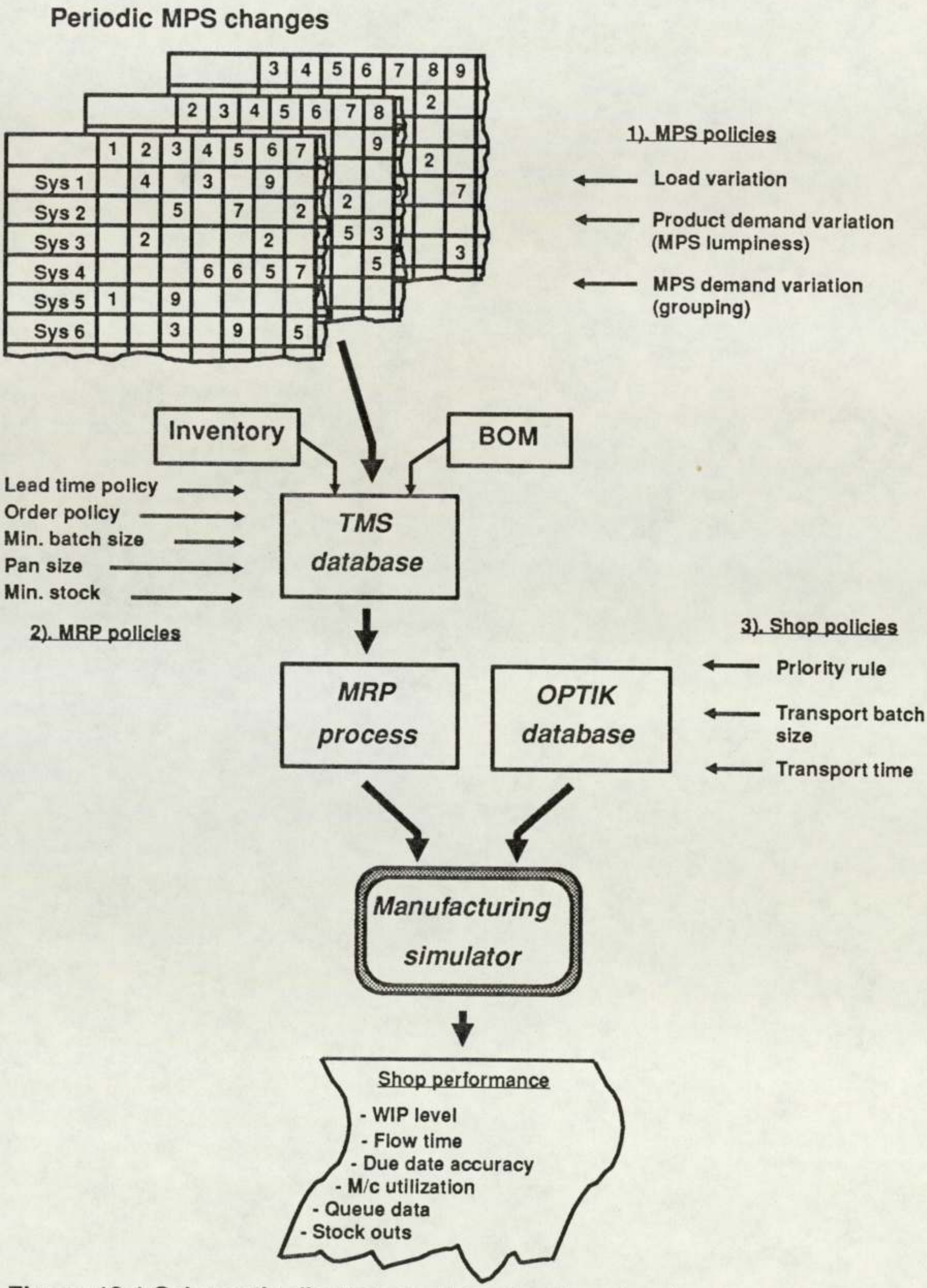


Figure 10.1 Schematic diagram to show the three levels of experimental policy design.

At the shop level, the factors were confined to three policies. The first shop configuration represented the original layout in the manufacturing facility, including a verified replication of the currently active labour and machine resource constraints. In addition to this, the existing 'first in first out'(FIFO) priority rule prevailed along with the jobs being processed in quantities as suggested by TMS. The second experimental factor introduced a batch splitting algorithm which allowed jobs to be transported and processed as conveniently small production quantities. The final shop experimental factor was based on the previous shop configuration, but with the addition of a priority rule for identifying and efficiently utilising bottleneck work centres. This algorithm is described in chapter 8 and its associated coding is documented in Appendix V.

A summary of these levels and the factors used is given below ;

MPS policy combinations

- Original MPS at FCL

- Grouped MPS

- Grouped MPS with no forecast demands

Production control policy combinations

- Experimentally determined order and stock policies including a linear lead time policy with a queue element factor of 0.5 days

- Experimentally determined order and stock policies including a linear lead time policy with a queue element factor of 1.0 days

Shop policy combinations

- Original shop

- Transport batch

- Transport batch and a bottleneck sequencer

The above represents 18 separate experiments which were each compared against a number of other base experiments run using the original policies adopted at FCL. Each experiment was run under rolling schedule conditions (i.e. using the interactive experimental mode) and as such took full advantage of the model's ability to evaluate each of the policy combinations.

The statistics generated from each experiment guided the assessment of the appropriate policy combinations both within and between levels. Performance data which was found to be extremely useful included average WIP, average flow time, average lateness and standard deviation of lateness. The statistical performance information was also used for each policy combination with results obtained from respective experiments run in the non-interactive mode, i.e. with no uncertainty in the MPS (this approach is associated with the traditional simulation studies of the production systems found in the literature).

In addition to the above experiments, an assessment of production plan lumpiness was made by recording the standard hour content of all jobs due for release on a period by period basis. This data was analysed since it effectively represented the interface between what the production control system suggests should be done and what the manufacturing system needs to achieve.

10.3 Experimental findings

Clarke(1988) reports on a number of important findings which supports the need for a total modelling approach to analysing a production business in the manufacturing environment. Perhaps among the most interesting and certainly the most significant are those associated with the introduction of uncertainty in the MPS model. A simple comparison of the results obtained by initially using the non-interactive mode of operation with those compatible results using the interactive approach reveals the following observations ;

i) An increase of between 10 - 15% in shop load measured in standard hours.

ii) A reduction in the average order quantity of between 5 - 8%.

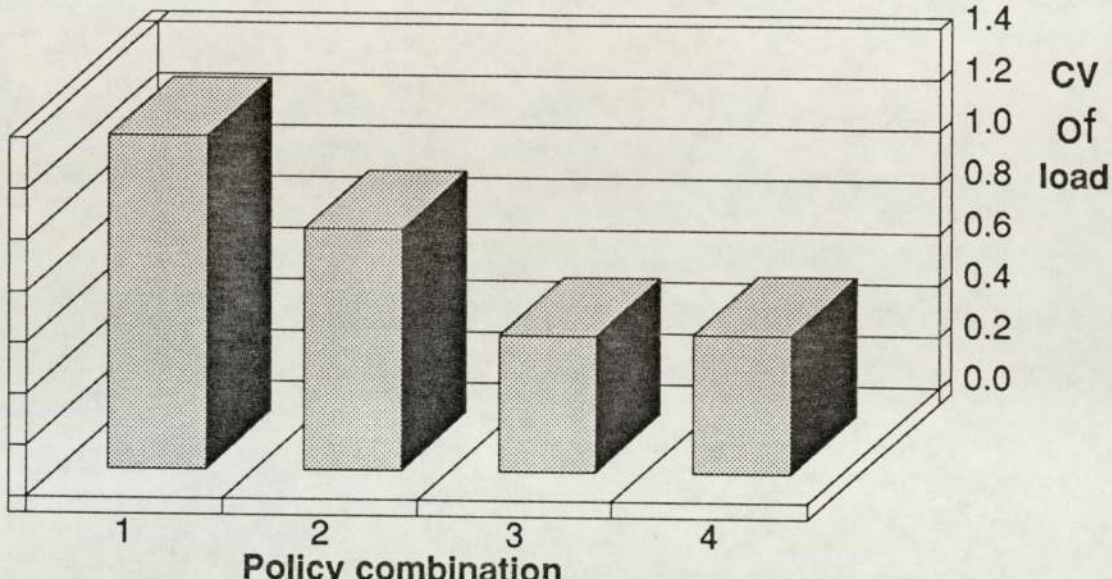
iii) A 50% increase in production plan lumpiness measured in standard hours.

Further comparisons revealed the disparity between the two modes of operation with changes in policy combination. It was observed that whilst a move from the original MPS policies to a grouped MPS policy (with a linear lead time policy) tended towards the generation of a lumpier production plan under non-rolling schedule conditions, the same policy decisions produced a much smoother plan when uncertainty was present.

Table 10.1 shows the differences between such plans and appropriately illustrates the points made earlier with regard production plan aspherity. Figure 10.2 further reinforces the argument by showing the standard deviation of

	NON - ROLLING SCHEDULE		ROLLING SCHEDULE	
Period	Original MPS & MRP parameters	Grouped MPS, original MRP parameters	Original MPS new MRP parameters	Grouped MPS new MRP parameters
7	1988.3	1741.1	2219.9	696.9
8	1736.9	904.2	1276.7	1275.9
9	738.9	1073.6	539.4	2185.4
10	984.9	1404.4	504.3	792.4
11	1152.6	890.2	1404.1	864.4
12	2620.2	2032.0	4316.8	519.7
13	710.1	546.9	640.8	1957.7
14	1190.5	1805.5	602.4	2827.1
15	1658.5	847.7	1050.8	1297.4
16	1128.0	1256.5	2630.6	1051.2
17	285.6	1399.9	1020.4	1762.5
18	584.0	613.5	775.4	1634.3
19	655.3	1138.1	883.5	1018.1
20	1288.6	378.6	4041.2	1434.9
21	1154.8	853.4	1659.4	2398.6
22	825.5	1617.7	310.4	2287.9
23	702.8	1276.5	786.3	992.5
24	1396.4	508.9	1109.0	890.5
25	1398.5	1329.3	1570.1	3544.1
26	940.6	1125.8	3076.5	1906.0
27	634.1	659.9	736.8	1385.2
28	1496.9	829.1	962.6	1888.6
29	805.4	368.9	824.3	662.7
30	773.1	1745.8	3270.9	1503.9
31	254.7	529.0	614.5	1697.3
32	448.7	393.2	82.0	2122.6
33	477.8	1144.5	72.0	1141.1
34	1332.3	167.9	5261.8	886.1
35	591.1	1164.7	265.2	982.9
36	971.2	1993.9	429.9	1593.0
37	1504.0	776.4	366.9	331.2
38	1561.9	1400.8	382.8	76.1
39	504.8	540.3	1006.0	535.3
Total	34496.1	34455.3	44693.7	46143.5
Mean	1045.3	1044.1	1354.4	1398.3
Std.	515.9	495.0	1277.2	739.0
c.v.	0.49	0.47	0.94	0.53

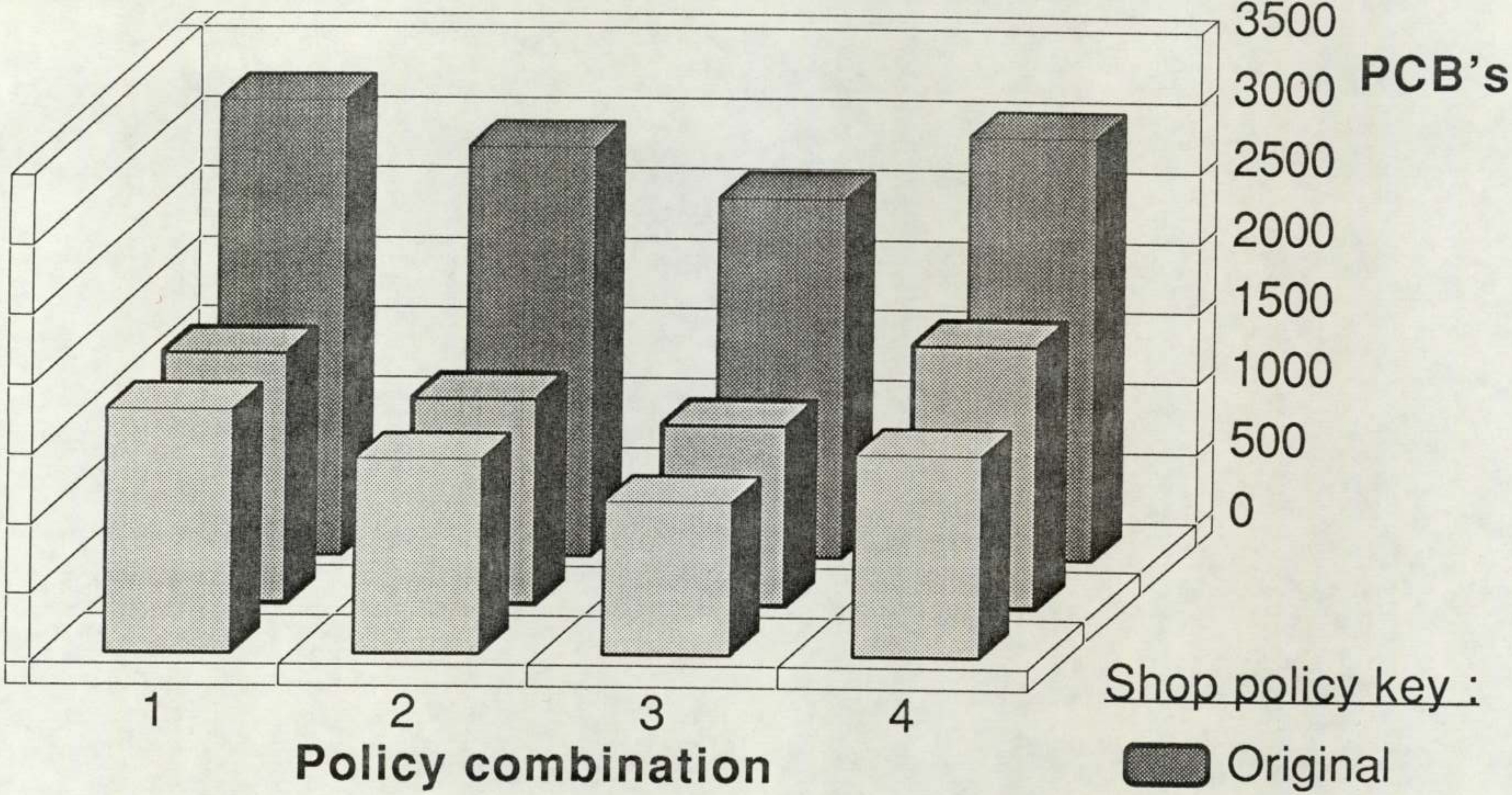
Table 10.1 Production plan comparisons for interactive and non interactive experiments (standard hours per week).



Key :

Policy 1-Original MPS & MRP parameters
Policy 2-Original MPS, New MRP parameters
Policy 3-Grouped MPS,New MRP parameters
Policy 4-No forecast MPS, new MRP parameters

Figure 10.2 A graph of load lumpiness vs policy combination under rolling schedule conditions



Shop policy key :

- Original
- Transport batch
- Bottleneck sequencer

MPS & MRP policy Key :

Policy 1-Original MPS & MRP
Policy 2-Original MPS, new MRP
Policy 3-Grouped MPS, new MRP
Policy 4-No forecast MPS, new MRP

Figure 10.3 A graph of average PCB WIP vs policy combination under rolling schedule conditions

production plan load measured against each of the policy combinations. The positive smoothing effect for policy combinations 3 and 4 was attributed to the induced staging of the MPS review period, which minimises the disruptive effect of MPS rescheduling.

From the point of view of FCL, the experiments revealed a number of general findings which were directly attributable to the simulated implementation of control policies and parameters under rolling schedule conditions. The introduction of the transport batch quantity and the bottleneck sequencing algorithm showed a substantial improvement in each of the shop performance monitors. This was particularly true for average PCB WIP which as shown in Figure 10.3 is almost halved when policy 1 is selected and a bottleneck sequencer is used in preference to the original shop floor policies . It is interesting to note that traditionally, the splitting up of large order quantities has been avoided on account of the extra work centre set-ups which may occur. The model has shown that careful selection of this average transfer quantity, with the use of an appropriate priority rule actually neutralises this effect and increases the dynamic utilisation of the work centre. Further improvements in shop performance were achieved in all experiments which adopted a linear lead time policy at the production control level with a grouped MPS configuration. The best combination was achieved using the bottleneck sequencer with policy 3 as illustrated in Figure 10.2. It was thus suggested that incorporating forecast demands in the MPS gave better performance than eliminating it all together. This, it was argued by Clarke(1988) is attributable to the 'banking' effect of excess stock caused by the propensity for uncertain demand when forecast is included in the MPS horizon. This 'banking' effect provides relief orders to cover unplanned changes and reduce the processing of recovery batches from MRP, thus providing a more stable input to the manufacturing system.

The results of the above experimental phase produced a number of interesting findings. These findings specifically relate to FCL's mode of operation, but their general implication can be applied to the manufacturing environment as a whole. The import of these findings are attributable to the introduction of uncertainty in the MPS. The majority of related studies in the area of production research have paid little regard to this aspect of model design and as argued in chapter 4, the conclusions produced from such studies must be reassessed in the light of a total modelling approach.

11 Research discussion

11.1 Generalised discussion

The development of a holistic approach to production system modelling and the subsequent design of an experimental facility for a relevant case study have been discussed.

The work was primarily concerned with addressing the needs of the production system management in their attempts to efficiently utilise limited resources in the manufacturing environment. It was argued that policy issues relating to both the integration of the major production system elements and the profitable running of the resulting facility should not be investigated independently. Instead, they should be reconciled with due cognizance of the interactive nature of the total system. It was shown however, that intuitive policy decision making was biased towards the particular objectives of the individual business functions. These factors have resulted in management resorting to existing knowledge and experience for policy decision support. A further important aspect of policy decision making was found to be the evaluation of one policy or policy combination against another. It was argued in chapter 2 that MRPII alone was unable to generate suitable performance criteria and was therefore incapable of assessing the success or otherwise of selected policy combinations. A modelling technique was thus sought for policy selection and analysis purposes. Modelling techniques were shown to be particularly suited to management decision support. It was however, argued that a number of simplifying assumptions had, in the past, limited studies to sub-sets of the total problem. The literature search has shown this approach to be of questionable validity and consequently raises doubts over previously accepted model results and conclusions.

The computer simulation based modelling philosophy was identified and its potential application in the light of traditional views in the manufacturing environment were discussed in chapter 4. Simulation was shown to be well suited to the dynamic environment of the production system, but it was emphasised that it makes no attempt to provide definitive answers or optimum solutions. The process of model analysis is instead concerned with trends and observations from experimental results. The literature revealed that many traditional simulation studies had been conducted for a variety of production system environments. A significant proportion of these studies were shown to have made simplistic assumptions about the real system they were modelling. More recent studies were presented which highlighted the importance of simulation model design for the generation of valid results and conclusions. It was thus argued that model validity was dependent on the validity of model data and elemental accuracy to such an extent that their neglect makes the model results highly questionable. Moreover, it was further argued that the conclusions from simplistic models are unsuitable as a basis for management decision support.

The need for a total modelling approach to control policy evaluation was addressed in chapter 5. The resulting approach was designed to incorporate sufficient realism in the simulation model whilst at the same time avoiding the overhead of redundant complexity. Model bounds were discussed in relation to the major production system elements which were considered to be the market place, production control and production supply. It was subsequently argued that the total approach philosophy would necessitate a model of the production control system driving an independent model of the production supply system. This methodology however, was only deemed to be feasible given the fulfilment of the following requirements;

i) an independent test database;

ii) data processing times which were acceptably rapid to enable extensive experimental analysis;

iii) reliable and accurate interface between the production control and production supply model. In addition, an operational mode which supported MPS demand uncertainty by allowing experiments to be conducted under interactive rolling schedule conditions was deemed to be mandatory.

Chapter 6 introduced a case study which involved the recently formed manufacturing company, Fulcrum Communications Plc. The lack of relevant operational experience coupled with the major system investments made raised a number of major issues in line with those discussed earlier in chapter 2 . This offered a suitable environment to apply the total modelling approach philosophy to a live industrial problem. Specifications for the proposed experimental facility were discussed in the light of the particular requirements of FCL and the system investments made. The specification options were discussed in chapter 7 which formed the basis of the model feasibility assessment.

A detailed discussion of the resulting total FCL model was included in chapter 8., whilst the validity of the model was addressed in chapter 9. The methodology adopted for this work was to produce a detailed simulation of the manufacturing, assembly and test facilities and to link their consumption and output via the MRP II system under investigation. Thus, an experimental facility was produced which covered both the manufacturing hardware and the production control system. Furthermore, the scope of the model was such that all of the interactions of the real system were reproduced. The FCL model incorporated a number of innovative features which contributed to its effectiveness as a tool for management decision support. These included;

i) a real MRPII system driving a comprehensive manufacturing simulator;

ii) highly configurable model elements which allow the selection of numerous control policy combinations at all system model levels;

iii) the capability of modelling the whole product range with the associated information for each part;

iv)the ability to conduct experiments under interactive rolling schedule conditions allowing the investigation of MPS demand uncertainty;

v) the ability to realistically simulate the interactions between job progression and resource constraints in the manufacturing system model;

vi) the ability of the two types of computer to interactively communicate with one another, thus simulating the role of the appropriate computer user.

A sub-set of the experimental results and conclusions obtained using the FCL model was discussed in chapter 10. The results showed that demand uncertainty in the MPS was a major factor and consequently influenced both the quality of the production plan and the performance of the manufacturing facility. This was an important finding since many previous conclusions from past research studies have been made in the absence of MPS uncertainty.

11.2 Future development work

The guidelines for future developments have been targeted at extending the work discussed in this research. Fundamental to this extended work is the continued adoption of a total modelling approach for management decision support.

To date, use of the production system model at FCL has concentrated on the choice of policies and parameters for the MRP II system, against the current configuration of manufacturing resources and the current product data base. However, the approach adopted for the design of the model, was such that it could be adapted to investigate a number of additional and important issues relating to short and medium term manufacturing capability with a given production plan. These issues include the investigation of alternative job routing, product mix, transport and process batch size, and shift work. The primary aim would be to support online planning, scheduling and sequencing decisions. In addition, longer term issues could be assessed with regard to manufacturing facility configuration. The recently installed manufacturing facility at FCL has numerous potential configurations, each appropriate to a particular set of requirements. For example, the PCB facility could be operated as a flow line dedicated at any particular moment to the production of one type of PCB. Alternatively it could be configured for mixed batch production using manufacturing cells thus, forming the basis for Group Technology . The suitability of a particular configuration is dependent on the size, type and frequency of demands for PCBs. This in turn is dependent on the Business Strategy and the production control policies.

The proposed model would need the capability of capturing the current status of the real manufacturing system(i.e. stock levels, job progress, machine states, available resource etc.) at any time and simulating the effect of possible policy decisions. It therefore follows that this model would be greatly enhanced by using the 'live' database and production control system(on the A9 mainframe computer) as its driving mechanism. This could be achieved by electronically linking the A9 computer to the proposed manufacturing simulator in the same way as the B5900 was linked in the original FCL model. The schematic diagram in Figure 11.1 gives a conceptual overview of the proposed model, and highlights the required inputs and suggested outputs. In addition to the above, the manufacturing simulator would need to incorporate a number of facilities and features in order to accurately assess the input data from the real production control system. These include;

i) the representation of process stoppage or machine breakdown;

ii) visually interactive 'what if' scenarios;

iii)a user friendly approach to changing the manufacturing facility configuration and layout;

iV)the ability to interactively modify available resources;

v) the capability to assess specific policies designed to schedule and sequence job batches;

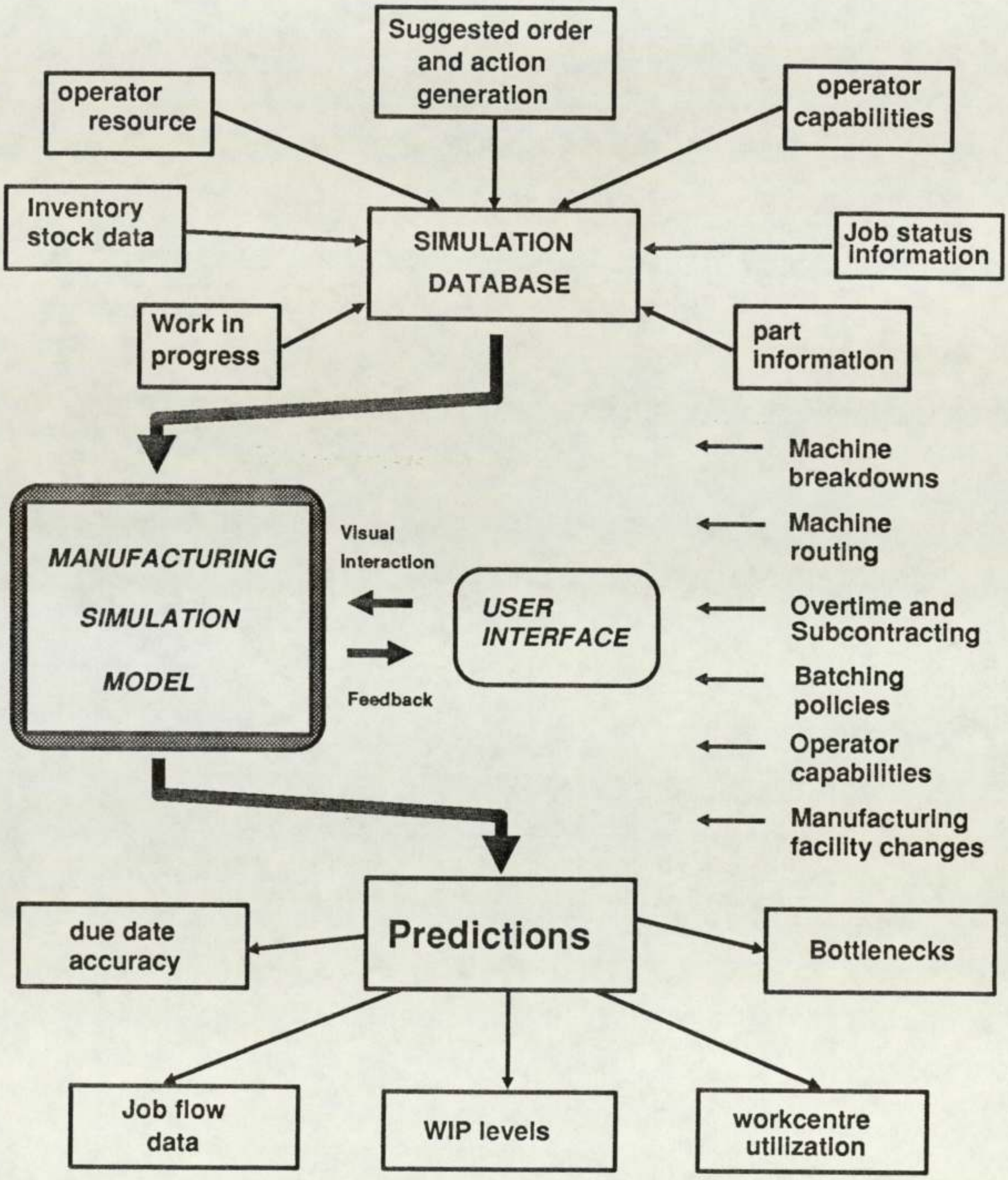


Figure 11.1 Schematic diagram of future online simulator

vi) the capability to frequently update the model status in real time in order to reflect the real system;

vii) a user friendly facility to incorporate new model part numbers with their associated route, operations and BOM details. The resulting output data would be used to assess the consequences of adopting a particular operational decision or a change in the manufacturing system configuration.

At a more general level, further developments could include the adoption of the total modelling approach to a market analysis model. The important requirements of such a model would include the following;

i). ability to assess what type of market a company should be in given available equipment, company objectives, technological changes and competition;

ii).ability of the company to realistically handle the demands of its current market;

iii).capability to allow the model user to explore the feasibility of manufacturing for alternative market sectors.

A model with the above capability would need to include a financial element. This would be used to assess the financial implications of specific strategic decisions, the ability of the company to attain a satisfactory return on investment (ROI) and consequently determine how efficiently the company investments are being utilised.

These suggestions for further work have been included because the author believes that they are commercially as well as academically important, especially given the increasing volatility of the market place and availability of high technology.

APPENDICES

APPENDIX I LIST OF MODELLED PART NUMBERS

Simulation model No.	TMS model No.
1	3HAA-00030AAL 1
2	3HEF-00126AAW 2
3	3HUR-00009AAM 3
4	ADUM-GATEWAY-1024
5	3HEF-00094AAP 5
6	3HUA-00003AAA 6
7	3HUA-00004AAE 7
8	3HEF-00103AAB 8
9	3HUC-00019AAC 9
10	3HEF-00067AAD 10
11	3HEB-00257AAJ 11
12	AEQP-V/DATA-102 12
13	3HEF-00085AAK 13
14	3HEF-00085ADF 14
15	3HEM-00102AAV 15
16	3HEF-00099AAH 16
17	3HES-00028AAL 17
18	ADUM-QUERC-102 18
19	AEQP-QUERC-101 19
20	AKIT-EDGELEY-11920
21	3HEU-00028AAG 21
22	3HEU-00029AAH 22
23	3HEF-00056AAY 23
24	3HEF-00054AAW 24
25	AEQP-EDG36-000 M25
26	3HEM-00080AAJ 26
27	AKIT-EDGELEY-12327
28	AKIT-EDGELEY-12428
29	ADUM-GATEWAY-10329
30	3HEF-00089AAS 30
31	3HEF-00097AAX 31
32	3HKA-00046AAV 32
33	3HEM-00022AAY 33
34	3HEF-00093AAM 34
35	3HUA-00005AAH 35
36	3HUA-00006AAJ 36
37	3HEF-00043AAP 37
38	3HEM-00117AAX 38
39	3HEC-00007AAZ 39
40	3HEM-00107AAY 40
41	3HER-00001ABK 41
42	3HES-00003AAF 42
43	3HEA-00017AAG 43
44	3HES-00004AAG 44
45	3HES-00005AAH 45
46	NOT MODELLED
47	ADUM-EDGELEY-10347
48	3HEF-00051AAT 48
49	3HEU-00046AAJ 49
50	AEQP-EDGELEY-10850
51	AKIT-EDGELEY-12151
52	AEQP-EDGELEY-10952

53	AKIT-EDGELEY-128	53
54	3HEU-00030AAJ	54
55	AKIT-EDGELEY-127	55
56	3HEF-00044AAQ	56
57	3HEM-00130AAY	57
58	ADUM-ECTE-102	58
59	3HEA-00016AAE	59
60	3HUA-00009AAZ	60
61	3HEM-00108ABA	61
62	3JAN-00043AAQ	62
63	3HAA-00011AAH	63
64	ADUM-PLANNING-01	64
65	ADUM-QUADMUX-101	65
66	3HEF-00128AAY	66
67	ADUM-AHS-102	67
68	3HEM-00029AAZ	68
69	3HUC-00033AAZ	69
70	3HEF-00122AAT	70
71	ADUM-M4000-01	71
72	ADUM-M4000-02	72
73	ADUM-M960-01	73
74	3HKA-00153AAN	74
75	3HUC-00022AAX	75
76	3HUB-00071AAZ	76
77	3HUB-00071ABA	77
78	3HEM-00023AAA	78
79	3HEM-00024AAB	79
80	3HEP-00045AAH	80
81	3HEM-00109ABB	81
82	3HEP-00043AAE	82
83	3HEP-00044AAG	83
84	3HEP-00038ABG	84
85	3HEM-00089AAE	85
86	3HEM-00072AAB	86
87	3HEF-00048AAY	87
88	3HEF-00072AAM	88
89	3HEM-00091AAZ	89
90	3HEM-00092AAB	90
91	3HER-00002AAH	91
92	3HUC-00023AAX	92
93	NOT MODELLED	
94	3HEC-00001AAJ	94
95	3HUB-00062AAZ	95
96	3HEC-00002AAW	96
97	3HUC-00033ABA	97
98	3HEU-00024AAL	98
99	3HEM-00130ABZ	99
100	3HEB-00631AAV	100
101	3HEB-00693AAE	101
102	3HEB-00692AAD	102
103	3HEB-00704AAR	103
104	3HEB-00700AAM	104
105	3HEB-00705AAS	105
106	3HEB-00344AAX	106
107	3HEB-00706AAT	107

108	3HEB-00699AAL	108
109	3HEB-00694AAF	109
110	3HEB-00695AAG	110
111	3HEB-00696AAH	111
112	3HEB-00697AAJ	112
113	3HEB-00324AAD	113
114	3HEB-00317AAC	114
115	3HEB-00582AAV	115
116	3HEB-00234ABR	116
117	3HEB-00235AAH	117
118	3HEB-00236AAJ	118
119	3HEB-00237AAK	119
120	3HEB-00238AAL	120
121	3HEB-00239AAM	121
122	3HEB-00241AAP	122
123	3HEB-00242AAQ	123
124	3HEB-00243AAR	124
125	3HEB-00244AAS	125
126	3HEB-00245AAT	126
127	3HEB-00277ABL	127
128	3HEB-00278AAY	128
129	3HUB-00077AAP	129
130	3HUB-00072AAA	130
131	3HEB-00280AAA	131
132	3HEB-00309AAP	132
133	3HEB-00310AAQ	133
134	3HUB-00072ACD	134
135	3HEB-00313AAT	135
136	3HEB-00312AAS	136
137	3HEB-00583AAW	137
138	3HEB-00665AAH	138
139	3HEB-00343AAW	139
140	3HEB-00206AAL	140
141	3HEB-00196AAE	141
142	3HEB-00314AAU	142
143	3HUB-00071ACC	143
144	3HEB-00564AAX	144
145	3HEB-00565AAY	145
146	3HEB-00566AAZ	146
147	3HEB-00567AAA	147
148	3HEB-00568AAB	148
149	3HEB-00075AAK	149
150	3HEB-00076AAM	150
151	3HEB-00250AAB	151
152	3HEB-00251AAC	152
153	3HEB-00610AAA	153
154	3HEB-00264AAR	154
155	3HEB-00262AAP	155
156	NOT MODELLED	
157	3HEB-00258AAK	157
158	3HEB-00163AAV	158
159	3HEB-00143AAX	159
160	3HEB-00233ABG	160
161	3HEB-00279AAZ	161
162	3HEB-00234AAG	162

163	3HEB-00321AAH	163
164	3HEB-00277AAX	164
165	3HEB-00157AEV	165
166	3HEB-00535AAV	166
167	3HEB-00536AAW	167
168	3HUB-00070AEB	168
169	3HUB-00085AAR	169
170	3HUB-00062ABV	170
171	3HEB-00728AAS	171
172	3HEB-00013AAY	172
173	3HEB-00087AAP	173
174	3HEB-00001AAY	174
175	3HEB-00002AAG	175
176	3HEB-00003AAH	176
177	3HEB-00004AAJ	177
178	3HEB-00005AAK	178
179	3HEB-00007AAM	179
180	3HEB-00053AAL	180
181	3HEB-00015AAD	181
182	3HEB-00267AAU	182
183	3HEB-00253AAE	183
184	3HEB-00331AAW	184
185	3HEB-00252AAD	185
186	3HEB-00157ADU	186
187	3HEB-00537AAX	187
188	3HEM-00148AAW	188
189	3HEB-00606AAW	189
190	3HEB-00154AAS	190
191	3HEB-00150AAN	191
192	3HEB-00534AAT	192
193	3HEB-00118AAV	193
194	3HEB-00153ABR	194
195	3HEB-00155AAT	195
196	3HEB-00128AAK	196
197	3HEB-00096AAX	197
198	3HEB-00152AAQ	198
199	3HEB-00165AAN	199
200	3HEB-00151AAP	200
201	3HEB-00741AAF	201
202	3HEB-00739AAD	202
203	3HEB-00650AAA	203
204	3HEB-00651AAB	204
205	3HEB-00620AAV	205
206	3HEB-00619AAU	206
207	3HEB-00617AAS	207
208	3HEB-00613AAY	208
209	3HEB-00615AAA	209
210	3HEB-00618AAT	210
211	3HEB-00614AAZ	211
212	3HEB-00616AAB	212
213	3HEB-00094AAE	213
214	3HEB-00072AAM	214
215	3HEB-00071AAL	215
216	3HEB-00070AAK	216
217	3HEB-00067AAZ	217

218	3HEB-00064AAA	218
219	3HEB-00060AAL	219
220	3HEB-00073AAN	220
221	3HEB-00119AAD	221
222	3HEB-00684AAN	222
223	3HEB-00729AAT	223
224	3HEB-00728AAS	224
225	3HEB-00756AAW	225
226	3HEB-00753AAT	226
227	3HEB-00738AAC	227
228	3HEB-00749AAP	228
229	3HEB-00763AAD	229
230	3HEB-00233AAF	230
231	3HEB-00281AAB	231
232	3HEB-00311AAR	232
233	3HEB-00316AAW	233
234	3HEB-00605AAV	234
235	3HEB-00607AAX	235
236	3HEB-00209AAS	236
237	3HEB-00586AAH	237
238	3HEB-00604AAU	238
239	3HEM-00143AAN	239
240	3HEM-00142AAM	240
241	3HEU-00043AAJ	241
242	3HEU-00016AAC	242
243	3HUB-00061AAY	243

APPENDIX ii LIST OF ASSEMBLY ROUTE AND OPERATION DATA

Model no.
Workcentre no.
process time(mins.)

1	
18.0000	10.0000
2	
18.0000	10.0000
3	
18.0000	240.0000
21.0000	60.0000
4	
18.0000	50.0000
5	
18.0000	12.0000
19.0000	60.0000
6	
15.0000	12.6780
15.0000	7.8900
16.0000	1.1840
15.0000	13.0100
16.0000	1.9650
15.0000	9.5500
16.0000	1.4330
15.0000	8.7600
16.0000	1.3140
15.0000	30.0000
16.0000	4.5000
15.0000	19.1800
16.0000	2.8770
7	
15.0000	1.6300
15.0000	3.0800
16.0000	.6100
15.0000	5.0900
16.0000	.6100
8	
15.0000	4.2780
15.0000	8.7300
16.0000	1.3100
15.0000	12.6600
16.0000	1.9000
17.0000	2.1390
9	
18.0000	73.4700
18.0000	8.5620
18.0000	56.0000

18.0000	93.9980
21.0000	84.1800
18.0000	13.0020
18.0000	43.0020
21.0000	84.1800
18.0000	19.0000
21.0000	84.1800
18.0000	17.0000
18.0000	75.0000
18.0000	85.0000
18.0000	14.7960
18.0000	32.0000
18.0000	14.0020
18.0000	34.0020
21.0000	14.1800
18.0000	41.9980
21.0000	14.1800
21.0000	14.1800
19.0000	36.7300

10

15.0000	23.2860
15.0000	5.2860
16.0000	.7920
15.0000	5.0400
16.0000	.7560
15.0000	.7500
16.0000	.1080
15.0000	1.9980
16.0000	.3000
15.0000	5.7480
16.0000	.8580
15.0000	6.1500
16.0000	.9180
15.0000	2.7600
16.0000	.4140
15.0000	30.6300
16.0000	4.9500
15.0000	10.6080
16.0000	1.5900
15.0000	16.6860
16.0000	2.5020
17.0000	32.7960

11

15.0000	8.3520
16.0000	1.2540
15.0000	10.0380
16.0000	1.2540

12

18.0000	5.0000

13

18.0000	180.0000

21.0000	20.0000

14

15.0000	11.1100
15.0000	1.3120
16.0000	.1900
15.0000	9.7200
16.0000	1.4500
15.0000	13.4660
16.0000	2.0100
15.0000	6.1330
16.0000	.9100
15.0000	1.9360
16.0000	.9200
15.0000	2.4780
16.0000	.3700
15.0000	2.2000
16.0000	.3300
15.0000	12.2790
16.0000	1.8400
15.0000	6.1770
16.0000	.9200

15

15.0000	5.0000
15.0000	6.0000
16.0000	.9000

16

15.0000	15.1020
15.0000	6.1500
16.0000	.9180
15.0000	2.5080
16.0000	.3720
15.0000	1.0740
16.0000	1.0020
15.0000	9.6960
16.0000	1.4520
15.0000	10.6260
16.0000	1.5900
15.0000	2.4960
16.0000	.3720
15.0000	4.3500
16.0000	.6480
15.0000	7.8180
16.0000	1.1700
15.0000	1.9980
16.0000	.3000
15.0000	5.7480
16.0000	.8580
15.0000	6.7500
16.0000	1.0080
15.0000	12.6480
16.0000	1.8960
19.0000	19.9980

21.0000	9.9960

17

15.0000	99.1260
16.0000	6.6480
15.0000	27.0120
16.0000	6.6480
15.0000	68.0340
16.0000	6.6480
15.0000	25.8600
16.0000	6.6480
15.0000	1.5480
16.0000	6.6480

18

18.0000	60.0000
20.0000	180.0000
19.0000	120.0000
21.0000	18.0000

19

18.0000	5.0000
21.0000	5.0000

20

18.0000	10.0000

21

15.0000	15.9720
15.0000	11.0880
16.0000	1.7100
15.0000	24.6000
16.0000	1.7100
15.0000	8.0760
15.0000	5.0640
16.0000	1.7100
15.0000	4.8360
16.0000	1.7100
15.0000	22.3020
16.0000	1.7100
15.0000	2.0100
16.0000	1.7100
15.0000	1.8840
16.0000	1.7100
19.0000	7.9860

22

15.0000	10.6560
15.0000	7.7400
16.0000	1.1580
15.0000	13.2420
16.0000	1.9860
15.0000	2.3400
16.0000	.3480
19.0000	30.0000

23

15.0000	2.7360
15.0000	1.8300
16.0000	.3420
15.0000	5.0820
16.0000	.3420
16.0000	1.7580
16.0000	.3420
15.0000	3.4740
16.0000	.3420
15.0000	1.5840
16.0000	.3420
19.0000	30.0000
21.0000	.3420

24

15.0000	13.9560
15.0000	.5760
16.0000	.0840
15.0000	6.7320
16.0000	1.0080
15.0000	6.0900
16.0000	.9120
15.0000	5.8620
16.0000	.8760
15.0000	25.6860
16.0000	3.8520
15.0000	2.4480
16.0000	.3660
19.0000	22.3980
21.0000	38.5200

25

18.0000	6.0000
19.0000	30.0000
18.0000	30.0000

26

15.0000	28.7700
15.0000	3.9960
16.0000	.6000
15.0000	18.7380
16.0000	2.8080
15.0000	19.9800
16.0000	2.9940
15.0000	92.8560
16.0000	13.9260
15.0000	8.2680
16.0000	1.2360

27

18.0000	10.0000

28

```
18.0000  10.0000

29
18.0000  10.0000

30
15.0000   6.0900
16.0000    .7800
15.0000   4.3800
16.0000    .7800
15.0000  32.3880
16.0000   3.3540
19.0000  15.0000

31
15.0000   6.0900
16.0000    .7800
15.0000   4.3800
16.0000    .7800
15.0000  22.3260
16.0000   3.3480
19.0000  15.0000

32
18.0000 500.0000

33
15.0000   4.2480
16.0000    .6360

34
19.0000  75.0000
18.0000  30.0000

35
15.0000   8.0280
16.0000   1.2000
15.0000  22.9860
16.0000   3.4440
15.0000 204.9960
16.0000  30.7500
15.0000   9.7080
16.0000   1.4520
15.0000   6.1080
16.0000    .9120
15.0000   5.1180
16.0000    .7680
15.0000   6.0780
16.0000    .9120
15.0000   4.1400
16.0000    .6180
15.0000   6.5460
16.0000    .9780

36
```

```
15.0000   3.7260
16.0000   1.8300
15.0000  13.6560
16.0000   1.8300
15.0000  19.1160
16.0000   1.8300

37
19.0000  60.0000
21.0000   9.0000

38
18.0000  13.6380
21.0000   2.0460

40
15.0000 369.6000
16.0000  55.4400

41
18.0000  15.0000
18.0000  60.0000
18.0000  15.0000
21.0000  12.0000
18.0000  10.0000
18.0000  15.0000
18.0000 120.0000
21.0000  20.0000
18.0000  10.0000
18.0000  60.0000
18.0000 120.0000
21.0000  30.0000
18.0000  60.0000
18.0000  10.0000
18.0000   5.0000
18.0000 480.0000
21.0000  83.0000
18.0000 180.0000
18.0000  60.0000
18.0000  30.0000
18.0000  20.0000
18.0000  15.0000
19.0000 240.0000
19.0000 500.0000

43
15.0000 420.0000
16.0000  63.0000
18.0000 120.0000
20.0000  18.0000

47
18.0000 600.0000
20.0000 180.0000
19.0000 180.0000
```

21.0000	60.0000

48

15.0000	5.0500
16.0000	1.2800
15.0000	1.6400
16.0000	1.2800
15.0000	10.3400
16.0000	1.2800
15.0000	11.4000
16.0000	1.2800
15.0000	14.2400
16.0000	1.2800
19.0000	.4000

49

15.0000	5.2240
16.0000	.9870
15.0000	7.9430
16.0000	.9870
19.0000	24.0000

50

18.0000	10.0000
21.0000	2.0000
22.0000	3.0000

51

18.0000	5.0000

52

18.0000	5.0000
21.0000	2.0000
22.0000	3.0000

53

18.0000	5.0000
21.0000	2.0000
22.0000	3.0000

54

15.0000	18.4900
16.0000	1.8700
15.0000	27.0600
16.0000	1.8700
15.0000	7.0900
16.0000	1.8700
15.0000	1.7500
16.0000	1.8700
15.0000	7.9000
16.0000	1.8700
19.0000	43.2000

55

18.0000	10.0000

56
19.0000 60.0000
21.0000   9.0000

57
15.0000   3.0000
16.0000   .4500
15.0000   5.0000
16.0000   .7500
15.0000 12.0000
16.0000   1.8000
15.0000 16.0000
16.0000   2.4000

58
18.0000 960.0000
20.0000 180.0000
19.0000 180.0000
21.0000 60.0000

59
18.00001080.0000
19.0000 240.0000

60
18.00001300.0000
19.0000 200.0000

61
15.0000 50.0000
16.0000   7.5000

62
15.0000 50.0000
16.0000   8.0000

63
15.0000 280.0000
16.0000 42.0000

66
18.0000 10.0000
19.0000 10.0000
21.0000   8.0000

67
18.0000 480.0000
19.0000 240.0000
20.0000 180.0000
19.0000 180.0000
21.0000 60.0000

68
18.0000 180.0000

```
20.0000 180.0000
19.0000 180.0000
21.0000  12.0000

69
18.0000  60.0000
21.0000   6.0000

70
18.0000 180.0000
20.0000 180.0000
19.0000 180.0000
21.0000  12.0000

74
18.0000  18.0000

75
15.0000   8.3200
16.0000   5.7900
15.0000  68.8800
16.0000   5.7900

78
18.0000   5.0000
21.0000    .7500
18.0000   8.2310
21.0000   1.2300
18.0000   2.0000
21.0000    .3000
18.0000  16.0000
21.0000   2.4000
18.0000   4.2000
21.0000    .6300
18.0000   3.1500
21.0000    .4730

79
18.0000   5.0000
21.0000    .7500
18.0000   8.2310
21.0000   1.2300
18.0000   2.0000
21.0000    .3000
18.0000  16.0000
21.0000   2.4000
18.0000   4.2000
21.0000    .6300
18.0000   3.1500
21.0000    .4730

80
18.0000   6.7780
21.0000   1.0170
```

81

15.0000	120.0000
16.0000	13.0000

82

15.0000	75.0000
16.0000	10.0000

83

15.0000	60.0000
16.0000	10.0000

84

15.0000	77.9500
16.0000	11.6920

85

18.0000	10.0000
21.0000	1.5000

86

18.0000	8.0000
21.0000	1.2000

87

18.0000	20.0000
21.0000	3.0000

88

15.0000	150.0000
16.0000	22.5000

89

18.0000	60.0000
21.0000	9.0000

90

18.0000	4.9980
21.0000	.7500

91

15.0000	30.1800
16.0000	4.5270
15.0000	58.1400
16.0000	8.7210
15.0000	87.1800
16.0000	13.0770
15.0000	48.6000
16.0000	7.2900
15.0000	8.9600
17.0000	20.0000

92

15.0000	15.0000
16.0000	2.2500

15.0000	3.0000
16.0000	.4500
15.0000	10.0000
16.0000	1.5000
15.0000	12.0000
16.0000	1.8000
15.0000	17.0000
16.0000	2.5000
15.0000	12.0000
16.0000	1.8000
17.0000	15.0000

94

15.0000	4.3680
16.0000	11.9400
15.0000	14.0940
16.0000	11.9400
15.0000	87.4740

95

15.0000	41.5800
16.0000	6.2700

96

15.0000	4.3650
16.0000	16.4500
15.0000	17.8790
16.0000	16.4500
15.0000	51.3460
16.0000	16.4500
15.0000	344.5920
15.0000	20.6200
16.0000	16.4500
17.0000	15.0000

97

18.0000	60.0000
21.0000	6.0000

98

15.0000	41.5800
16.0000	6.2700

99

15.0000	41.5800
16.0000	6.2700

APPENDIX iii LIST OF PCB ROUTE AND OPERATION DATA

Type
Workcentre no.
process time(mins.)
set time(mins.)

Type	Workcentre no.	process time(mins.)	set time(mins.)
100			
	2.0000	.7860	35.0000
	4.0000	1.7700	10.0000
	4.0000	.7860	5.0000
	8.0000	.2160	15.0000
	9.0000	6.0240	7.0000
	10.0000	1.0200	.0000
	9.0000	2.0700	3.0000
	10.0000	1.0200	.0000
	11.0000	1.3560	10.0000
101			
	2.0000	1.3020	35.0000
	4.0000	2.8740	10.0000
	4.0000	.8640	5.0000
	7.0000	1.2300	.0000
	8.0000	.2640	15.0000
	9.0000	2.0760	7.0000
	10.0000	1.2300	.0000
	9.0000	15.6900	3.0000
	10.0000	1.2300	.0000
	11.0000	2.4660	10.0000
102			
	9.0000	12.6840	7.0000
	10.0000	1.9560	.0000
103			
	9.0000	2.8140	7.0000
	10.0000	.4800	.0000
	11.0000	.3180	10.0000
104			
	9.0000	7.2480	7.0000
	10.0000	.7500	.0000
	11.0000	1.0080	10.0000
	9.0000	1.8420	3.0000
	10.0000	.7500	.0000
105			
	9.0000	17.5680	7.0000
	10.0000	2.6880	.0000
107			
	9.0000	7.1640	7.0000
	10.0000	1.1280	.0000
108			
	4.0000	2.4900	10.0000
	1.0000	2.1000	50.0000

7.0000	1.1220	.0000
2.0000	.8520	45.0000
3.0000	.4380	50.0000
5.0000	3.4260	72.0000
7.0000	1.1220	.0000
8.0000	.3120	15.0000
9.0000	9.9120	14.1000
10.0000	1.1220	.0000
11.0000	2.2500	10.0000

109

1.0000	1.1520	40.0000
7.0000	.6960	.0000
2.0000	.6960	35.0000
5.0000	3.1020	57.0000
7.0000	.6960	.0000
8.0000	.2160	15.0000
9.0000	7.7340	7.0000
10.0000	.6960	.0000
11.0000	1.3980	10.0000

110

1.0000	.8340	70.0000
7.0000	.4900	.0000
2.0000	1.2710	55.0000
3.0000	.3010	70.0000
4.0000	.7070	10.0000
6.0000	1.7630	125.0000
7.0000	.4900	.0000
8.0000	.2670	15.0000
9.0000	3.3870	19.0000
10.0000	.4900	.0000
9.0000	.6550	8.1000
11.0000	.9800	10.0000

111

1.0000	.7850	70.0000
7.0000	.4900	.0000
2.0000	1.1960	55.0000
3.0000	.3120	70.0000
4.0000	.7820	10.0000
6.0000	1.9730	125.0000
7.0000	.4900	.0000
8.0000	.2670	15.0000
9.0000	3.3870	19.0000
10.0000	.4900	.0000
9.0000	.6650	8.1000
11.0000	.9900	10.0000

112

1.0000	.7830	70.0000
7.0000	.4400	.0000
2.0000	1.0290	55.0000
3.0000	.3230	70.0000
4.0000	1.0240	10.0000

5.0000	1.8430	125.0000
7.0000	.4400	.0000
8.0000	.2510	15.0000
9.0000	2.9520	19.0000
10.0000	.4400	.0000
11.0000	.8800	10.0000

113

4.0000	2.4900	10.0000
1.0000	2.1000	50.0000
7.0000	1.1220	.0000
2.0000	.8520	45.0000
3.0000	.4380	50.0000
5.0000	3.4260	72.0000
7.0000	1.1220	.0000
8.0000	.3120	15.0000
9.0000	9.9120	14.1000
10.0000	1.1220	.0000
11.0000	2.2500	10.0000

114

1.0000	1.0920	50.0000
2.0000	.8760	45.0000
4.0000	.9180	10.0000
4.0000	1.9620	5.0000
4.0000	2.4480	5.0000
5.0000	5.9700	72.0000
7.0000	1.0980	.0000
8.0000	.2760	15.0000
9.0000	4.4520	14.1000
10.0000	1.0980	.0000
9.0000	5.9100	6.1000
10.0000	1.0980	.0000
11.0000	2.9700	10.0000

115

9.0000	4.8540	7.0000
10.0000	.7860	.0000
11.0000	.5220	10.0000

116

1.0000	1.2720	40.0000
2.0000	.9060	35.0000
4.0000	2.7420	10.0000
4.0000	1.9260	5.0000
5.0000	3.6780	57.0000
7.0000	1.7280	.0000
8.0000	.2700	15.0000
9.0000	7.6500	7.0000
10.0000	1.7280	.0000
9.0000	8.0460	3.0000
10.0000	1.7280	.0000
9.0000	.8820	3.0000
10.0000	1.7280	.0000
11.0000	7.0680	10.0000

117

2.0000	.8760	35.0000
4.0000	.4440	10.0000
4.0000	1.9620	10.0000
5.0000	3.3660	57.0000
7.0000	1.1580	.0000
8.0000	.2400	15.0000
9.0000	3.8220	7.0000
9.0000	2.3520	3.0000
10.0000	1.1640	.0000
11.0000	1.6500	10.0000

118

4.0000	2.1900	10.0000
1.0000	1.7700	40.0000
4.0000	1.9620	10.0000
4.0000	.4680	5.0000
7.0000	.6900	.0000
8.0000	.1980	15.0000
9.0000	5.2080	7.0000
10.0000	.6900	.0000
9.0000	4.2660	3.0000
10.0000	.6900	.0000
11.0000	1.8420	10.0000

119

2.0000	.9240	35.0000
4.0000	2.6400	10.0000
4.0000	.1920	5.0000
5.0000	1.6800	57.0000
7.0000	1.5240	.0000
8.0000	.2100	15.0000
9.0000	4.5240	7.0000
10.0000	1.5180	.0000
9.0000	12.5760	3.0000
10.0000	1.5180	.0000
11.0000	2.7300	10.0000

120

1.0000	.8280	40.0000
2.0000	.8940	35.0000
4.0000	.7740	10.0000
5.0000	1.8660	57.0000
7.0000	.8100	.0000
4.0000	1.9620	10.0000
8.0000	.2400	15.0000
9.0000	3.4680	7.0000
9.0000	4.1460	3.0000
10.0000	.8100	.0000
11.0000	1.6260	10.0000

121

2.0000	.8760	45.0000
4.0000	3.2220	10.0000

4.0000	1.9620	5.0000
5.0000	3.0840	72.0000
7.0000	2.4060	.0000
8.0000	.2160	15.0000
9.0000	4.6920	14.1000
10.0000	1.6080	.0000
9.0000	15.0660	6.1000
10.0000	1.6080	.0000
11.0000	3.2040	10.0000

122

1.0000	1.0920	40.0000
2.0000	.8280	35.0000
4.0000	.7740	10.0000
4.0000	1.9620	5.0000
4.0000	.5520	5.0000
7.0000	.7260	.0000
8.0000	.2100	15.0000
9.0000	4.5180	7.0000
10.0000	.5400	.0000
9.0000	2.6040	3.0000
10.0000	.5400	.0000
11.0000	1.4400	10.0000

123

4.0000	4.2600	10.0000
1.0000	.9960	50.0000
2.0000	1.0440	45.0000
4.0000	1.9620	10.0000
5.0000	3.7680	72.0000
7.0000	1.2060	.0000
8.0000	.2640	15.0000
9.0000	8.8260	14.1000
10.0000	1.2060	.0000
9.0000	5.5620	6.1000
10.0000	1.2060	.0000
11.0000	5.1480	10.0000

124

4.0000	1.9320	10.0000
4.0000	1.7580	5.0000
5.0000	3.7980	57.0000
7.0000	1.4280	.0000
8.0000	.1980	15.0000
9.0000	5.3220	7.0000
10.0000	1.4280	.0000
9.0000	3.1260	3.0000
10.0000	1.4280	.0000
11.0000	1.9020	10.0000

125

2.0000	.5520	35.0000
4.0000	1.8660	10.0000
4.0000	1.9620	5.0000
4.0000	.7860	5.0000

7.0000	1.6800	.0000
8.0000	.1860	15.0000
9.0000	3.4740	7.0000
9.0000	11.5200	3.0000
10.0000	1.6860	.0000
11.0000	2.2380	10.0000

126

2.0000	.8180	35.0000
4.0000	2.1910	10.0000
4.0000	1.9660	10.0000
5.0000	2.1320	57.0000
7.0000	2.7700	.0000
8.0000	.2100	15.0000
9.0000	14.3830	7.0000
10.0000	2.7720	.0000
9.0000	3.8200	3.0000
10.0000	2.7720	.0000
11.0000	2.8340	10.0000

127

4.0000	1.9620	10.0000
4.0000	1.7820	5.0000
4.0000	.6900	5.0000
4.0000	.9900	5.0000
7.0000	1.5060	.0000
8.0000	.1860	15.0000
9.0000	5.0100	7.0000
9.0000	2.3880	3.0000
10.0000	1.5060	.0000
11.0000	1.5300	10.0000

128

4.0000	3.2220	10.0000
1.0000	1.6440	40.0000
4.0000	1.9620	10.0000
4.0000	.4680	5.0000
7.0000	1.1400	.0000
8.0000	.1980	15.0000
9.0000	5.1240	7.0000
9.0000	7.4100	3.0000
10.0000	1.1460	.0000
11.0000	2.2740	10.0000

129

4.0000	2.9590	10.0000
4.0000	8.6240	5.0000
7.0000	3.4800	.0000
8.0000	.2190	15.0000
9.0000	19.0480	14.1000
10.0000	3.4800	.0000
9.0000	35.7780	6.1000
10.0000	3.4800	.0000
11.0000	6.9600	10.0000

130

4.0000	6.3230	10.0000
4.0000	8.2670	5.0000
7.0000	3.5700	.0000
8.0000	.1880	15.0000
9.0000	1.5570	14.1000
10.0000	3.5700	.0000
9.0000	6.6810	6.1000
10.0000	3.5700	.0000
9.0000	7.7370	6.1000
10.0000	3.5700	.0000
11.0000	3.3600	10.0000

131

1.0000	1.2180	40.0000
2.0000	.6660	35.0000
4.0000	.7980	10.0000
5.0000	2.0160	57.0000
7.0000	.6720	.0000
8.0000	.2160	15.0000
9.0000	5.8080	7.0000
10.0000	.6720	.0000
9.0000	4.8660	3.0000
10.0000	.6720	.0000
11.0000	1.8000	10.0000

132

9.0000	22.4700	19.0000
10.0000	3.5100	.0000
9.0000	23.4060	8.1000
10.0000	3.5100	.0000
11.0000	4.6860	10.0000

133

1.0000	1.1520	70.0000
2.0000	1.5240	55.0000
3.0000	.3660	70.0000
4.0000	1.1040	10.0000
5.0000	7.1640	125.0000
7.0000	1.4760	.0000
8.0000	.3720	15.0000
9.0000	.3900	19.0000
9.0000	9.9120	8.1000
10.0000	1.4760	.0000
11.0000	2.9460	10.0000

134

4.0000	9.9900	10.0000
4.0000	5.3900	5.0000
7.0000	3.5700	.0000
8.0000	.5800	15.0000
9.0000	2.4600	14.1000
9.0000	14.4450	6.1000
9.0000	6.6100	6.1000
10.0000	3.5700	.0000

135

9.0000	1.8420	7.0000
10.0000	.2760	.0000
11.0000	2.2800	10.0000

136

2.0000	.8460	35.0000
4.0000	1.6740	10.0000
5.0000	2.4300	57.0000
7.0000	.6660	.0000
8.0000	.2160	15.0000
9.0000	.0600	7.0000
9.0000	3.4680	3.0000
10.0000	.6660	.0000
9.0000	2.0760	3.0000
10.0000	.6660	.0000
11.0000	1.3500	10.0000

137

1.0000	.6540	70.0000
2.0000	.7860	55.0000
4.0000	1.4280	10.0000
4.0000	1.5900	5.0000
5.0000	1.8300	125.0000
7.0000	.4260	.0000
8.0000	.9600	15.0000
9.0000	1.6440	19.0000
10.0000	.4260	.0000
9.0000	.7740	8.1000
10.0000	.4260	.0000
11.0000	1.1400	10.0000

140

1.0000	3.8640	70.0000
2.0000	1.0440	55.0000
3.0000	.5580	70.0000
4.0000	3.3180	10.0000
5.0000	3.5340	125.0000
7.0000	1.5480	.0000
8.0000	.3720	15.0000
9.0000	8.0820	19.0000
10.0000	1.5480	.0000
9.0000	21.6540	8.1000
10.0000	1.5480	.0000
11.0000	3.6420	10.0000

141

1.0000	1.5840	70.0000
2.0000	1.0740	55.0000
3.0000	.9180	70.0000
4.0000	4.0020	10.0000
6.0000	9.8040	125.0000
7.0000	1.8060	.0000
8.0000	.4200	15.0000

9.0000	13.5780	19.0000
10.0000	1.8060	.0000
9.0000	16.0020	8.1000
10.0000	1.8060	.0000
11.0000	5.3700	10.0000

142

9.0000	23.7540	14.1000
10.0000	3.4560	.0000
11.0000	2.2800	10.0000

143

2.0000	1.0600	35.0000
4.0000	1.0090	10.0000
6.0000	2.9670	57.0000
7.0000	2.4700	.0000
9.0000	27.3000	7.0000
10.0000	2.4700	.0000
11.0000	3.3000	10.0000

144

2.0000	.8580	45.0000
4.0000	1.7820	10.0000
5.0000	7.6320	72.0000
7.0000	2.1960	.0000
8.0000	.6300	15.0000
9.0000	9.0360	14.1000
10.0000	2.2020	.0000
9.0000	6.1680	6.1000
10.0000	2.1960	.0000
11.0000	3.0240	10.0000

145

1.0000	.9720	40.0000
2.0000	2.7600	35.0000
4.0000	.5460	10.0000
5.0000	2.3460	57.0000
7.0000	.6960	.0000
8.0000	.7860	15.0000
9.0000	7.2960	7.0000
10.0000	.6960	.0000
11.0000	9.0960	10.0000

146

4.0000	.4740	10.0000
1.0000	.9600	70.0000
2.0000	1.9800	55.0000
5.0000	14.7360	125.0000
7.0000	1.7100	.0000
8.0000	.4080	15.0000
9.0000	11.4720	19.0000
10.0000	1.7100	.0000
11.0000	9.0960	10.0000

147

1.0000	.8520	40.0000
4.0000	.5880	10.0000
4.0000	.7860	5.0000
7.0000	.3960	.0000
8.0000	.5760	15.0000
9.0000	2.0760	7.0000
10.0000	.3960	.0000
11.0000	.6360	10.0000

148

2.0000	.9960	45.0000
4.0000	.4740	10.0000
5.0000	6.2460	72.0000
7.0000	1.1400	.0000
8.0000	.3300	15.0000
9.0000	5.5620	14.1000
10.0000	1.1400	.0000
11.0000	2.2800	10.0000

149

4.0000	2.0280	10.0000
1.0000	1.4460	40.0000
2.0000	.6900	35.0000
5.0000	2.8800	57.0000
7.0000	.8340	.0000
8.0000	.2340	15.0000
9.0000	1.7700	7.0000
9.0000	4.8660	3.0000
11.0000	2.2800	10.0000
10.0000	.8340	.0000
11.0000	23.1960	10.0000

150

2.0000	.9060	45.0000
4.0000	2.5440	10.0000
5.0000	5.4720	72.0000
7.0000	1.6980	.0000
8.0000	.6300	15.0000
9.0000	4.1700	14.1000
9.0000	6.3840	6.1000
10.0000	1.6980	.0000
11.0000	2.5080	10.0000

151

4.0000	2.7060	10.0000
4.0000	2.7900	5.0000
4.0000	3.4740	5.0000
4.0000	5.5800	5.0000
7.0000	3.6900	.0000
8.0000	.2160	15.0000
9.0000	6.8640	19.0000
10.0000	3.6900	.0000
11.0000	15.0000	10.0000

152

2.0000	1.2240	45.0000
4.0000	2.5740	10.0000
4.0000	4.9200	5.0000
4.0000	3.4020	5.0000
7.0000	1.8840	.0000
8.0000	.2940	15.0000
9.0000	5.1240	14.1000
10.0000	1.8840	.0000
11.0000	2.2800	10.0000

153

9.0000	5.0160	7.0000
10.0000	.7500	.0000

154

4.0000	2.5440	10.0000
2.0000	.8280	45.0000
4.0000	2.1360	10.0000
4.0000	2.3040	5.0000
5.0000	2.5500	72.0000
7.0000	1.0680	.0000
8.0000	.2160	15.0000
9.0000	2.5140	14.1000
10.0000	1.0680	.0000
11.0000	2.2800	10.0000

155

4.0000	1.2240	10.0000
4.0000	1.4580	5.0000
5.0000	3.9120	57.0000
7.0000	1.0500	.0000
8.0000	.2940	15.0000
9.0000	4.3800	7.0000
10.0000	1.0500	.0000
11.0000	2.2800	10.0000

156

9.0000	8.3460	7.0000
10.0000	1.2480	.0000
9.0000	10.0380	3.0000
10.0000	1.5060	.0000

157

4.0000	1.4580	10.0000
4.0000	4.0680	5.0000
2.0000	.8280	35.0000
4.0000	1.5000	10.0000
7.0000	.8700	.0000
8.0000	.2340	15.0000
9.0000	.4800	7.0000
9.0000	1.6440	3.0000
10.0000	.8700	.0000
11.0000	1.1580	10.0000

158

4.0000	5.3160	10.0000
7.0000	1.5660	.0000
9.0000	15.6360	19.0000
7.0000	1.5660	.0000
11.0000	43.2000	10.0000

159

4.0000	4.0020	10.0000
1.0000	1.1700	40.0000
2.0000	.9060	35.0000
5.0000	3.1440	57.0000
7.0000	1.3020	.0000
8.0000	.3120	15.0000
9.0000	4.6920	7.0000
9.0000	8.2020	3.0000
10.0000	1.3020	.0000
11.0000	2.2800	10.0000

160

4.0000	1.7580	10.0000
4.0000	1.9680	5.0000
5.0000	5.0820	72.0000
7.0000	1.0740	.0000
8.0000	.8460	15.0000
9.0000	.1860	14.1000
9.0000	6.6900	6.1000
11.0000	3.2880	10.0000

161

1.0000	1.3020	40.0000
2.0000	.6660	35.0000
4.0000	.7980	10.0000
5.0000	1.8780	57.0000
7.0000	1.0260	.0000
8.0000	.2100	15.0000
9.0000	5.9100	7.0000
9.0000	5.9340	3.0000
10.0000	1.0260	.0000
11.0000	2.0520	10.0000

162

1.0000	1.4680	50.0000
2.0000	.9090	45.0000
4.0000	2.0870	10.0000
4.0000	1.9660	5.0000
5.0000	4.1580	72.0000
7.0000	.9000	.0000
8.0000	.2830	15.0000
9.0000	5.9970	14.1000
10.0000	.9000	.0000
9.0000	5.8220	6.1000
10.0000	.9000	.0000
9.0000	1.4110	6.1000
10.0000	.9000	.0000
11.0000	3.0200	10.0000

163

4.0000	.7740	10.0000
4.0000	1.9620	5.0000
5.0000	3.5580	57.0000
7.0000	1.0200	.0000
8.0000	.1980	15.0000
9.0000	2.7780	7.0000
10.0000	1.0200	.0000
9.0000	2.0700	3.0000
11.0000	1.3440	10.0000

164

4.0000	1.1580	10.0000
4.0000	19.6200	5.0000
4.0000	1.3860	5.0000
4.0000	.6060	5.0000
7.0000	.7380	.0000
8.0000	.1860	15.0000
9.0000	1.7280	7.0000
10.0000	.7380	.0000
9.0000	2.2380	3.0000
10.0000	.7380	.0000
11.0000	.9300	10.0000

165

1.0000	6.7020	70.0000
2.0000	1.4940	55.0000
3.0000	1.0320	70.0000
4.0000	3.3900	10.0000
4.0000	5.7060	5.0000
5.0000	10.5300	125.0000
7.0000	.9780	.0000
8.0000	.5940	15.0000
9.0000	16.8600	19.0000
10.0000	.9780	.0000
9.0000	11.5080	8.1000
9.0000	36.0120	8.1000
10.0000	.9780	.0000
11.0000	7.8300	10.0000

166

1.0000	3.1260	50.0000
2.0000	.7860	45.0000
3.0000	.3900	50.0000
4.0000	1.1040	10.0000
5.0000	3.8520	72.0000
7.0000	1.0740	.0000
8.0000	.2940	15.0000
9.0000	2.4480	14.1000
10.0000	1.0740	.0000
9.0000	11.2140	6.1000
10.0000	1.0740	.0000
11.0000	2.8620	10.0000

167

4.0000	1.1400	10.0000
4.0000	12.5220	5.0000
10.0000	2.2920	.0000
11.0000	1.5240	10.0000

168

4.0000	5.6900	10.0000
5.0000	1.0900	72.0000
5.0000	.6670	72.0000
4.0000	3.5230	10.0000
7.0000	3.0400	.0000
8.0000	.6800	15.0000
4.0000	0.7130	10.0000
9.0000	17.3100	14.1000
10.0000	3.0400	.0000
9.0000	2.7800	6.1000
10.0000	3.0400	.0000
11.0000	4.2700	10.0000

169

4.0000	3.5340	10.0000
4.0000	.8050	5.0000
8.0000	.1880	15.0000
9.0000	6.8670	14.1000
10.0000	2.4800	.0000
9.0000	20.0340	6.1000
10.0000	2.4800	.0000
11.0000	3.3000	10.0000

170

4.0000	37.2440	10.0000
7.0000	5.6610	.0000
11.0000	2.5000	10.0000

171

2.0000	.7120	35.0000
4.0000	1.1590	10.0000
4.0000	.7960	5.0000
7.0000	.8200	.0000
9.0000	8.2730	7.0000
7.0000	.8200	.0000
11.0000	1.0900	10.0000

172

1.0000	1.1460	70.0000
7.0000	.6180	.0000
2.0000	.9420	55.0000
4.0000	2.5440	10.0000
4.0000	1.9680	5.0000
5.0000	2.6280	125.0000
7.0000	.6180	.0000
8.0000	.2520	15.0000
9.0000	11.3220	19.0000
10.0000	.6180	.0000

9.0000	5.1300	8.1000
10.0000	.6180	.0000
9.0000	12.8280	8.1000
10.0000	.6180	.0000
11.0000	4.2000	10.0000

173

4.0000	6.0900	10.0000
4.0000	6.9360	5.0000
7.0000	5.8320	.0000
8.0000	.6600	15.0000
9.0000	46.1400	19.0000
10.0000	5.8320	.0000
9.0000	14.9580	8.1000
10.0000	5.8320	.0000
11.0000	7.7700	10.0000

174

1.0000	1.7280	50.0000
2.0000	.5340	45.0000
3.0000	.4860	50.0000
4.0000	4.5900	10.0000
4.0000	.7080	5.0000
4.0000	3.5400	5.0000
8.0000	.2460	15.0000
9.0000	1.1100	14.1000
9.0000	10.8360	6.1000
10.0000	1.1700	.0000
11.0000	2.1060	10.0000

175

1.0000	2.9520	50.0000
2.0000	.4560	45.0000
4.0000	1.6740	10.0000
4.0000	3.5400	5.0000
5.0000	1.7100	72.0000
7.0000	1.4760	.0000
8.0000	.2640	15.0000
9.0000	12.5160	14.1000
10.0000	1.4760	.0000
9.0000	12.3480	6.1000
10.0000	1.8360	.0000

176

4.0000	2.9580	10.0000
1.0000	1.7220	70.0000
2.0000	.9120	55.0000
3.0000	.6060	70.0000
4.0000	2.4000	10.0000
5.0000	3.1500	125.0000
7.0000	.7800	.0000
8.0000	.7080	15.0000
9.0000	1.8060	19.0000
10.0000	.7800	.0000
9.0000	10.4280	8.1000

10.0000	.7800	.0000
11.0000	2.8140	10.0000

177

1.0000	3.1560	70.0000
2.0000	1.1160	55.0000
3.0000	.7500	70.0000
4.0000	2.1360	10.0000
4.0000	1.9620	5.0000
4.0000	1.7160	5.0000
6.0000	6.4020	125.0000
7.0000	2.4660	.0000
8.0000	.4080	15.0000
9.0000	13.9980	19.0000
10.0000	2.4660	.0000
9.0000	28.3860	8.1000
10.0000	2.4660	.0000
11.0000	6.5880	10.0000

178

4.0000	2.8500	10.0000
1.0000	2.3520	70.0000
2.0000	.9360	55.0000
3.0000	.5820	70.0000
4.0000	.6900	10.0000
4.0000	3.5400	5.0000
4.0000	.4740	5.0000
7.0000	1.6680	.0000
8.0000	.3120	15.0000
5.0000	7.0020	125.0000
9.0000	11.5500	19.0000
10.0000	1.6680	.0000
9.0000	13.9140	8.1000
10.0000	1.6680	.0000
11.0000	4.6140	10.0000

179

1.0000	2.6880	70.0000
2.0000	1.2360	55.0000
3.0000	.7380	70.0000
4.0000	4.0980	10.0000
4.0000	3.5400	5.0000
5.0000	3.2880	125.0000
7.0000	1.5300	.0000
8.0000	.4080	15.0000
9.0000	19.7040	19.0000
10.0000	1.5300	.0000
9.0000	19.9140	8.1000
10.0000	1.5300	.0000
11.0000	2.2800	10.0000

180

1.0000	1.3140	70.0000
2.0000	.9420	55.0000
3.0000	.4500	70.0000

4.0000	3.4500	10.0000
4.0000	6.9000	5.0000
4.0000	.3840	5.0000
8.0000	.2700	15.0000
9.0000	8.6100	19.0000
10.0000	1.9980	.0000
9.0000	15.9840	8.1000
10.0000	1.9980	.0000
11.0000	4.0020	10.0000

182

4.0000	32.1360	10.0000
7.0000	4.8180	.0000
4.0000	31.5000	10.0000
7.0000	4.7220	.0000
4.0000	15.7980	5.0000
7.0000	2.3700	.0000
9.0000	9.5280	19.0000
7.0000	1.4280	.0000
11.0000	19.9800	10.0000

183

4.0000	9.6780	10.0000
1.0000	1.4580	70.0000
2.0000	.8760	55.0000
4.0000	1.6740	10.0000
5.0000	3.9180	125.0000
7.0000	1.5060	.0000
8.0000	.2640	15.0000
9.0000	6.8640	19.0000
10.0000	1.5060	.0000
11.0000	2.2800	10.0000

184

4.0000	2.5460	10.0000
4.0000	3.8400	5.0000
2.0000	1.7850	55.0000
5.0000	10.3140	125.0000
7.0000	1.8100	.0000
8.0000	.3930	15.0000
9.0000	7.7370	19.0000
10.0000	1.8100	.0000
9.0000	2.4370	8.1000
10.0000	1.8100	.0000
11.0000	3.6300	10.0000

185

1.0000	2.6580	70.0000
4.0000	3.9360	10.0000
4.0000	7.7160	5.0000
4.0000	3.2220	5.0000
5.0000	8.7300	125.0000
7.0000	3.8880	.0000
8.0000	.4860	15.0000
9.0000	13.2180	19.0000

10.0000	3.8880	.0000

186

1.0000	6.7020	70.0000
2.0000	1.4940	55.0000
3.0000	1.0320	70.0000
4.0000	3.3900	10.0000
4.0000	5.7060	5.0000
5.0000	10.4520	125.0000
7.0000	3.6360	.0000
8.0000	.5940	15.0000
9.0000	17.1900	19.0000
10.0000	3.6360	.0000
9.0000	10.3260	8.1000
9.0000	36.0120	8.1000
10.0000	3.6360	.0000
11.0000	9.6960	10.0000

187

1.0000	2.8740	50.0000
2.0000	.7080	45.0000
4.0000	1.1280	10.0000
5.0000	4.5240	72.0000
7.0000	1.0200	.0000
8.0000	.2640	15.0000
9.0000	7.1880	14.1000
10.0000	1.0200	.0000
9.0000	9.4740	6.1000
10.0000	1.0200	.0000
11.0000	2.7240	10.0000

188

4.0000	7.1140	10.0000
7.0000	1.1200	.0000
11.0000	.7500	10.0000

190

4.0000	2.8740	10.0000
5.0000	10.0860	125.0000
7.0000	2.1360	.0000
8.0000	.2820	15.0000
9.0000	6.7200	19.0000
10.0000	2.1360	.0000
9.0000	17.8500	8.1000
10.0000	2.1360	.0000
9.0000	13.8240	8.1000
10.0000	2.1360	.0000
11.0000	5.7000	10.0000

191

4.0000	2.1360	10.0000
5.0000	13.8000	125.0000
7.0000	2.4720	.0000
8.0000	.3420	15.0000
9.0000	14.7300	19.0000

10.0000	2.4720	.0000
9.0000	14.6940	8.1000
10.0000	2.4720	.0000
11.0000	5.2200	10.0000
192		
9.0000	3.1200	7.0000
10.0000	.3900	.0000
11.0000	10.1700	10.0000
193		
1.0000	.9780	40.0000
5.0000	3.7440	57.0000
7.0000	.7020	.0000
8.0000	.6000	15.0000
9.0000	5.9100	7.0000
10.0000	.7020	.0000
11.0000	2.2800	10.0000
194		
4.0000	1.4280	10.0000
4.0000	1.9620	5.0000
4.0000	1.1340	5.0000
7.0000	1.3440	.0000
8.0000	.5760	15.0000
9.0000	2.0760	7.0000
9.0000	8.8260	3.0000
10.0000	1.3440	.0000
11.0000	.9300	10.0000
195		
4.0000	1.8660	10.0000
3.0000	.4320	50.0000
4.0000	29.5200	10.0000
5.0000	3.6840	72.0000
7.0000	4.1760	.0000
7.0000	3.5580	.0000
8.0000	.2640	15.0000
9.0000	39.9300	14.1000
9.0000	1.3260	6.1000
9.0000	1.4520	6.1000
10.0000	4.1760	.0000
11.0000	13.1700	10.0000
196		
9.0000	22.0680	7.0000
10.0000	2.7300	.0000
9.0000	3.1260	3.0000
10.0000	2.7300	.0000
11.0000	3.0120	10.0000
197		
4.0000	2.0280	10.0000
4.0000	4.9860	5.0000
7.0000	1.2180	.0000

8.0000	.2820	15.0000
9.0000	6.6780	7.0000
10.0000	1.2180	.0000
9.0000	7.6500	3.0000
10.0000	1.2180	.0000
11.0000	2.4360	10.0000
198		
1.0000	1.4160	70.0000
2.0000	.9240	55.0000
3.0000	.5460	70.0000
4.0000	3.1980	10.0000
5.0000	10.8480	125.0000
7.0000	2.3400	.0000
8.0000	.3600	15.0000
9.0000	17.4180	19.0000
9.0000	11.0400	8.1000
10.0000	2.3400	.0000
9.0000	14.5800	8.1000
10.0000	2.3400	.0000
11.0000	6.2460	10.0000
199		
4.0000	1.7580	10.0000
4.0000	1.5600	5.0000
5.0000	3.5220	57.0000
7.0000	1.0260	.0000
8.0000	.3900	15.0000
9.0000	2.7780	7.0000
9.0000	4.6980	3.0000
10.0000	1.0260	.0000
11.0000	2.2800	10.0000
200		
4.0000	1.1040	10.0000
4.0000	1.5600	5.0000
8.0000	.1980	15.0000
9.0000	4.2120	7.0000
10.0000	.5760	.0000
9.0000	2.0760	3.0000
10.0000	.5760	.0000
11.0000	1.1760	10.0000
201		
1.0000	3.7680	70.0000
3.0000	.4800	70.0000
4.0000	3.5400	10.0000
4.0000	1.7580	5.0000
6.0000	41.4840	125.0000
7.0000	3.3840	.0000
8.0000	.7380	15.0000
9.0000	31.2300	.0000
10.0000	3.3840	.0000
9.0000	3.0960	19.0000
11.0000	9.0300	10.0000

202

1.0000	2.3220	70.0000
2.0000	1.5720	55.0000
3.0000	.6360	70.0000
4.0000	2.9700	10.0000
4.0000	.7740	5.0000
5.0000	9.0480	125.0000
7.0000	2.0160	.0000
8.0000	.4860	15.0000
9.0000	18.9540	19.0000
10.0000	2.0160	.0000
11.0000	4.0320	10.0000

205

2.0000	.7420	35.0000
4.0000	2.1910	10.0000
4.0000	.6460	5.0000
8.0000	.2030	15.0000
9.0000	1.2110	7.0000
10.0000	1.3100	.0000
9.0000	8.9690	3.0000
10.0000	1.3100	.0000
11.0000	1.7500	10.0000

206

4.0000	1.9280	10.0000
4.0000	3.4850	5.0000
5.0000	2.5450	57.0000
7.0000	.9700	.0000
8.0000	.2030	15.0000
9.0000	1.6460	7.0000
10.0000	.9700	.0000
9.0000	5.6270	3.0000
10.0000	.9700	.0000
11.0000	1.9500	10.0000

207

4.0000	1.0060	10.0000
7.0000	.6500	.0000
4.0000	6.2210	5.0000
7.0000	.6500	.0000
11.0000	.8700	10.0000

208

1.0000	1.0030	40.0000
2.0000	.6520	35.0000
4.0000	1.7870	10.0000
4.0000	1.9660	5.0000
4.0000	.8890	5.0000
7.0000	.6200	.0000
8.0000	.2030	15.0000
9.0000	3.3870	7.0000
10.0000	.6200	.0000
9.0000	4.2340	3.0000

10.0000	.6200	.0000
11.0000	1.6500	10.0000

209

1.0000	1.4090	50.0000
2.0000	.6830	45.0000
4.0000	1.4590	10.0000
4.0000	1.9660	5.0000
5.0000	3.3190	72.0000
7.0000	.9300	.0000
8.0000	.2350	15.0000
9.0000	7.3020	14.1000
10.0000	.9300	.0000
9.0000	5.3380	6.1000
10.0000	.9300	.0000
11.0000	2.4900	10.0000

210

9.0000	.9680	7.0000
4.0000	.7760	10.0000
4.0000	1.4600	5.0000
7.0000	.8100	.0000
8.0000	.1880	15.0000
9.0000	1.2110	3.0000
10.0000	.8100	.0000
9.0000	10.6030	3.0000
10.0000	.8100	.0000
11.0000	1.6200	10.0000

211

1.0000	.1200	50.0000
2.0000	1.0440	45.0000
4.0000	1.7470	10.0000
4.0000	1.9660	5.0000
5.0000	3.3140	72.0000
7.0000	.9600	.0000
8.0000	.2670	15.0000
9.0000	6.4320	14.1000
10.0000	.9600	.0000
9.0000	10.9630	6.1000
10.0000	.9600	.0000
11.0000	3.2000	10.0000
9.0000	2.3650	14.1000
10.0000	.9600	.0000

212

1.0000	1.5540	40.0000
2.0000	.7260	35.0000
4.0000	1.1040	10.0000
4.0000	1.9620	5.0000
5.0000	2.7060	57.0000
7.0000	.8580	.0000
8.0000	.2340	15.0000
9.0000	8.1720	7.0000
10.0000	.8580	.0000

9.0000	3.6600	3.0000
10.0000	.8580	.0000
11.0000	2.3100	10.0000

213

9.0000	5.8740	7.0000
9.0000	.6660	3.0000
10.0000	.9780	.0000
11.0000	2.2800	10.0000

214

9.0000	4.9560	7.0000
9.0000	.6660	3.0000
9.0000	7.7100	3.0000
10.0000	1.9980	.0000
11.0000	2.2800	10.0000

215

9.0000	5.7480	7.0000
9.0000	.6660	3.0000
9.0000	1.6380	3.0000
10.0000	.9600	.0000
11.0000	2.2800	10.0000

216

9.0000	21.0000	7.0000
10.0000	3.0480	.0000

217

4.0000	1.1040	10.0000
4.0000	1.9620	5.0000
4.0000	1.4880	5.0000
7.0000	.7560	.0000
8.0000	.1980	15.0000
9.0000	3.0540	14.1000
10.0000	.7560	.0000
9.0000	1.6440	6.1000
10.0000	.7560	.0000
11.0000	2.2800	10.0000

218

4.0000	1.7400	10.0000
4.0000	4.7820	5.0000
4.0000	.9900	5.0000
7.0000	1.6680	.0000
8.0000	.5100	15.0000
9.0000	2.0700	14.1000
9.0000	9.7380	6.1000
10.0000	1.6680	.0000
9.0000	2.7960	6.1000
10.0000	1.6680	.0000
11.0000	5.7960	10.0000

219

1.0000	2.0940	70.0000

2.0000	.9540	55.0000
4.0000	2.7060	10.0000
6.0000	10.0380	125.0000
7.0000	2.3760	.0000
4.0000	3.5400	10.0000
8.0000	.3600	15.0000
9.0000	2.6100	19.0000
10.0000	2.3760	.0000
9.0000	14.6940	8.1000
10.0000	2.3760	.0000
11.0000	4.7760	10.0000
9.0000	5.9940	8.1000
10.0000	2.3760	.0000

220

4.0000	8.3640	10.0000
4.0000	1.2840	5.0000
1.0000	1.8300	50.0000
2.0000	.8100	45.0000
4.0000	1.9920	10.0000
6.0000	10.5400	72.0000
7.0000	2.4100	.0000
8.0000	.4000	15.0000
9.0000	9.9100	19.0000
10.0000	2.4100	.0000
9.0000	18.0420	8.1000
9.0000	5.3700	8.1000
11.0000	6.4200	10.0000

221

1.0000	4.6440	70.0000
2.0000	.7920	55.0000
2.0000	2.4480	55.0000
3.0000	1.2900	70.0000
4.0000	1.4280	10.0000
4.0000	3.5400	5.0000
5.0000	2.5920	125.0000
7.0000	1.5060	.0000
8.0000	.7560	15.0000
9.0000	20.3520	19.0000
10.0000	1.5060	.0000
9.0000	1.4040	8.1000
10.0000	1.5060	.0000
11.0000	4.0140	10.0000

222

1.0000	2.0740	70.0000
2.0000	1.0900	55.0000
4.0000	.7760	10.0000
4.0000	1.9660	5.0000
5.0000	8.1520	125.0000
7.0000	1.5100	.0000
8.0000	.3620	15.0000
9.0000	16.0030	19.0000
10.0000	1.5100	.0000

9.0000	4.9220	8.1000
10.0000	1.5100	.0000
11.0000	4.0200	10.0000
223		
1.0000	2.9580	70.0000
2.0000	.6970	55.0000
4.0000	.7760	10.0000
4.0000	.6920	5.0000
4.0000	3.5410	5.0000
5.0000	4.6970	125.0000
7.0000	1.0600	.0000
8.0000	.2670	15.0000
9.0000	9.9120	19.0000
10.0000	1.0600	.0000
9.0000	.6410	8.1000
10.0000	1.0600	.0000
11.0000	2.8300	10.0000
224		
2.0000	.7140	35.0000
4.0000	1.1580	10.0000
4.0000	.7980	5.0000
7.0000	.8220	.0000
9.0000	8.2740	7.0000
10.0000	.8220	.0000
11.0000	1.0920	10.0000
225		
1.0000	7.3440	70.0000
2.0000	1.9620	55.0000
4.0000	8.6160	10.0000
4.0000	.7980	5.0000
6.0000	5.7600	125.0000
7.0000	2.1660	.0000
8.0000	.5460	15.0000
9.0000	1.7640	19.0000
10.0000	2.1660	.0000
9.0000	39.4920	8.1000
10.0000	2.1660	.0000
9.0000	5.4900	8.1000
10.0000	2.1660	.0000
11.0000	7.2360	10.0000
226		
1.0000	3.3780	50.0000
2.0000	.8460	45.0000
4.0000	3.5400	10.0000
5.0000	1.6020	72.0000
7.0000	1.2900	.0000
8.0000	.2640	15.0000
9.0000	14.8500	14.1000
10.0000	1.2900	.0000
11.0000	5.1720	10.0000

228

1.0000	3.9540	50.0000
2.0000	1.1640	45.0000
4.0000	.8280	10.0000
5.0000	1.8420	72.0000
7.0000	1.0080	.0000
8.0000	.3120	15.0000
9.0000	16.0020	14.1000
10.0000	1.0080	.0000
9.0000	2.2920	6.1000
10.0000	1.0080	.0000
11.0000	5.3880	10.0000

229

1.0000	3.5940	70.0000
2.0000	1.8600	55.0000
4.0000	1.1040	10.0000
6.0000	7.5660	125.0000
7.0000	1.4700	.0000
8.0000	.4380	15.0000
9.0000	19.0440	19.0000
10.0000	1.4700	.0000
9.0000	2.5980	8.1000
10.0000	1.4700	.0000
11.0000	3.9180	10.0000

230

4.0000	3.1500	10.0000
5.0000	4.6080	57.0000
7.0000	1.3980	.0000
8.0000	.2220	15.0000
9.0000	17.9700	7.0000
10.0000	1.3980	.0000
11.0000	7.3020	10.0000

231

4.0000	1.4340	10.0000
5.0000	1.0740	57.0000
7.0000	.2220	.0000
8.0000	.1860	15.0000
9.0000	11.5680	7.0000
10.0000	.4200	.0000
11.0000	3.2880	10.0000

232

2.0000	2.0580	55.0000
4.0000	2.6340	10.0000
4.0000	2.9700	5.0000
5.0000	14.9040	125.0000
7.0000	2.5080	.0000
8.0000	.4740	15.0000
9.0000	13.3920	19.0000
9.0000	5.4480	8.1000
10.0000	2.5080	.0000
9.0000	1.8300	8.1000

10.0000	2.5080	.0000
11.0000	5.0220	10.0000

233

2.0000	1.4820	55.0000
4.0000	2.1900	10.0000
4.0000	2.2680	5.0000
5.0000	11.1000	125.0000
7.0000	1.2780	.0000
8.0000	.3600	15.0000
9.0000	1.4880	19.0000
10.0000	1.2780	.0000
9.0000	3.9600	8.1000
10.0000	1.2780	.0000
9.0000	8.1720	8.1000
10.0000	1.2780	.0000
11.0000	3.4320	10.0000

239

2.0000	1.9920	45.0000
4.0000	26.2800	10.0000
4.0000	3.4140	5.0000
6.0000	14.1840	72.0000
7.0000	2.2380	.0000
8.0000	.4380	15.0000
9.0000	1.2060	14.1000
9.0000	5.5260	6.1000
10.0000	2.2380	.0000
9.0000	1.1580	6.1000
10.0000	2.2380	.0000
9.0000	12.5220	6.1000
9.0000	8.5980	6.1000
10.0000	2.2380	.0000
11.0000	3.2880	10.0000

240

2.0000	1.4820	45.0000
4.0000	2.1900	10.0000
4.0000	2.2680	5.0000
6.0000	11.1000	72.0000
7.0000	1.2780	.0000
8.0000	.3600	15.0000
9.0000	1.0200	14.1000
9.0000	1.4880	6.1000
10.0000	1.2780	.0000
9.0000	3.9600	6.1000
10.0000	1.2780	.0000
9.0000	8.1720	6.1000
10.0000	1.2780	.0000
11.0000	3.4260	10.0000

241

1.0000	1.1460	50.0000
2.0000	.9420	45.0000
4.0000	2.5440	10.0000

4.0000	1.9680	5.0000
6.0000	2.6280	72.0000
7.0000	.6180	.0000
8.0000	.2520	15.0000
9.0000	1.2060	14.1000
9.0000	11.3220	6.1000
10.0000	.6180	.0000
9.0000	5.1300	6.1000
10.0000	.6180	.0000
9.0000	12.8280	6.1000
10.0000	.6180	.0000
11.0000	4.2000	10.0000

242

1.0000	1.3140	50.0000
2.0000	.9420	45.0000
3.0000	.4500	50.0000
4.0000	3.4500	10.0000
4.0000	6.9000	5.0000
4.0000	.3840	5.0000
8.0000	.2700	15.0000
9.0000	8.6100	14.1000
10.0000	1.9980	.0000
9.0000	15.9840	6.1000
10.0000	1.9980	.0000
11.0000	4.0020	10.0000

243

9.0000	17.4180	7.0000
10.0000	2.6457	.0000

APPENDIX iv ASSEMBLY PICK LIST

Assembly type
Component type
Quantity used

1	
1.0000	1.0000
2.0000	1.0000
2	
3.0000	1.0000
104.0000	1.0000
105.0000	1.0000
107.0000	1.0000
108.0000	1.0000
109.0000	1.0000
110.0000	4.0000
111.0000	4.0000
112.0000	4.0000
113.0000	1.0000
3	
100.0000	1.0000
101.0000	1.0000
102.0000	1.0000
103.0000	1.0000
4	
5.0000	1.0000
5	
6.0000	1.0000
116.0000	1.0000
117.0000	1.0000
118.0000	1.0000
119.0000	1.0000
120.0000	1.0000
121.0000	1.0000
122.0000	1.0000
123.0000	1.0000
124.0000	1.0000
125.0000	1.0000
126.0000	1.0000
127.0000	1.0000
128.0000	1.0000
131.0000	1.0000
230.0000	1.0000
231.0000	1.0000
6	
7.0000	1.0000
7	
8.0000	1.0000
129.0000	1.0000
8	

114.0000	1.0000
115.0000	1.0000
9	
38.0000	1.0000
80.0000	1.0000
10	
132.0000	1.0000
133.0000	1.0000
239.0000	1.0000
136.0000	1.0000
11	
157.0000	1.0000
12	
13.0000	1.0000
13	
14.0000	1.0000
15.0000	1.0000
14	
137.0000	1.0000
100.0000	1.0000
15	
140.0000	1.0000
141.0000	1.0000
16	
136.0000	1.0000
142.0000	1.0000
233.0000	2.0000
135.0000	1.0000
18	
19.0000	1.0000
23.0000	1.0000
24.0000	1.0000
25.0000	1.0000
158.0000	10.0000
27.0000	1.0000
28.0000	1.0000
9.0000	1.0000
19	
144.0000	2.0000
145.0000	3.0000
146.0000	5.0000
147.0000	3.0000
148.0000	1.0000
149.0000	1.0000
150.0000	1.0000

21.0000	1.0000
22.0000	1.0000
153.0000	1.0000

21

151.0000	1.0000

22

152.0000	1.0000

23

154.0000	1.0000

24

155.0000	1.0000

25

26.0000	1.0000
144.0000	1.0000
145.0000	3.0000
146.0000	5.0000
147.0000	3.0000
148.0000	1.0000

26

11.0000	1.0000

27

10.0000	1.0000
158.0000	1.0000
149.0000	1.0000
150.0000	1.0000

28

144.0000	1.0000
145.0000	1.0000
146.0000	1.0000
147.0000	1.0000
148.0000	1.0000
10.0000	1.0000
158.0000	1.0000
149.0000	1.0000
150.0000	1.0000

29

30.0000	1.0000
31.0000	1.0000
32.0000	1.0000
34.0000	1.0000

30

159.0000	1.0000
40.0000	1.0000

31

40.0000	1.0000
159.0000	1.0000

32

8.0000	1.0000
161.0000	1.0000
162.0000	1.0000
163.0000	1.0000

34

35.0000	1.0000
164.0000	1.0000
161.0000	1.0000
162.0000	1.0000
163.0000	1.0000

35

36.0000	1.0000
75.0000	1.0000

36

8.0000	1.0000

37

165.0000	1.0000
166.0000	1.0000
167.0000	1.0000
99.0000	1.0000

41

130.0000	2.0000
143.0000	2.0000
168.0000	4.0000
173.0000	1.0000
174.0000	1.0000
175.0000	1.0000
176.0000	1.0000
177.0000	5.0000
178.0000	1.0000
179.0000	1.0000
241.0000	1.0000
242.0000	1.0000
78.0000	2.0000
79.0000	1.0000

47

17.0000	1.0000
23.0000	1.0000
24.0000	1.0000
25.0000	1.0000
182.0000	2.0000
48.0000	2.0000
49.0000	1.0000
50.0000	1.0000
52.0000	1.0000

53.0000	1.0000
55.0000	1.0000
54.0000	7.0000
9.0000	1.0000

48

183.0000	1.0000

49

184.0000	1.0000

50

21.0000	1.0000
22.0000	1.0000
153.0000	1.0000
144.0000	1.0000
145.0000	3.0000
146.0000	5.0000
147.0000	3.0000
148.0000	1.0000

52

21.0000	1.0000
22.0000	1.0000
153.0000	1.0000
144.0000	1.0000
145.0000	3.0000
146.0000	5.0000
147.0000	3.0000
148.0000	1.0000

53

54.0000	1.0000
16.0000	1.0000

54

185.0000	1.0000

55

144.0000	1.0000
145.0000	1.0000
146.0000	1.0000
147.0000	1.0000
148.0000	1.0000
54.0000	1.0000
16.0000	1.0000

56

186.0000	1.0000
167.0000	1.0000
187.0000	1.0000
57.0000	1.0000

58

59.0000	1.0000

198.0000	15.0000
199.0000	11.0000
200.0000	17.0000
43.0000	1.0000
85.0000	1.0000
86.0000	1.0000
87.0000	7.0000
88.0000	2.0000
89.0000	10.0000
90.0000	1.0000
91.0000	3.0000

59

190.0000	1.0000
191.0000	1.0000
192.0000	1.0000
193.0000	1.0000
60.0000	1.0000

60

61.0000	1.0000
195.0000	1.0000
63.0000	1.0000
82.0000	1.0000
83.0000	1.0000
84.0000	1.0000

61

62.0000	1.0000
81.0000	1.0000

62

194.0000	2.0000

63

196.0000	1.0000
197.0000	1.0000

64

201.0000	1.0000
202.0000	1.0000

66

208.0000	1.0000
209.0000	1.0000
92.0000	1.0000

67

213.0000	2.0000
214.0000	5.0000
215.0000	4.0000
216.0000	4.0000
217.0000	3.0000
218.0000	3.0000
219.0000	14.0000

220.0000	6.0000
33.0000	6.0000
94.0000	3.0000
96.0000	1.0000

68

221.0000	5.0000
222.0000	1.0000
223.0000	1.0000
69.0000	1.0000

69

224.0000	1.0000

70

221.0000	5.0000
222.0000	1.0000
223.0000	1.0000
97.0000	1.0000

74

228.0000	1.0000
229.0000	1.0000
188.0000	1.0000

92

169.0000	1.0000

95

243.0000	1.0000
95.0000	1.0000

96

170.0000	1.0000

97

171.0000	1.0000

APPENDIX V FACTORY SIMULATOR PROGRAM SOURCE LISTING

```
C Primary Subroutine.

          SUBROUTINE OPTIK

$INCLUDE COM.ICL

          CALL VSPACE (1,30,150)
          CALL FMSPIC
          CALL ASYPIC
          CALL OTHERS
          CALL UPDATE(-1.0,10.0)
          CALL ESPACE(900)
          CALL ALCLFB(20000,20)
          CALL FMSINT
          CALL ASYINT
          CALL MISINT
          CALL STOINT
          CALL FMSDAT
          CALL ASYDAT
          CALL PRTNUM
          CALL EXPARM
          IFLAG = 2
          IBABEL = 1
          CALL SETMOD('V')
          CALL REDRAW
          CALL PCBLST
          CALL SYSLST
          CALL CALDAT
          CALL SFNAME('F:SAVER.DAT')
          IDAY =0
          IPER =0
          IYEAR=87
          CALL COMS
          CALL STPER
          CALL EXEC

          RETURN
          END

C Picture, PCB work center & ASSY work center initialization
C is held in the following include files.

$INCLUDE PINIT.ICL
$INCLUDE FMSINT.ICL
$INCLUDE ASYINT.ICL

C Stores initialisation.

          SUBROUTINE STOINT
$INCLUDE COM.ICL

          REAL LEVEL
          BSTK = 0.0

          DO 10 J = 1,255
```

```
            CALL MAKEPT(IBUFST(J),CFI(J,3),'R',1)
            CALL SRPV(IBUFST(J),1,BSTK)
            CALL ABSROB(SLEV(J),BSTK)
  10    CONTINUE

        DO 20 JJ = 1,2
        CALL MAKEPT(ISTK(JJ),'SK'//CFI(JJ,1),'R',1)
  20    CONTINUE

        CALL SRPV(ISTK(1),1,BSTK*156.0)
        CALL SRPV(ISTK(2),1,BSTK*99.0)
        CALL MAKEPT(ISTOCK,'STK','R',1)
        CALL SRPV(ISTOCK,1,BSTK*255.0)

        RETURN
        END

C Miscellaneous initialisation, including statistics
C and event processors.

        SUBROUTINE MISINT

$INCLUDE COM.ICL

        CALL MAKEPT(NAME,'EXNA','I',1)
        CALL MAKEPT(ITRANS,'TRAN','R',2)
        CALL MAKEPT(ITRBAT,'TRB','R',2)
        CALL MAKEPT(ISAMPL,'SAMP','R',2)
        CALL MAKEPT(ASYLAB,'LABR','I',2)
        CALL SIPV(ASYLAB,1,30)
        CALL SIPV(ASYLAB,2,30)
        CALL MAKEPT(IWIP(1),'BWP','R',1)
        CALL MAKEPT(IWIP(2),'SWP','R',1)
        CALL MAKEPT(NMAC(1),'NMB','I',11)
        CALL SIPV(NMAC(1),1,1)
        CALL SIPV(NMAC(1),2,1)
        CALL SIPV(NMAC(1),3,1)
        CALL SIPV(NMAC(1),4,9)
        CALL SIPV(NMAC(1),5,3)
        CALL SIPV(NMAC(1),6,6)
        CALL SIPV(NMAC(1),7,2)
        CALL SIPV(NMAC(1),8,2)
        CALL SIPV(NMAC(1),9,20)
        CALL SIPV(NMAC(1),10,3)
        CALL SIPV(NMAC(1),11,8)
        CALL MAKEPT(NMAC(2),'NMS','I',2)
        CALL SIPV(NMAC(2),1,8)
        CALL SIPV(NMAC(2),2,2)
        CALL MAKEPT(IPLAN,'PLAN','R',10)
        CALL SRPV(IPLAN,5,1.0)
        CALL MAKEET (ITOPS,'TOPS',0,255)
        CALL MAKEET(IMACH(20,1),'M20H',0,5)
        CALL SETIAT(IMACH(20,1),1,20)
        CALL SETIAT(IMACH(20,1),2,1)
        CALL MAKEPT(ICAL87,'C87','I',365)
```

```
      CALL MAKEPT(ICAL88,'C88','I',366)
      CALL MAKEPT(ICAL89,'C89','I',365)
      CALL MAKEPT(IPER87,'P87','I',52)

      DO 100 II =1,52
              IF(II.EQ.16.OR.II.EQ.17.OR.II.EQ.19
     &         .OR.II.EQ.36)THEN
                      ILL=4
              ELSEIF(II.EQ.22)THEN
                      ILL=3
              ELSEIF(II.EQ.52)THEN
                      ILL=2
              ELSEIF(II.EQ.1)THEN
                      ILL=1
              ELSE
                      ILL=5
              ENDIF

              CALL SIPV(IPER87,II,ILL)

 100  CONTINUE

      CALL MAKEPT(IPER88,'P88','I',51)

      DO 200 II =1,51
      IF(II.EQ.13.OR.II.EQ.14.OR.II.EQ.35)THEN
                      ILL=4
              ELSEIF(II.EQ.22)THEN
                      ILL=3
              ELSE
                      ILL=5
              ENDIF

              CALL SIPV(IPER88,II,ILL)

 200  CONTINUE

      CALL MAKEPT(IPER89,'P89','I',51)

      DO 300 II =1,51
      IF(II.EQ.1.OR.II.EQ.14.OR.II.EQ.15
     &   .OR.II.EQ.18.OR.II.EQ.35)THEN
           ILL=4
      ELSEIF(II.EQ.22)THEN
           ILL=3
      ELSE
           ILL=5
      ENDIF

      CALL SIPV(IPER89,II,ILL)

 300  CONTINUE

C Statistics held in the following include file.
```

```
     $INCLUDE STATS.ICL

             CALL MAKEBE('TRNSPT','ENDPRO','BEGPRO'
           & ,'ENDED','RECORD','ASTRAN','ENASSY'
           & ,'ENDDAY',8)

             CALL MAKECE('BEGSET','BEGASY',2)

             RETURN
             END

C EVENT PROCESSOR ROUTINES.

C Next day, end of experement and record events are held
C in thefollowing include file.

     $INCLUDE NEWDAY.ICL
     $INCLUDE ENDED.ICL
     $INCLUDE RECORD.ICL

C PCB - Transpotr event.

             SUBROUTINE TRNSPT

     $INCLUDE COM.ICL
             WORK = RAT(IPCURE(),7)
             IWORK = INT(WORK)
             CALL ADDRAT(IPCURE(),6,1.0)
             TYPE=RAT(IPCURE(),1)
             OPER=RAT(IPCURE(),6)
             WRKSTN=RPV(IPBDAT(INT(TYPE),INT(OPER)),1)
             CALL SETRAT(IPCURE(),7,WRKSTN)

             IF(OPER.EQ.1.0)THEN
                     CALL SRPV(IWIP(1),1,(RPV(IWIP(1),1)
        &                          + RAT(IPCURE(),4)))

                     RELE=TIME()
           IPOS = LOCAT(IPCURE(),IBAOUT(IWORK))
           CALL SETRAT(MEMBER(IBAOUT(IWORK),IPOS),5,RELE)

             ENDIF

             II=INT(WRKSTN)
             IF(II.EQ.99)THEN
             IPOS = LOCAT(IPCURE(),IBAOUT(IWORK))
             CALL ADD(IPCURE(),IOUT(1),0)
             CALL REMOVE(IBAOUT(IWORK),IPOS)
             INSTOR = ISIZOF(IOUT(1))
             CALL DISIV (112,1,1,INSTOR,3)

             ELSE

             IPOS = LOCAT(IPCURE(),IBAOUT(IWORK))
             CALL ADD(IPCURE(),IBATIN(II),0)
```

```
          CALL REMOVE(IBAOUT(IWORK),IPOS)

          ENDIF

          RETURN
          END

C PCB - Begin setting operation.

          SUBROUTINE BEGSET

      $INCLUDE COM.ICL

          DO 10 II=1,11

              IWORK = IAT(MEMBER(IORDER,II),1)
              CALL SETWLD
              CALL OPTWLD
              IF(ISIZOF(IBATIN(IWORK)).GT.0.AND.
      &          ISIZOF(IWLDMQ(IWORK)).GT.0.AND.
      &                  IAT(IWDSET(JSET),1).GT.
      &                 IAT(IWDSET(JSET),2).AND.
      &                 IAT(IWDOPT(JOPT),1).GT.
      &                 IAT(IWDOPT(JOPT),2))THEN

           CALL SUCCES
           IMASH=IAT(MEMBER(IWLDMQ(IWORK),1),2)

          CALL ADD(MEMBER(IWLDMQ(IWORK),1),
      &            IMCSET(IWORK,IMASH),0)
          CALL REMOVE(IWLDMQ(IWORK),1)
          CALL ADDIAT(IWDSET(JSET),2,1)
          CALL ADDIAT(IWDOPT(JOPT),2,1)
          IAB=ISIZOF(IBATIN(IWORK))

          IF(IAB.GE.5)THEN

          RTYPE=RAT(MEMBER(IMCSET(IWORK,IMASH),0),3)
          ROPER=RAT(MEMBER(IMCSET(IWORK,IMASH),0),4)

          DO 15 ILK=1,IAB

               INM=MEMBER(IBATIN( IWORK),ILK)
          IF(RAT(INM,1).EQ.RTYPE.AND.RAT(INM,6).EQ.
      &   ROPER.AND.ILK.EQ.1)THEN
                     GO TO 16
          ELSEIF(RAT(INM,1).EQ.RTYPE.AND.RAT(INM,6)
      &        .EQ.ROPER.AND.ILK.GT.1)THEN
          CALL SWAP(MEMBER(IBATIN(IWORK),ILK),
      &           IBATIN(IWORK),1,ILK)

                     GO TO 16
          ENDIF
15        CONTINUE
          ENDIF
```

```
16        CALL ADD(MEMBER(IBATIN(IWORK),1),
&         IBASET(IWORK,IMASH),0)
          CALL REMOVE(IBATIN(IWORK),1)

          IF(RAT(MEMBER(IMCSET(IWORK,MASH),1),3).EQ.
&            RAT(MEMBER(IBASET(IWORK,IMASH),1),1)
&            .AND.RAT(MEMBER(IMCSET(IWORK,IMASH)
&            ,1),4).EQ.RAT(MEMBER(IBASET
&               (IWORK,IMASH),1),6))THEN

          SETIME=0.0

          IDIS=IAT(MEMBER(IMCSET(IWORK,IMASH),0),5)
          CALL SETCOL(IDIS,'GG')
          CALL RESET

          ELSE

          TYPE=RAT(MEMBER(IBASET(IWORK,IMASH),1),1)
          OPER=RAT(MEMBER(IBASET(IWORK,IMASH),1),6)
          SETIME=RPV(IPBDAT(INT(TYPE),INT(OPER)),3)
          IDIS=IAT(MEMBER(IMCSET(IWORK,IMASH),0),5)
          CALL SETCOL(IDIS,'YY')
          CALL RESET

          ENDIF

        CALL SETRAT(MEMBER(IMCSET(IWORK,IMASH),1),3,
 &       RAT(MEMBER(IBASET(IWORK,IMASH),1),1))
        CALL SETRAT(MEMBER(IMCSET(IWORK,IMASH),1),4,
 &       RAT(MEMBER(IBASET(IWORK,IMASH),1),6))
        CALL SCHEDL('BEGPRO',SETIME,IMACH(IWORK,IMASH))

        ENDIF

   10     CONTINUE

          DO 20 KK =1,3

          IRNI = INT(SUFM(1.0,11.0,16))
          IRNO = INT(SUFM(1.0,11.0,26))

          CALL SWAP(MEMBER(IORDER,IRNO),IORDER,IRNI,IRNO)

   20     CONTINUE

          RETURN
          END

C PCB - Begin process operation.

          SUBROUTINE BEGPRO

$INCLUDE COM.ICL
```

```
          ID = IPCURE()
          IWORK = IAT(ID,1)
          IMASH = IAT(ID,2)

          IF(IWORK.EQ.7)THEN
                SAMPLE = RPV(ISAMPL,1)
          ELSEIF(IWORK.EQ.10)THEN
                SAMPLE = RPV(ISAMPL,2)
          ELSE
                SAMPLE = 1.0
          ENDIF
          CALL ADD(MEMBER(IMCSET(IWORK,IMASH
     &                  ),1),IMCPRO(IWORK,IMASH),0)
          CALL REMOVE(IMCSET(IWORK,IMASH),1)
          CALL ADD(MEMBER(IBASET(IWORK,IMASH
     &                  ),1),IBAPRO(IWORK,IMASH),0)
          CALL REMOVE(IBASET(IWORK,IMASH),1)
          CALL SETWLD
          CALL ADDIAT(IWDSET(JSET),2,-1)
          TYPE=RAT(MEMBER(IBAPRO(IWORK,IMASH),1),1)
          OPER=RAT(MEMBER(IBAPRO(IWORK,IMASH),1),6)
          EACH=RPV(IPBDAT(INT(TYPE),INT(OPER)),2)
          BSIZE=RAT(MEMBER(IBAPRO(IWORK,IMASH),1),4)
          PROTIM=EACH*BSIZE*SAMPLE
          CALLADDRAT(MEMBER(IBAPRO(IWORK,IMASH)
     &                             ,1),9,PROTIM)
          CALLSCHEDL('ENDPRO',PROTIM,IMACH
     &                  (IWORK,IMASH))
          IDIS=IAT(MEMBER(IMCPRO(IWORK,IMASH),0),5)
          CALL SETCOL(IDIS,'GG')
          CALL RESET

          DO 5 KK =1,5

             IRNI = INT(SUFM(1.0,11.0,16))
             IRNO = INT(SUFM(1.0,11.0,26))

        CALL SWAP(MEMBER(IORDER,IRNO),IORDER,IRNI,IRNO)
 5                 CONTINUE
 15             CONTINUE
 20   CONTINUE

          RETURN
          END

C PCB - End process operation.

          SUBROUTINE ENDPRO

     $INCLUDE COM.ICL

          ID = IPCURE ()
          IWORK=IAT(ID,1)
          IMASH=IAT(ID,2)
```

```
          CALL ADD(MEMBER(IMCPRO(IWORK,IMASH)
     &         ,1),IWLDMQ(IWORK),0)
          CALL REMOVE(IMCPRO(IWORK,IMASH),1)
          CALLADD(MEMBER(IBAPRO(IWORK,IMASH)
     &          ,1),IBAO UT(IWORK),0)
          CALL REMOVE(IBAPRO(IWORK,IMASH),1)

          IF(IPRIOR.EQ.1)THEN
          ITYPE = INT(RAT(MEMBER(IBAOUT(IWORK),0),1))
          LEAD  =  RAT(MEMBER(IBAOUT(IWORK),0),8)
          ICOPS = INT(RAT(MEMBER(IBAOUT(IWORK),0),6))
          FLOWT =  RAT(MEMBER(IBAOUT(IWORK),0),9)
          IROPS = IAT(ITOPS,ITYPE) - ICOPS
          PRNUM = (LEAD - FLOWT)/REAL(IROPS)
          CALL SETRAT(MEMBER(IBAOUT(IWORK),0),10,PRNUM)
          ENDIF

          CALL OPTWLD

          CALL ADDIAT(IWDOPT(JOPT),2,-1)

          CALL SCHEDL('TRNSPT',RPV(ITRANS,1),
     &             (MEMBER(IBAOUT(IWORK),0)))

          IDIS=IAT(MEMBER(IWLDMQ(IWORK),0),5)
          CALL SETCOL(IDIS,'RR')
          CALL RESET

          RETURN
          END

C ASSY - Begin assembly.

          SUBROUTINE BEGASY

     $INCLUDE COM.ICL
          DO 10 II=1,8

          IWORK = IAT(MEMBER(IASORD,II),1)
          IF(IWORK.GE.18)THEN
          JKL = 2
          ELSE
          JKL = 1
          ENDIF
          ILABOR = IPV(ASYLAB,JKL)

          IF(ISIZOF(IASYIN(IWORK)).GT.0.AND.ILABOR.GT.0)TI

          DO 8 KJ = 1,20

          IDENT =MEMBER(IWC(IWORK,1),KJ)

          IF(DSCOF(IDENT).EQ.'    ')GOTO 3
     8    CONTINUE
```

```
      GOTO 10
3     CALL SUCCES

      CALL AIPV(ASYLAB,JKL,-1)
      IMASH=1
      CALL ADD(MEMBER(IWC(IWORK,
&             IMASH),KJ),OCUPYD,1)
      CALL REMOVE(IWC(IWORK,IMASH),KJ)
      CALL ADD(MEMBER(IASYIN(IWORK),1)
&            ,IWC(IWORK,IMASH),KJ)
      CALL REMOVE(IASYIN(IWORK),1)
      TYPE=RAT(MEMBER(IWC(IWORK,IMASH),KJ),1)
      OPER=RAT(MEMBER(IWC(IWORK,IMASH),KJ),6)
      OPTIME=RPV(ISYSOP(INT(TYPE),INT(OPER)),2)
      BSIZE=RAT(MEMBER(IWC(IWORK,IMASH),KJ),4)
      PROTIM = OPTIME*BSIZE

      CALL SCHEDL('ENASSY',PROTIM,
&                 MEMBER(IWC(IWORK,IMASH),KJ))

      ENDIF

   10 CONTINUE

      DO 20 KK =1,3

      IRNI = INT(SUFM(1.0,8.0,16))
      IRNO = INT(SUFM(1.0,8.0,26))

      CALL SWAP(MEMBER(IASORD,IRNO),IASORD,IRNI,IRNO)

   20 CONTINUE

      RETURN
      END

C ASSY - End assembly operation.

      SUBROUTINE ENASSY

$INCLUDE COM.ICL

      ID = IPCURE ()
      IWORK=INT(RAT(ID,7))

      IF(IWORK.GE.18)THEN
              JKL = 2
      ELSE
              JKL = 1
      ENDIF

      CALL AIPV(ASYLAB,JKL,1)
      IPOS = LOCAT(ID,IWC(IWORK,1))
      CALL ADD(ID,IASYOT(IWORK),1)
      CALL REMOVE(IWC(IWORK,1),IPOS)
```

```
        CALL ADD(MEMBER(OCUPYD,1),
     &          IWC(IWORK,1),IPOS)
        CALL REMOVE(OCUPYD,1)

        CALL SCHEDL('ASTRAN',RPV(ITRANS,2),
     &               MEMBER(IASYOT(IWORK),1))

        RETURN
        END

C ASSY - Transport event.

        SUBROUTINE ASTRAN

     $INCLUDE COM.ICL

        ID = IPCURE ()
        IWORK = INT(RAT(ID,7))
        CALL ADDRAT(IPCURE(),6,1.0)
        TYPE=RAT(IPCURE(),1)
        OPER=RAT(IPCURE(),6)
        WRKSTN=RPV(ISYSOP(INT(TYPE),INT(OPER)),1)
        CALL SETRAT(IPCURE(),7,WRKSTN)
        II=INT(WRKSTN)

        IF(II.EQ.99)THEN
                IPOS = LOCAT(ID,IASYOT(IWORK))
                CALL ADD(ID,IOUT(2),0)
                CALL REMOVE(IASYOT(IWORK),IPOS)
        ELSE
                IPOS = LOCAT(ID,IASYOT(IWORK))
                CALL ADD(ID,IASYIN(II),0)
                CALL REMOVE(IASYOT(IWORK),IPOS)
        ENDIF

        RETURN
        END

C determine setter world.

        SUBROUTINE SETWLD

     $INCLUDE COM.ICL

        IF(IWORK.LE.3)THEN
                JSET = 1
        ELSEIF(IWORK.LE.6)THEN
                JSET = 4
        ELSE
                JSET = IWORK
        ENDIF

        RETURN
        END
```

```
C determine operator world.

            SUBROUTINE OPTWLD

      $INCLUDE COM.ICL
            IF(IWORK.LE.3)THEN
                    JOPT = 1
            ELSEIF(IWORK.LE.6)THEN
                    JOPT = 4
            ELSE
                    JOPT = IWORK
            ENDIF

            RETURN
            END

C Reset graphics.

            SUBROUTINE RESET

      $INCLUDE COM.ICL

            IF(IWORK.EQ.1)THEN
                    CALL DIPBX
            ELSEIF(IWORK.EQ.2)THEN
                    CALL AXLBX
            ELSEIF(IWORK.EQ.3)THEN
                    CALL RADBX
            ELSEIF(IWORK.EQ.4)THEN
                    CALL MSKBX
            ELSEIF(IWORK.EQ.5)THEN
                    CALL SAWCC
            ELSEIF(IWORK.EQ.6)THEN
                    CALL SAWOCC
            ELSEIF(IWORK.EQ.7)THEN
                    CALL PI1BX
            ELSEIF(IWORK.EQ.8)THEN
                    CALL FLSOBX
            ELSEIF(IWORK.EQ.9)THEN
                    CALL MANBX
            ELSEIF(IWORK.EQ.10)THEN
                    CALL PI2BX
            ELSEIF(IWORK.EQ.11)THEN
                    CALL ATEBX
            ENDIF

            RETURN
            END

C Run Time interactions.

            SUBROUTINE COMAND

            COMMON /FLAG/IFLAG
            CHARACTER*3 WORD
```

```
     &        OFF: 1=ON, 0=OFF')
                      IF(INUM.EQ.1)THEN
                              IBABEL=1
                              COMMD='COMMS ARE ON'
                              CALL COMPRT(COMMD)
                              CALL   DISTXT(1,-1,0,'Comms
     &                         are ON.')
                      ENDIF
              ELSE
                      CALL  IKEYB(INUM,1,-1,0,'Comms  are
     &                          ON: 1=ON, 0=OFF')
                      IF(INUM.EQ.0)THEN
                              IBABEL=0
                              COMMD='COMMS ARE OFF'
                              CALL COMPRT(COMMD)
                              CALL   DISTXT(1,-1,0,'Comms
     &                                         are OFF.')
                      ENDIF
              ENDIF

              RETURN
              END

C Save model status.

              SUBROUTINE SAVER
     $INCLUDE COM.ICL
              IDAT(1)=IPER
              IDAT(2)=IDAY
              CALL SAVEBK(IDAT,8)
              RETURN
              END

C Restore model status.

              SUBROUTINE RESTR
     $INCLUDE COM.ICL
              CALL RESTBK(IDAT,8)
              IPER=IDAT(1)
              IDAY=IDAT(2)
              RETURN
              END

C Set initial stock levels.

              SUBROUTINE STOSET

     $INCLUDE COM.ICL

              REAL LEVEL
              CALL  RKEYB(BSTK,1,-1,0,'What shall I set stock
     &        levels to ? ')

              DO 10 J = 1,255
                      CALL SRPV(IBUFST(J),1,BSTK)
```

```
                  LEVEL = BFSK
                  CALL ABSROB(SLEV(J),LEVEL)
   10       CONTINUE

            RETURN
            END

C Set number of setters.

            SUBROUTINE SETERS

      $INCLUDE COM.ICL

   10        CALL DISTXT(1,-1,0,'Setter world  defaults
      &      are....')

            CALL DISTXT(1,-1,0,'IWDSET(1)
      &      ='//CFI(IAT(IWDSET(1),1),2)//
      &     'IWDSET(4) ='//CFI(IAT(IWDSET(4),1),2)//
      &     ' IWDSET(10) ='//CFI(IAT(IWDSET(10),1),2))

            CALL  IKEYB(INUM,1,-1,0,'Which  do you wish  to
      &          change 1,4,10  or 0 (quit) ')

            IF(INUM.EQ.1.OR.INUM.EQ.4.OR.INUM.EQ.10)THEN

            CALL IKEYB(II,1,-1,0,'Enter number of Setters')

            IF(II.LT.1)THEN
            GO TO 10
            ELSE
            CALL SETIAT(IWDSET(INUM),1,II)
            CALL DISTXT(1,-1,0,'Setters updated ')
            GO TO 10
            ENDIF

            ENDIF

            RETURN
            END

C Set number of operators.

            SUBROUTINE OPERS

      $INCLUDE COM.ICL

   10       CALL DISTXT(1,-1,0,'Operator world defaults
      &                                    are....')
            CAL TXLSTXT(1,-1,0,' IWDOPT(1) =
      &                 '//CFI(IAT(IWDOPT(1),1),2)//
      &    'IWDOPT(4) = '//CFI(IAT(IWDOPT(4),1),2)//
      &    'IWDOPT(7) = '//CFI(IAT(IWDOPT(7),1),2)//
      &    'IWDOPT(8) = '//CFI(IAT(IWDOPT(8),1),2) )
```

```
          CALL DISTXT(1,-1,0,' IWDOPT(9) =
     &                  '//CFI(IAT(IWDOPT(9),1),2)//
     &    'IWDOPT(10) = '//CFI(IAT(IWDOPT(10),1),2)//
     &    'IWDOPT(11) = '//CFI(IAT(IWDOPT(11),1),2) )

          CALL  IKEYB(INUM,1,-1,0,'Which do you  wish
     &        to change 1,4,7,8,9,10,11 or 0 (quit) ')

          IF(INUM.EQ.1.OR.INUM.EQ.4.OR.INUM.EQ.7
     &               .OR.INUM.EQ.8.OR.INUM.EQ.9
     &               .OR.INUM.EQ.10.OR.INUM.EQ.11)THEN

          CALL IKEYB(II,1,-1,0,'Enter operators ')

          IF(II.LT.1)THEN
          GO TO 10
          ELSE
          CALL SETIAT(IWDOPT(INUM),1,II)
          CALL DISTXT(1,-1,0,'Operators updated ')
          GO TO 10
          ENDIF

          ENDIF

          RETURN
          END

C Set number of machines.

          SUBROUTINE MACHIN

     $INCLUDE COM.ICL

       10 CALL  DISTXT(1,-1,0,'M/C's default  to  maximum
     &                               shown on screen')
          CALL  IKEYB(IWORK,1,-1,0,'Which W/C do  you
     &              wish to change 1 to 11 or 0 (quit)')

          IF(IWORK.GT.0.AND.IWORK.LE.11.)THEN
          IMAX = IAT(IWLDMQ(IWORK),1)
          K = ISIZOF(IWLDMQ(IWORK))
          CALL  IKEYB(II,1,-1,0,'Enter number of machines
     &                                  0 = quit')

          IF(II.EQ.0.OR.II.GT.IMAX)THEN
          GO TO 40
          ELSEIF(II.GT.K)THEN
          KK = II - K

          DO 20 JJ = 1,KK

          CALL MAKEET(IMACH(IWORK,JJ),'MACH ',0,0)
          CALL ADD(IMACH(IWORK,JJ),IWLDMQ(IWORK),0)
          CALL SIPV(NMAC(1),IWORK,
     &         SIPV(NMAC(1),IWORK) + 1))
```

```
       20  CONTINUE

           CALL DISTXT(1,-1,0,'machines updated')

           GO TO 10

            ELSEIF(II.LT.K)THEN
            KK = K - II
            DO 30 JJ = 1,KK
            IDIS=IAT(MEMBER(IWL DMQ(IWORK),1),5)
            CALL SETCOL(IDIS,'KK')
            CALL RESET
            CALL REMOVE(IWLDMQ(IWORK),1)
            CALL SIPV(NMAC(1),IWORK,
     &               (IPV(NMAC(1),IWORK) - 1))

       30   CONTINUE
            CALL DISTXT(1,-1,0,'machines updated')
            GO TO 10
            ENDIF
            ENDIF

       40   RETURN
            END

C  Communication routines and graphics held in the following
C  include files.

     $INCLUDE COMMS.ICL
     $INCLUDE GRAPH.ICL

C Database Initialization and Loading.

C Part number translator.

            SUBROUTINE PRTNUM

     $INCLUDE COM.ICL

            OPEN(4,FILE='TRANSLT.DAT',STATUS='OLD',ERR=85)

       10   READ(4,15,END=100,ERR=90) J,TMSNUM
       15   FORMAT(I5,A20)

            CALL MAKEPT(NCNVRT(J),'T'//CFI(J,3),'T',20)
            CALL STPV(NCNVRT(J),1,TMSNUM)

            IF(J.EQ.255)THEN
                    GO TO 100
            ELSE
                    GO TO 10
            ENDIF

       85   CALL DISTXT(1,-1,0,'Cannot open TRANSLT.DAT
```

```
     &                                          ')
            GO TO 999
         90 CALL DISTXT(1,-1,0,'ERROR reading
     &                            TRANSLT.DAT')
       100 CLOSE(4)
       999 RETURN
            END
```

```
C Julian - working day translator database.

            SUBROUTINE CALDAT

      $INCLUDE COM.ICL

            OPEN(4,FILE='CALEN87.DAT',STATUS='OLD',ERR=600)
       5    READ(4,7,END=610,ERR =595) JJ,II
       7    FORMAT(I3,I4)

            CALL SIPV(ICAL87,JJ,II)
            GO TO 5

       595 CALL DISTXT(1,-1,0,'ERROR reading CALEN87.DAT')
            GO TO 610
       600  CALL DISTXT(1,-1,0,'Cannot open CALEN87.DAT')
       610  CLOSE(4)

            RETURN
            END
```

```
C Experemental parameters.

            SUBROUTINE EXPARM

      $INCLUDE COM.ICL
            REAL VALUE(6)
            CHARACTER*10 EXPER
            NNN = 10
         17 CALL SIPV(NAME,1,NNN)
            EXPER = 'EX'//CFI(NNN,2)
            OPEN(4,FILE=EXPER,STATUS='OLD',ERR=26)
            READ(4,20,END=25)VALUE(1),VALUE(2),VALUE(3),
     &      VALUE(4),VALUE(5),VALUE(6),VALUE(7)
         20 FORMAT(7F9.3)
         25 CLOSE(4)
            GO TO 27
         26 CALL DISTXT(1,-1,0,'cannot open '//EXPER)
            GO TO 99
         27 CALL SRPV(IPLAN,5,(RPV(IPLAN,5)*VALUE(1)))
            CALL SRPV(ITRANS,1,VALUE(2))
            CALL SRPV(ITRANS,2,VALUE(3))
            CALL SRPV(ITRBAT,1,REAL(VALUE(4)))
            CALL SRPV(ITRBAT,2,REAL(VALUE(5)))

            IPRIOR = INT(VALUE(6))
            IDUR = 5
```

```
          ILEN = INT(VALUE(7))
          PER =REAL(IDUR)
          RLN =REAL(ILEN - 1)
          CALL SRPV(IPLAN,1,PER)
          CALL SRPV(IPLAN,2,RLN)
          CALL SCHEDL('ENDED',52800.0,IMACH(20,1))
          CALL SCHEDL('ENDED',64800.0,IMACH(20,1))
          CALL SCHEDL('ENDED',76800.0,IMACH(20,1))
          CALL SCHEDL('ENDED',88800.0,IMACH(20,1))
          CALL SCHEDL('ENDED',
     &            REAL(IDUR*ILEN*480),IMACH(20,1))

          IST = 7
          CALL SRPV(IPLAN,3,REAL(IST*PER*480))
          NUM = 1500
          RLOW  = REAL(IST*IDUR*480)
          RHIGH = REAL(ILEN*IDUR*480)
          CREM = (RHIGH - RLOW) / NUM
          CALL SRPV(IPLAN,4,CREM)
          CALL SCHEDL('RECORD',RLOW,IMACH(20,1))
          RUM = 100.0
          CALL SRPV(ISAMPL,1,(RUM/100))
          RUM = 100.0
          CALL SRPV(ISAMPL,2,(RUM/100))

   99     RETURN
          END

C Load PCB database.

          SUBROUTINE FMSDAT

     $INCLUDE COM.ICL

          REAL VALUE(3)

          OPEN(4,FILE='PCB.DAT',STATUS='OLD',ERR=85)

    5     READ(4,7,END=100,ERR=90) I

    7     FORMAT(I3)

          READ(4,7,END=100,ERR=90) IOPS
          CALL SETIAT (ITOPS,I,IOPS)

          CALL   TRACE('Loading   PCB.DAT;   PCB   No   =
     &                 '//CFI(I,3))

          DO 40 J = 1,IOPS

          CALL MAKEPT(IPBDAT(I,J),'P'//CFI(I,3),'R',3)
          READ(4,15,ERR=90) VALUE(1),VALUE(2),VALUE(3)
   15     FORMAT(F9.3,F9.4,F9.4)
          VALUE(2) = VALUE(2)
```

```
      VALUE(3) = VALUE(3)

      DO 20 K = 1,3

      CALL SRPV(IPBDAT(I,J),K,VALUE(K))
 20   CONTINUE
 40   CONTINUE

      GO TO 5
 85   CALL DISTXT(1,-1,0,'Cannot open PCB.DAT ')
 90   CALL DISTXT(1,-1,0,'ERROR reading PCB.DAT')
100   CLOSE(4)

      RETURN
      END

C Load assembly operation and pick list databases.

      SUBROUTINE ASYDAT

$INCLUDE COM.ICL

  7   FORMAT(I3)
      OPEN(4,FILE='SYSPIC.DAT',STATUS='OLD',ERR=295)
110   READ(4,7,END=310,ERR =300) L
      READ(4,120,ERR=300) VAL1,VAL2
      READ(4,120,ERR=300) VAL3,VAL4
      READ(4,120,ERR=300) VAL5,VAL6
120   FORMAT(2F9.4)

      DO 130 J = 1,3

      CALL MAKEPT(ISYDAT(L,J),'S'//CFI(L,2),'R',2)

130   CONTINUE

      CALL SRPV(ISYDAT(L,1),1,VAL1)
      CALL SRPV(ISYDAT(L,1),2,VAL2)
      CALL SRPV(ISYDAT(L,2),1,VAL3)
      CALL SRPV(ISYDAT(L,2),2,VAL4)
      CALL SRPV(ISYDAT(L,3),1,VAL5)
      CALL SRPV(ISYDAT(L,3),2,VAL6)

      K = (INT(VAL1)+3)

      DO 150 J = 4,K

      READ(4,140,END = 310,ERR=300) VAL7,VAL8
140   FORMAT(2F9.4)

      CALL MAKEPT(ISYDAT(L,J),'S'//CFI(L,2),'R',2)
      CALL SRPV(ISYDAT(L,J),1,VAL7)
      CALL SRPV(ISYDAT(L,J),2,VAL8)

150   CONTINUE
```

```
      GO TO 110

295   CALL DISTXT(1,-1,0,'Cannot open SYSPIC.DAT')
300   CALL DISTXT(1,-1,0,'ERROR reading SYSPIC.DAT')
310   CLOSE(4)

      OPEN(4,FILE='SYS.DAT',STATUS='OLD',ERR=600)
350   READ(4,360,END=610,ERR =600) L,J
360   FORMAT(2I4)


      DO 390 II = 1,J

      READ(4,370,ERR=595) VAL1,VAL2
370   FORMAT(2F9.4)

      CALL MAKEPT(ISYSOP(L,II),'OL'//CFI(L,2),'R',2)
      CALL SRPV(ISYSOP(L,II),1,VAL1)
      CALL SRPV(ISYSOP(L,II),2,VAL2)

390   CONTINUE

      GO TO 350

595   CALL DISTXT(1,-1,0,'ERROR reading SYSOPS.DAT')
      GO TO 610
600   CALL DISTXT(1,-1,0,'Cannot open SYSOPS.DAT')
610   CLOSE(4)

      RETURN
      END

      SUBROUTINE FKEY(NFK)
      RETURN
      END


      FUNCTION IO0104(NAME)
      CHARACTER NAME *(*)
      IF (NAME.EQ.'LST:')THEN
      IO0104 = 2
      ELSE
      IO0104 = 0
      ENDIF
      RETURN
      END
```

Listing of PCB facility initialization FMSINIT.ICL:

```
        SUBROUTINE FMSINT

$INCLUDE COM.ICL

      CALL LSTMAK
      CALL WC01
      CALL WC02
      CALL WC03
      CALL WC04
      CALL WC05
      CALL WC06
      CALL WC07
      CALL WC08
      CALL WC09
      CALL WC10
      CALL WC11
      CALL MAKELS(IBATWD,5,700,'BATW',  0,0,0,0,0,0, 0,0,0)

      DO 16 I = 1,700
      CALL MAKEET(IBAT,'BACH',0,10)
      CALL ADD (IBAT,IBATWD,0)
16    CONTINUE
      RETURN
      END

      SUBROUTINE LSTMAK

$INCLUDE COM.ICL

CALL MAKELS(IBATIN(1), 5,400,'BI1 ', 8 , 4 ,2, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(2), 5,450,'BI2 ', 16, 4, 2, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(3), 5,60,'BI3 ', 20, 4, 2, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(4), 5,100,'BI4 ',116,16, 3, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(5), 5,100,'BI5 ', 36, 1, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(6), 5,50,'BI6 ', 45, 7, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(7), 5,50,'BI7 ', 49, 6, 1, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(8), 5,50,'BI8 ', 54, 1, 2, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(9), 5,100,'BI9 ', 72, 2, 7, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(10),5,100,'BI10', 58, 7, 1, 3, 0,1,1,3,0)
CALL MAKELS(IBATIN(11),5,50,'BI11',102, 3, 3, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(1) ,5,100,'BO1 ',  9, 4, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(2) ,5,100,'BO2 ', 17, 4, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(3) ,5,50,'BO3 ', 21, 4, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(4) ,5,50,'BO4 ', 31,16, 5, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(5) ,5,50,'BO5 ', 37, 1, 6, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(6) ,5,50,'BO6 ', 46, 7,10, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(7) ,5,50,'BO7 ', 50, 6, 3, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(8) ,5,50,'BO8 ', 55,20, 2, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(9) ,5,50,'BO9 ', 87,22, 4, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(10),5,60,'BO10', 59, 7, 3, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(11),5,30,'BO11',103, 3,10, 3, 0,1,1,3,0)
CALL MAKELS(IBAOUT(20),20 ,900,'BI20',  5,3,6,0,1,1, 9,3,0)
CALL MAKELS(IORDER,11,11,'ORDR',0,0,0,0,0,0,0,0,0)
```

```
      DO 15 JJ = 1,11
      CALL MAKEET(IJJJ,'WORK',0,1)
      CALL SETIAT(IJJJ,1,JJ)
      CALL ADD (IJJJ,IORDER,
15    CONTINUE
      CALL MAKELS(IOUT(1),10,150,'OUT1',13,19,3,-3,0,1,8,3,0)

      RETURN
      END

      SUBROUTINE WC01

$INCLUDE COM.ICL
     CALL MAKELS(IBASET(1,1), 1,1,'BS1 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IBAPRO(1,1), 1,1,'BP1 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IWLDMQ(1),1,1,'MW1 ',0,0,0,0,0,0,0,0,1)
     CALL SETIAT(IWLDMQ(1),1,1)
     CALL MAKELS(IMCSET(1,1), 1, 1,'MS1 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IMCPRO(1,1), 1, 1,'MP1 ',0,0,0,0,0,0,0,0,0)
     CALL MAKEET(IMACH(1,1),'MC1 ',0,5)
     CALL SETIAT(IMACH(1,1),1,1)
     CALL SETIAT(IMACH(1,1),2,1)
     CALL SETIAT(IMACH(1,1),5,6)
     CALL ADD(IMACH(1,1),IWLDMQ(1),0)
     CALL MAKEET(IWDSET(1),'SW1 ',0,2)
     CALL SETIAT(IWDSET(1),1,3)
     CALL MAKEET(IWDOPT(1),'OW1 ',0,2)
     CALL SETIAT(IWDOPT(1),1,3)

     RETURN
     END


     SUBROUTINE WC02

$INCLUDE COM.ICL

     CALL MAKELS(IBASET(2,1), 1,1,'BS2 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IBAPRO(2,1), 1,1,'BP2 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IWLDMQ(2),1,1,'MW2 ',0,0,0,0,0,0,0,0,1)
     CALL SETIAT(IWLDMQ(2),1,1)
     CALL MAKELS(IMCSET(2,1), 1, 1,'MS2 ',0,0,0,0,0,0,0,0,0)
     CALL MAKELS(IMCPRO(2,1), 1, 1,'MP2 ',0,0,0,0,0,0,0,0,0)
     CALL MAKEET(IMACH(2,1),'MC2 ',0,5)
     CALL SETIAT(IMACH(2,1),1,2)
     CALL SETIAT(IMACH(2,1),2,1)
     CALL SETIAT(IMACH(2,1),5,14)
     CALL ADD(IMACH(2,1),IWLDMQ(2),0)

     RETURN
     END

     SUBROUTINE WC03
```

```
 $INCLUDE COM.ICL

      CALL MAKELS(IBASET(3,1), 1,1,'BS3 ',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(3,1), 1,1,'BP3 ',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IWLDMQ(3),1,1,'MW3 ',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(3),1,1)
      CALL MAKELS(IMCSET(3,1), 1, 1,'MS3 ',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(3,1), 1, 1,'MP3 ',0,0,0,0,0,0,0,0,0)
      CALL MAKEET(IMACH(3,1),'MC3 ',0,5)
      CALL SETIAT(IMACH(3,1),1,3)
      CALL SETIAT(IMACH(3,1),2,1)
      CALL SETIAT(IMACH(3,1),5,18)
      CALL ADD(IMACH(3,1),IWLDMQ(3),0)

      RETURN
      END

      SUBROUTINE WC04
 $INCLUDE COM.ICL

       DO 10 I = 1,9

  CALL MAKELS(IBASET(4,I), 1,1,'BS4'//CFI(I,1),0,0,0,0,0,0,0,(
  CALL MAKELS(IBAPRO(4,I), 1,1,'BP4'//CFI(I,1),0,0,0,0,0,0,0,(
  CALL MAKELS(IMCSET(4,I), 1,1,'MS4'//CFI(I,1),0,0,0,0,0,0,0,(
  CALL MAKELS(IMCPRO(4,I), 1,1,'MP4'//CFI(I,1),0,0,0,0,0,0,0,(

 10    CONTINUE
       CALL MAKELS(IWLDMQ(4),10,10,'MW4',0,0,0,0,0,0,0,0,1)
       CALL SETIAT(IWLDMQ(4),1,9)
       DO 20 J = 1,9
       CALL MAKEET(IMACH(4,J),'MC4'//CFI(J,1),0,5)
       CALL SETIAT(IMACH(4,J),1,4)
       CALL SETIAT(IMACH(4,J),2,J)
       CALL SETIAT(IMACH(4,J),5,21+J)
       CALL ADD(IMACH(4,J),IWLDMQ(4),0)
 20    CONTINUE

       CALL MAKEET(IWDSET(4),'SW4 ',0,2)
       CALL SETIAT(IWDSET(4),1,3)
       CALL MAKEET(IWDOPT(4),'OW4 ',0,2)
       CALL SETIAT(IWDOPT(4),1,12)

       RETURN
       END

       SUBROUTINE WC05

&INCLUDE COM.ICL
      CALL MAKELS(IWLDMQ(5),3,3,'MW5'1)
      CALL SETIAT(IWLDMQ(5),1,3)

      DO 10 J = 1,3
      CALL MAKELS(IBASET(5,J),1,1,'BS5'//CFI(J,1))
      CALL MAKELS(IBAPRO(5,J),1,1,'BP5'//CFI(J,1))
```

```
      CALL MAKELS(IMCSET(5,J),1, 1,'MS5'//CFI(J,1))
      CALL MAKELS(IMCPRO(5,J),1, 1,'MP5'//CFI(J,1))
      CALL MAKEET(IMACH(5,J),'MC5'//CFI(J,1),0,5)
      CALL SETIAT(IMACH(5,J),1,5)
      CALL SETIAT(IMACH(5,J),2,J)
      CALL SETIAT(IMACH(5,J),5,(31+J))
      CALL ADD(IMACH(5,J),IWLDMQ(5),0)

 10    CONTINUE

      RETURN
      END

       SUBROUTINE WC06

$INCLUDE COM.ICL

      CALL MAKELS(IWLDMQ(6),6,6,'MW6',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(6),1,6)
      DO 10 J = 1,6

 CALL MAKELS(IBASET(6,J), 1,1,'BS6'//CFI(J,1),0,0,0,0,0,0,0,0,
 CALL MAKELS(IBAPRO(6,J), 1,1,'BP6'//CFI(J,1),0,0,0,0,0,0,0,0,
 CALL MAKELS(IMCSET(6,J), 1,1,'MS6'//CFI(J,1),0,0,0,0,0,0,0,0,
 CALL MAKELS(IMCPRO(6,J), 1,1,'MP6'//CFI(J,1),0,0,0,0,0,0,0,0,
 CALL MAKEET(IMACH(6,J),'MC6'//CFI(J,1),0,5)
 CALL SETIAT(IMACH(6,J),1,6)
 CALL SETIAT(IMACH(6,J),2,J)
 CALL SETIAT(IMACH(6,J),5,(37+J))
 CALL ADD(IMACH(6,J),IWLDMQ(6),0)

10    CONTINUE

      RETURN
      END

      SUBROUTINE WC07

$INCLUDE COM.ICL

      CALL MAKELS(IWLDMQ(7),2,2,'MW7 ',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(7),1,2)
      DO 10 J = 1,2

CALL MAKELS(IBASET(7,J),1,1,'BS7'//CFI(J,1),0,0,0,0,0,0,0,0,0
CALL MAKELS(IBAPRO(7,J),1,1,'BP7'//CFI(J,1),0,0,0,0,0,0,0,0,0
CALL MAKELS(IMCSET(7,J),1,1,'MS7'//CFI(J,1),0,0,0,0,0,0,0,0,0
CALL MAKELS(IMCPRO(7,J),1,1,'MP7'//CFI(J,1),0,0,0,0,0,0,0,0,0
CALL MAKEET(IMACH(7,J),'MC7'//CFI(J,1),0,5)
CALL SETIAT(IMACH(7,J),1,7)
CALL SETIAT(IMACH(7,J),2,J)
CALL ADD(IMACH(7,J),IWLDMQ(7),0)

 10    CONTINUE
```

```
      CALL SETIAT(IMACH(7,1),5,47)
      CALL SETIAT(IMACH(7,2),5,11)
      CALL MAKEET(IWDSET(7),'SW7 ',0,2)
      CALL SETIAT(IWDSET(7),1,2)
      CALL MAKEET(IWDOPT(7),'OW7 ',0,2)
      CALL SETIAT(IWDOPT(7),1,2)

      RETURN
      END

      SUBROUTINE WC08

$INCLUDE COM.ICL

      CALL MAKELS(IBASET(8,1), 1,1,'BS81',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(8,1), 1,1,'BP81',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBASET(8,2), 1,1,'BS82',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(8,2), 1,1,'BP82',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IWLDMQ(8),2,2,'MW8',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(8),1,2)
      CALL MAKELS(IMCSET(8,1), 1, 1,'MS81',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(8,1), 1, 1,'MP81',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCSET(8,2), 1, 1,'MS82',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(8,2), 1, 1,'MP82',0,0,0,0,0,0,0,0,0)
      CALL MAKEET(IMACH(8,1),'MC81',0,5)
      CALL SETIAT(IMACH(8,1),1,8)
      CALL SETIAT(IMACH(8,1),2,1)
      CALL SETIAT(IMACH(8,1),5,51)
      CALL ADD(IMACH(8,1),IWLDMQ(8),0)
      CALL MAKEET(IMACH(8,2),'MC82',0,5)
      CALL SETIAT(IMACH(8,2),1,8)
      CALL SETIAT(IMACH(8,2),2,2)
      CALL SETIAT(IMACH(8,2),5,52)
      CALL ADD(IMACH(8,2),IWLDMQ(8),0)
      CALL MAKEET(IWDSET(8),'SW8 ',0,2)
      CALL SETIAT(IWDSET(8),1,2)
      CALL MAKEET(IWDOPT(8),'OW8 ',0,2)
      CALL SETIAT(IWDOPT(8),1,5)

      RETURN
      END

      SUBROUTINE WC09

$INCLUDE COM.ICL
      DO 10 J = 1,20
CALL MAKELS(IBASET(9,J), 1,1,'BS'//CFI(J,2),0,0,0,0,0,0,0,0,0
CALL MAKELS(IBAPRO(9,J), 1,1,'BP'//CFI(J,2),0,0,0,0,0,0,0,0,0
CALL MAKELS(IMCSET(9,J), 1,1,'MS'//CFI(J,2),0,0,0,0,0,0,0,0,0
CALL MAKELS(IMCPRO(9,J), 1,1,'MP'//CFI(J,2),0,0,0,0,0,0,0,0,0
10    CONTINUE

     CALL MAKELS(IWLDMQ(9),20,20,'MW 9',0,0,0,0,0,0,0,0,1)
     CALL SETIAT(IWLDMQ(9),1,20)
     DO 30 J = 1,10
```

```
      CALL MAKEET(IMACH(9,J),'M9'//CFI(J,2),0,5)
      CALL SETIAT(IMACH(9,J),1,9)
      CALL SETIAT(IMACH(9,J),2,J)
      CALL SETIAT(IMACH(9,J),5,(59+J))
      CALL ADD(IMACH(9,J),IWLDMQ(9),0)

 30   CONTINUE

      DO 40 J = 11,20
      CALL MAKEET(IMACH(9,J),'M9'//CFI(J,2),0,5)
      CALL SETIAT(IMACH(9,J),1,9)
      CALL SETIAT(IMACH(9,J),2,J)
      CALL SETIAT(IMACH(9,J),5,(63+J))
      CALL ADD(IMACH(9,J),IWLDMQ(9),0)

 40   CONTINUE

      CALL MAKEET(IWDSET(9),'SW9',0,2)
      CALL SETIAT(IWDSET(9),1,3)
      CALL MAKEET(IWDOPT(9),'OW9',0,2)
      CALL SETIAT(IWDOPT(9),1,20)

      RETURN
      END

      SUBROUTINE WC10

$INCLUDE COM.ICL

      CALL MAKELS(IBASET(10,1), 1,1,'BSA1',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(10,1), 1,1,'BPA1',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBASET(10,2), 1,1,'BSA2',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(10,2), 1,1,'BPA2',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBASET(10,3), 1,1,'BSA3',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IBAPRO(10,3), 1,1,'BPA3',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IWLDMQ(10),3,3,'MWB ',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(10),1,3)
      CALL MAKELS(IMCSET(10,1), 1, 1,'MSA1',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(10,1), 1, 1,'MPA1',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCSET(10,2), 1, 1,'MSA2',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(10,2), 1, 1,'MPA2',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCSET(10,3), 1, 1,'MSA3',0,0,0,0,0,0,0,0,0)
      CALL MAKELS(IMCPRO(10,3), 1, 1,'MPA3',0,0,0,0,0,0,0,0,0)
      CALL MAKEET(IMACH(10,1),'MCA1',0,5)
      CALL SETIAT(IMACH(10,1),1,10)
      CALL SETIAT(IMACH(10,1),2,1)
      CALL SETIAT(IMACH(10,1),5,56)
      CALL ADD(IMACH(10,1),IWLDMQ(10),0)
      CALL MAKEET(IMACH(10,2),'MCA2',0,5)
      CALL SETIAT(IMACH(10,2),1,10)
      CALL SETIAT(IMACH(10,2),2,2)
      CALL SETIAT(IMACH(10,2),5,88)
      CALL ADD(IMACH(10,2),IWLDMQ(10),0)
      CALL MAKEET(IMACH(10,3),'MCA3',0,5)
      CALL SETIAT(IMACH(10,3),1,10)
```

```
      CALL SETIAT(IMACH(10,3),2,3)
      CALL SETIAT(IMACH(10,3),5,89)
      CALL ADD(IMACH(10,3),IWLDMQ(10),0)
      CALL MAKEET(IWDSET(10),'SW10',0,2)
      CALL SETIAT(IWDSET(10),1,3)
      CALL MAKEET(IWDOPT(10),'OW10',0,2)
      CALL SETIAT(IWDOPT(10),1,3)
      RETURN
      END


      SUBROUTINE WC11

$INCLUDE COM.ICL
      CALL MAKELS(IWLDMQ(11),8,8,'MW E',0,0,0,0,0,0,0,0,1)
      CALL SETIAT(IWLDMQ(11),1,8)

      DO 10 J = 1,8
CALL MAKELS(IBASET(11,J),1,1,'BSB'//CFI(J,1),0,0,0,0,0,0,0,0,(
CALL MAKELS(IBAPRO(11,J),1,1,'BPB'//CFI(J,1),0,0,0,0,0,0,0,0,(
CALL MAKELS(IMCSET(11,J),1,1,'MSB'//CFI(J,1),0,0,0,0,0,0,0,0,(
CALL MAKELS(IMCPRO(11,J),1,1,'MPB'//CFI(J,1),0,0,0,0,0,0,0,0,(
CALL MAKEET(IMACH(11,J),'MCB'//CFI(J,1),0,5)
CALL SETIAT(IMACH(11,J),1,11)
CALL SETIAT(IMACH(11,J),2,J)
CALL SETIAT(IMACH(11,J),5,(91+J))
CALL ADD(IMACH(11,J),IWLDMQ(11),0)

10    CONTINUE
      CALL MAKEET(IWDSET(11),'SW11',0,2)
      CALL SETIAT(IWDSET(11),1,8)
      CALL MAKEET(IWDOPT(11),'OW11',0,2)
      CALL SETIAT(IWDOPT(11),1,8)

      RETURN
      END
```

```
      SUBROUTINE SYSLST

$INCLUDE COM.ICL

        ISLIST(1) = IASPLN
        ISLIST(2) = IASYIN(15)
        ISLIST(3) = IASYIN(16)
        ISLIST(4) = IASYIN(17)
        ISLIST(5) = IASYIN(18)
        ISLIST(6) = IASYIN(19)
        ISLIST(7) = IASYIN(20)
        ISLIST(8) = IASYIN(21)
        ISLIST(9) = IASYIN(22)
        ISLIST(10) = IASYOT(15)
        ISLIST(11) = IASYOT(16)
        ISLIST(12) = IASYOT(17)
        ISLIST(13) = IASYOT(18)
        ISLIST(14) = IASYOT(19)
        ISLIST(15) = IASYOT(20)
        ISLIST(16) = IASYOT(21)
        ISLIST(17) = IASYOT(22)
        ISLIST(18) = IWC(15,1)
        ISLIST(19) = IWC(16,1)
        ISLIST(20) = IWC(17,1)
        ISLIST(21) = IWC(18,1)
        ISLIST(22) = IWC(19,1)
        ISLIST(23) = IWC(20,1)
        ISLIST(24) = IWC(21,1)
        ISLIST(25) = IWC(22,1)
        ISLIST(26) = IOUT(2)

        RETURN
        END

        SUBROUTINE PCBLST

$INCLUDE COM.ICL

        IPLIST(1) = IBAOUT(20)
        IPLIST(2) = IBATIN(1)
        IPLIST(3) = IBATIN(2)
        IPLIST(4) = IBATIN(3)
        IPLIST(5) = IBATIN(4)
        IPLIST(6) = IBATIN(5)
        IPLIST(7) = IBATIN(6)
        IPLIST(8) = IBATIN(7)
        IPLIST(9) = IBATIN(8)
        IPLIST(10) = IBATIN(9)
        IPLIST(11) = IBATIN(10)
        IPLIST(12) = IBATIN(11)
        IPLIST(13) = IBAOUT(1)
        IPLIST(14) = IBAOUT(2)
        IPLIST(15) = IBAOUT(3)
        IPLIST(16) = IBAOUT(4)
        IPLIST(17) = IBAOUT(5)
```

```
IPLIST(18) = IBAOUT(6)
IPLIST(19) = IBAOUT(7)
IPLIST(20) = IBAOUT(8)
IPLIST(21) = IBAOUT(9)
IPLIST(22) = IBAOUT(10)
IPLIST(23) = IBAOUT(11)
IPLIST(24) = IBASET(1,1)
IPLIST(25) = IBASET(2,1)
IPLIST(26) = IBASET(3,1)
IPLIST(27) = IBASET(4,1)
IPLIST(28) = IBASET(4,2)
IPLIST(29) = IBASET(4,3)
IPLIST(30) = IBASET(4,4)
IPLIST(31) = IBASET(4,5)
IPLIST(32) = IBASET(4,6)
IPLIST(33) = IBASET(4,7)
IPLIST(34) = IBASET(4,8)
IPLIST(35) = IBASET(4,9)
IPLIST(36) = IBASET(5,1)
IPLIST(37) = IBASET(5,2)
IPLIST(38) = IBASET(5,3)
IPLIST(39) = IBASET(6,1)
IPLIST(40) = IBASET(6,2)
IPLIST(41) = IBASET(6,3)
IPLIST(42) = IBASET(6,4)
IPLIST(43) = IBASET(6,5)
IPLIST(44) = IBASET(6,6)
IPLIST(45) = IBASET(7,1)
IPLIST(46) = IBASET(7,2)
IPLIST(47) = IBASET(8,1)
IPLIST(48) = IBASET(8,2)
IPLIST(49) = IBASET(9,1)
IPLIST(50) = IBASET(9,2)
IPLIST(51) = IBASET(9,3)
IPLIST(52) = IBASET(9,4)
IPLIST(53) = IBASET(9,5)
IPLIST(54) = IBASET(9,6)
IPLIST(55) = IBASET(9,7)
IPLIST(56) = IBASET(9,8)
IPLIST(57) = IBASET(9,9)
IPLIST(58) = IBASET(9,10)
IPLIST(59) = IBASET(9,11)
IPLIST(60) = IBASET(9,12)
IPLIST(61) = IBASET(9,13)
IPLIST(62) = IBASET(9,14)
IPLIST(63) = IBASET(9,15)
IPLIST(64) = IBASET(9,16)
IPLIST(65) = IBASET(9,17)
IPLIST(66) = IBASET(9,18)
IPLIST(67) = IBASET(9,19)
IPLIST(68) = IBASET(9,20)
IPLIST(69) = IBASET(10,1)
IPLIST(70) = IBASET(10,2)
IPLIST(71) = IBASET(10,3)
IPLIST(72) = IBASET(11,1)
```

```
IPLIST(73)  = IBASET(11,2)
IPLIST(74)  = IBASET(11,3)
IPLIST(75)  = IBASET(11,4)
IPLIST(76)  = IBASET(11,5)
IPLIST(77)  = IBASET(11,6)
IPLIST(78)  = IBASET(11,7)
IPLIST(79)  = IBASET(11,8)
IPLIST(80)  = IBAPRO(1,1)
IPLIST(81)  = IBAPRO(2,1)
IPLIST(82)  = IBAPRO(3,1)
IPLIST(83)  = IBAPRO(4,1)
IPLIST(84)  = IBAPRO(4,2)
IPLIST(85)  = IBAPRO(4,3)
IPLIST(86)  = IBAPRO(4,4)
IPLIST(87)  = IBAPRO(4,5)
IPLIST(88)  = IBAPRO(4,6)
IPLIST(89)  = IBAPRO(4,7)
IPLIST(90)  = IBAPRO(4,8)
IPLIST(91)  = IBAPRO(4,9)
IPLIST(92)  = IBAPRO(5,1)
IPLIST(93)  = IBAPRO(5,2)
IPLIST(94)  = IBAPRO(5,3)
IPLIST(95)  = IBAPRO(6,1)
IPLIST(96)  = IBAPRO(6,2)
IPLIST(97)  = IBAPRO(6,3)
IPLIST(98)  = IBAPRO(6,4)
IPLIST(99)  = IBAPRO(6,5)
IPLIST(100)  = IBAPRO(6,6)
IPLIST(101)  = IBAPRO(7,1)
IPLIST(102)  = IBAPRO(7,2)
IPLIST(103)  = IBAPRO(8,1)
IPLIST(104)  = IBAPRO(8,2)
IPLIST(105)  = IBAPRO(9,1)
IPLIST(106)  = IBAPRO(9,2)
IPLIST(107)  = IBAPRO(9,3)
IPLIST(108)  = IBAPRO(9,4)
IPLIST(109)  = IBAPRO(9,5)
IPLIST(110)  = IBAPRO(9,6)
IPLIST(111)  = IBAPRO(9,7)
IPLIST(112)  = IBAPRO(9,8)
IPLIST(113)  = IBAPRO(9,9)
IPLIST(114)  = IBAPRO(9,10)
IPLIST(115)  = IBAPRO(9,11)
IPLIST(116)  = IBAPRO(9,12)
IPLIST(117)  = IBAPRO(9,13)
IPLIST(118)  = IBAPRO(9,14)
IPLIST(119)  = IBAPRO(9,15)
IPLIST(120)  = IBAPRO(9,16)
IPLIST(121)  = IBAPRO(9,17)
IPLIST(122)  = IBAPRO(9,18)
IPLIST(123)  = IBAPRO(9,19)
IPLIST(124)  = IBAPRO(9,20)
IPLIST(125)  = IBAPRO(10,1)
IPLIST(126)  = IBAPRO(10,2)
IPLIST(127)  = IBAPRO(10,3)
```

```
IPLIST(128) = IBAPRO(11,1)
IPLIST(129) = IBAPRO(11,2)
IPLIST(130) = IBAPRO(11,3)
IPLIST(131) = IBAPRO(11,4)
IPLIST(132) = IBAPRO(11,5)
IPLIST(133) = IBAPRO(11,6)
IPLIST(134) = IBAPRO(11,7)
IPLIST(135) = IBAPRO(11,8)
IPLIST(136) = IOUT(1)
RETURN
END
```

Listing of the Assembly model initialization, ASSEMBLY.ICL:

```
           SUBROUTINE ASYINT

      $INCLUDE COM.ICL

           CALL LISTIO
           CALL WC15
           CALL WC16
           CALL WC17
           CALL WC18
           CALL WC19
           CALL WC20
           CALL WC21
           CALL WC22

           RETURN
           END

           SUBROUTINE LISTIO

      $INCLUDE COM.ICL

           CALL MAKELS(IPRODW,5,700,'PROW',0,0,0,0,0,0,0,0,0)

           DO 10 I = 1,700
           CALL MAKEET(IASY,'PROD',0,10)
           CALL ADD (IASY,IPRODW,0)
       10  CONTINUE

      CALL MAKELS(IASPLN,10,600,'APLN',110, 2, 2, 0, 1,1,9,4,1)
      CALL MAKELS(IASYIN(15),3,500,'AI15',118, 8, 2,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(16),3,500,'AI16',118, 8,11,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(17),3,300,'AI17',118, 8,15,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(18),3,500,'AI18',118, 8,21,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(19),3,300,'AI19',118, 8,25,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(20),3,300,'AI20',118, 8,29,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(21),3,300,'AI21',118, 8,33,0,1,1,3,3,0)
      CALL MAKELS(IASYIN(22),3,300,'AI22',118, 8,37,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(15),3,200,'AO15',119,33, 2,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(16),3,200,'AO16',119,33,11,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(17),3,100,'AO17',119,33,15,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(18),3,100,'AO18',119,33,21,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(19),3,100,'AO19',119,33,25,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(20),3,100,'AO20',119,33,29,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(21),3,100,'AO21',119,33,33,0,1,1,3,3,0)
      CALL MAKELS(IASYOT(22),3,100,'AO22',119,33,37,0,1,1,3,3,0)
      CALL MAKELS(IASORD,8,8,'AORD',0,0,0,0,0,0,0,0,0)

           DO 15 JJ = 15,22
           CALL MAKEET(IKKK,'ASWC',0,1)
           CALL SETIAT(IKKK,1,JJ)
           CALL ADD (IKKK,IASORD,0)
      15   CONTINUE
           CALL MAKELS(OCUPYD,10,190,'OCPD',0,0,0,0,0,0,0,0,0)
```

```
      CALL MAKELS(IOUT(2),10,150,'OUT2',123,37,1,0,1,1,20,

      RETURN
      END

      SUBROUTINE WC15

$INCLUDE COM.ICL
      DO 10 I = 1,4
      CALL
      MAKELS(IWC(15,I),20,20,'W15'//CFI(I,1)
&                              ,121,12,(2*I),1,0,1,20,1,0)
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(15,I),0)
 5    CONTINUE
 10   CONTINUE
      RETURN
      END

      SUBROUTINE WC16

$INCLUDE COM.ICL
 CALL MAKELS(IWC(16,1),20,20,'W16 ',121,12,11,1,0,1,20,1,
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(16,1),0)
 5    CONTINUE
      RETURN
      END

      SUBROUTINE WC17

$INCLUDE COM.ICL
 CALL MAKELS(IWC(17,1),20,20,'W17 ',121,12,15,1,0,1,20,1,
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(17,1),0)
 5    CONTINUE

      RETURN
      END

      SUBROUTINE WC18

$INCLUDE COM.ICL

   CALL MAKELS(IWC(18,1),20,20,'W18 ',122,12,21,1,0,1,20,:
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(18,1),0)
 5    CONTINUE

      RETURN
      END
```

```
      SUBROUTINE WC19

 $INCLUDE COM.ICL

CALL MAKELS(IWC(19,1),20,20,'W19 ',122,12,25,1,0,1,20,1,0)
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(19,1),0)
 5    CONTINUE

      RETURN
      END


      SUBROUTINE WC20

 $INCLUDE COM.ICL

 CALL MAKELS(IWC(20,1),20,20,'W20 ',122,12,29,1,0,1,20,1,(
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(20,1),0)
 5    CONTINUE

      RETURN
      END


      SUBROUTINE WC21

 $INCLUDE COM.ICL

 CALL MAKELS(IWC(21,1),20,20,'W211',122,12,33,1,0,1,20,1,(
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(21,1),0)
 5    CONTINUE

      RETURN
      END


      SUBROUTINE WC22

 $INCLUDE COM.ICL

 CALL MAKELS(IWC(22,1),20,20,'W22 ',122,12,37,1,0,1,20,1,(
      DO  5 J = 1,20
      CALL MAKEET(IASMAC,'  ',0,0)
      CALL ADD (IASMAC,IWC(22,1),0)
 5    CONTINUE
      RETURN
      END
```

Full listing of picture initialization, PINIT.ICL

```
      SUBROUTINE FMSPIC

$INCLUDE COM.ICL

      CALL SETPIC (2,2)
      CALL SETWND (17,6,2,1,1,1,64,41)
      CALL SETCOL (2,'GKD??')

      CALL SETPIC (3,3)
      CALL SETPIC (4,3)
      CALL SETPIC (5,3)
      CALL SETWND (10,18,3,1,1,1, 1, 6)
      CALL SETCOL ( 3,'WK???')
      CALL SETCOL ( 4,'CK???')
      CALL SETCOL ( 5,'YK???')

      CALL SETPIC (6,4)
      CALL SETPIC (7,4)
      CALL SETPIC (8,4)
      CALL SETPIC (9,4)
      CALL SETWND (35,4,4,1,1,1,12,7)
      CALL SETCOL (6,'RR')
      CALL SETCOL (7,'WK')
      CALL SETCOL (8,'YK')
      CALL SETCOL (9,'MK')

      CALL SETPIC (14,6)
      CALL SETPIC (15,6)
      CALL SETPIC (16,6)
      CALL SETPIC (17,6)
      CALL SETWND (6,4,6,1,1,1,12,12)
      CALL SETCOL (14,'RR')
      CALL SETCOL (15,'WK')
      CALL SETCOL (16,'YK')
      CALL SETCOL (17,'MK')

      CALL SETPIC (18,7)
      CALL SETPIC (19,7)
      CALL SETPIC (20,7)
      CALL SETPIC (21,7)
      CALL SETWND (6,4,7,1,1,1,10,17)
      CALL SETCOL (18,'RR')
      CALL SETCOL (19,'WK')
      CALL SETCOL (20,'YK')
      CALL SETCOL (21,'MK')

      CALL SETWND (18,7,8,1,1,1, 1,22)
      DO 5 IDIS =22,30
      CALL SETPIC (IDIS,8)
      CALL SETCOL (IDIS,'RR')
5     CONTINUE

      CALL SETPIC (115,8)
```

```
      CALL SETPIC (116,8)
      CALL SETCOL (115,'WK')
      CALL SETCOL (116,'YK')
      CALL SETPIC (31,8)
      CALL SETCOL (31,'MK')

      CALL SETWND (7,7,10,1,1,1, 1,34)
      DO 15 IDIS =32,34
      CALL SETPIC (IDIS,10)
      CALL SETCOL (IDIS,'RR')
15    CONTINUE
      CALL SETPIC (35,10)
      CALL SETPIC (36,10)
      CALL SETPIC (37,10)
      CALL SETCOL (35,'WK')
      CALL SETCOL (36,'YK')
      CALL SETCOL (37,'MK')

      CALL SETWND (9,13,11,1,1,1,9,29)
      DO 20 IDIS =38,43
      CALL SETPIC (IDIS,11)
      CALL SETCOL (IDIS,'RR')
20    CONTINUE
      CALL SETPIC (44,11)
      CALL SETPIC (45,11)
      CALL SETPIC (46,11)
      CALL SETCOL (44,'WK')
      CALL SETCOL (45,'YK')
      CALL SETCOL (46,'MK')

      CALL SETPIC (11,12)
      CALL SETPIC (47,12)
      CALL SETPIC (48,12)
      CALL SETPIC (49,12)
      CALL SETPIC (50,12)
      CALL SETWND (8, 3,12,1,1,1, 1,42)
      CALL SETCOL (11,'RR')
      CALL SETCOL (47,'RR')
      CALL SETCOL (48,'WK')
      CALL SETCOL (49,'YK')
      CALL SETCOL (50,'MK')

      CALL SETWND (22, 3,13,1,1,1,8,42)
      DO 25 IDIS =51,52
      CALL SETPIC (IDIS,13)
      CALL SETCOL (IDIS,'RR')
25    CONTINUE
      CALL SETPIC (53,13)
      CALL SETPIC (54,13)
      CALL SETPIC (55,13)
      CALL SETCOL (53,'WK')
      CALL SETCOL (54,'YK')
      CALL SETCOL (55,'MK')

      CALL SETPIC (88,14)
```

```
      CALL SETPIC (89,14)
      CALL SETPIC (56,14)
      CALL SETPIC (57,14)
      CALL SETPIC (58,14)
      CALL SETPIC (59,14)
      CALL SETWND (10, 3,14,1,1,1,32,42)
      CALL SETCOL (88,'RR')
      CALL SETCOL (89,'RR')
      CALL SETCOL (56,'RR')
      CALL SETCOL (57,'WK')
      CALL SETCOL (58,'YK')
      CALL SETCOL (59,'MK')

      CALL SETWND (32,14,15,1,1,1,17,28)
      DO 30 IDIS =60,69
      CALL SETPIC (IDIS,15)
      CALL SETCOL (IDIS,'RR')
3     CONTINUE

      DO 32 IDIS =74,83
      CALL SETPIC (IDIS,15)
      CALL SETCOL (IDIS,'RR')
35    CONTINUE

      CALL SETPIC (70,15)
      CALL SETPIC (71,15)
      CALL SETPIC (72,15)
      CALL SETPIC (73,15)
      CALL SETPIC (84,15)
      CALL SETPIC (85,15)
      CALL SETPIC (86,15)
      CALL SETPIC (87,15)

      CALL SETCOL (70,'WK')
      CALL SETCOL (71,'CK')
      CALL SETCOL (72,'YK')
      CALL SETCOL (73,'MK')
      CALL SETCOL (84,'WK')
      CALL SETCOL (85,'CK')
      CALL SETCOL (86,'YK')
      CALL SETCOL (87,'MK')

      CALL SETWND (22,14,18,1,1,1,20,13)
      DO 35 IDIS =92,99
      CALL SETPIC (IDIS,18)
      CALL SETCOL (IDIS,'RR???')
35    CONTINUE
      CALL SETPIC (100,18)
      CALL SETPIC (101,18)
      CALL SETPIC (102,18)
      CALL SETPIC (103,18)
      CALL SETCOL (100,'WK')
      CALL SETCOL (101,'CK')
      CALL SETCOL (102,'YK')
      CALL SETCOL (103,'MK')
```

```
      RETURN
      END

      SUBROUTINE ASYPIC

$INCLUDE COM.ICL
       IPIC=24
      CALL SETPIC (110,IPIC)
      CALL SETPIC (118,IPIC)
      CALL SETPIC (119,IPIC)
      CALL SETPIC (120,IPIC)
      CALL SETPIC (121,IPIC)
      CALL SETPIC (122,IPIC)
      CALL SETPIC (123,IPIC)
      CALL SETPIC (124,IPIC)
      CALL SETWND (39,40,IPIC,1,1,1,42,6)
      CALL SETCOL (110,'YK')
      CALL SETCOL (118,'YK')
      CALL SETCOL (119,'YK')
      CALL SETCOL (120,'CK')
      CALL SETCOL (121,'RC')
      CALL SETCOL (122,'CB')
      CALL SETCOL (123,'MW')
      CALL SETCOL (124,'WK')

      RETURN
      END

      SUBROUTINE OTHERS

$INCLUDE COM.ICL
      CALL SETPIC (12,9)
      CALL SETWND (9,3,9,1,1,1,32,6)
      CALL SETPIC (13,17)
      CALL SETWND (22,6,17,1,1,1,20,6)
      CALL SETPIC (105,19)
      CALL SETWND (12,5,19,1,1,1,37,7)
      CALL SETPIC (90,21)
      CALL SETWND (12,5,21,1,1,1,69,7)
      CALL SETPIC (91,22)
      CALL SETWND (16,3,22,1,1,1,42,45)
      CALL SETPIC (117,5)
      CALL SETWND (16,3,5,1,1,1,1,45)

      CALL SETPIC (104,23)
      CALL SETWND (1,43,23,1,1,1,41,6)
      CALL SETCOL (104,'CK')

      CALL SETPIC (106,19)
      CALL SETPIC (107,21)
      CALL SETPIC (112,17)

      CALL SETCOL (112,'WK')
```

```
CALL SETCOL (106,'WK')
CALL SETCOL (107,'WK')
CALL SETCOL (12,'WK')
CALL SETCOL (13,'MW')
CALL SETCOL (105,'CK')
CALL SETCOL (90,'CK')
CALL SETCOL (91,'CKD')
CALL SETCOL (117,'CKD')

RETURN
END
```

Common list include file COM.ICL.

```
 COMMON /BIAS/ IORDER,IJJJ,IASORD,IKKK
 COMMON /ENT1/ IBAT,ITRANS,ITRBAT,IMACH(20,20),ISETER(15),ITOI
 COMMON /ENT2/ IASY
 COMMON /EXNAME/ NAME,IBABEL
 COMMON /FLAG/IFLAG
 COMMON /LIST1/ IBATWD,IBAEND,IBADAT,IBAOUT(20),IBATIN(20),
& IBASET(20,20),IBAPRO(20,20),IWLDMQ(20),IMCSET(20,20),
& IMCPRO(20,20),IMCAVA(20,20),IWDSET(20),IWDOPT(20),ISETT(20,:
 COMMON /LIST2/IPRODW,IASPLN,IOUT(2),IWC(22,4),IASYIN(22),
& IASYOT(22),IASMAC,IASEMQ,OCUPYD,ASYLAB
 COMMON /ORDER/ BORD,SORD
 COMMON /PACKET/ IPBDAT(255,30),ISYDAT(99,30),ISYSOP(99,30),
& NCNVRT(255),IDAY,IPER,IYEAR,ICAL87,ICAL88,ICAL89,
& IPER87,IPER88,IPER89
  COMMON /SAMPLE/ ISAMPL
  COMMON /SETOP/ JSET,JOPT
  COMMON /STAT/ IUTILZ(15),INPROG(2),INQUE(15),INMOVE(2),
& IPLAN,ILATE(2),FLOTM(2),IBASIZ(2),BLED(255),BSIZE(255),
& SLED(100), SSIZE(100),BLAT(255),SLAT(100)
  COMMON /STOR/ IBUFST(255),BLEV(255),SLEV(255),IACESS(255)
  COMMON /WIP/IWIP(2),NMAC(2)
  COMMON /WORK/ IWORK
  COMMON /STOC/ ISTK(2),ISTOCK
  COMMON /PARM/ IPRIOR,IRANG(200),INAME(2),IPLIST(136),ISLIST
  COMMON /SAVR/IDAT(2)


   CHARACTER COMMD*80
   CHARACTER DSCOF*4
   CHARACTER CFI*20
   CHARACTER CFR*20
   CHARACTER ORDER*10,QUANT*13,BIN*8,PERD*2,TPV*20,CHAR*20,
&  TMSNUM*20,PRTNUM*20,T*20
   CHARACTER TOUT*80,DATE*6,PPLAN*80,START*8,FINISH*8
```

Full listing of the next day event include file NEWDAY.ICL

```
        SUBROUTINE ENDDAY

$INCLUDE COM.ICL
        CALL DISTXT(1,-1,0,'End of day '//CFI(IDAY,1)//'.')
        CALL WIPCLS

        IF(IDAY.EQ.IAT(IASPLN,1))THEN
                CALL ENDPER
                CALL STPER
        ELSE
                CALL STTDAY
        ENDIF
        RETURN
        END


        SUBROUTINE STTDAY

$INCLUDE COM.ICL

C.......Increment day counter.

        IDAY=IDAY+1

C.......Launch next assembly batches.

        CALL RUNSYS

        RETURN
        END


        SUBROUTINE ENDPER

$INCLUDE COM.ICL

        CALL DISTXT(1,-1,0,'End of Period '//CFI(IPER,2)//'.'

        IF(IPER.EQ.39)CALL SHUT

C.......1.Write stock update from disc to TMS.

        CALL DISTXT(1,-1,0,'TMS: WIP update.')
        CALL WIPOUT

C.......2.Send MPS to TMS.

        CALL DISTXT(1,-1,0,'TMS: MPS update.')
        CALL NXTMPS

C.......3.Send SALES.(this automatically triggers an MRP run)

        CALL DISTXT(1,-1,0,'TMS: SALES update.')
```

```
      CALL SALES

      RETURN
      END

      SUBROUTINE STPER

$INCLUDE COM.ICL
      CHARACTER YELL*8

C.......Set period counter to the next period and IDAY to 1
      IPER = IPER+1
      IDAY =1
      CALL DRWTIM(TIM())
      CALL DISTXT(1,-1,0,'Start of Period '//CFI(IPER,2)//'

      IF(IYEAR.EQ.87)THEN
              NN = IPV(IPER87,IPER)
      ELSEIF(IYEAR.EQ.88)THEN
              NN = IPV(IPER88,IPER)
      ELSEIF(IYEAR.EQ.89)THEN
              NN = IPV(IPER89,IPER)
      ENDIF

      CALL SETIAT(IASPLN,1,NN)
      DO 10 II = 1,NN
           CALL SCHEDL('ENDDAY',II*480.0,IASPLN)
      10   CONTINUE

C.......Download WIP launches
      CALL WIPDWN
      CALL RUNBAT

C.......Download Suggested actions from TMS.
      CALL MSGDWN
      CALL RSCHED

C.......Update IPLAN,2
      IF(RPV(IPLAN,2).GT.0.0)THEN
         CALL SRPV(IPLAN,2,(RPV(IPLAN,2)-1.0))
      ENDIF

C.......Launch appropriate batches.
      CALL RUNSYS
      CALL DISTXT(1,-1,0,'Running.')

      RETURN
      END


      SUBROUTINE RUNSYS

$INCLUDE COM.ICL

      IF(ISIZOF(IASPLN).EQ.0)GO TO 99
```

```
      IPOS = 1
25    LAUN = INT(RAT(MEMBER(IASPLN,IPOS),5))
      NOW = IFIX(TIM())

      IF(LAUN.LE.NOW)THEN
              I  = INT(RAT(MEMBER(IASPLN,IPOS),1))
              II = INT(RPV(ISYDAT(I,1),1)) + 3
              CALL AIPV(IACESS(I),1,1)
              J = 3
30            J = J + 1
              JJ = INT(RPV(ISYDAT(I,J),1))
              IF(JJ.EQ.999)GO TO 53
              K  = INT(RPV(ISYDAT(I,J),2))
              KK = K*INT(RAT(MEMBER(IASPLN,IPOS),4))
              R  = RPV(ISYDAT(I,J),2)
              RR = R*RAT(MEMBER(IASPLN,IPOS),4)

              IF(RPV(IBUFST(JJ),1).LT.RR)THEN
              CALL AIPV(IACESS(I),2,1)
              CALL SETRAT(MEMBER(IASPLN,IPOS),5,0.0)
      CALL SETDSC(MEMBER(IASPLN,IPOS),'S'//CFI(I,2)//'*')

              IF(IPOS.EQ.ISIZOF(IASPLN))THEN
                         GO TO 99
              ELSE
                         IPOS = IPOS + 1
                         GO TO 25
              ENDIF
              ENDIF

              IF(J.LT.II)GO TO 30
              J = 3
40            J = J + 1
              JJ = INT(RPV(ISYDAT(I,J),1))
              R  = RPV(ISYDAT(I,J),2)
              RR = R*RAT(MEMBER(IASPLN,IPOS),4)
              CALL ARPV(IBUFST(JJ),1,-RR)
              CALL ARPV(ISTK(1),1,-RR)
              CALL ABSROB(SLEV(JJ),RPV(IBUFST(JJ),1))

              IF(J.LT.II)THEN
                 GO TO 40
              ENDIF

53            ITYP = I
              NOP  = 1
              IWORK=INT(RPV(ISYSOP(ITYP,NOP),1))
               WORK=    (RPV(ISYSOP(ITYP,NOP),1))
              RELE=TIME()
              CALL SETRAT(MEMBER(IASPLN,IPOS),5,RELE)
              CALL ADD(MEMBER(IASPLN,IPOS),IASYIN(IWORK),0)
              CALL REMOVE(IASPLN,IPOS)
         CALL ARPV(IWIP(2),1,RAT(MEMBER(IASYIN(IWORK),0),4)
         CALL ADDRAT(MEMBER(IASYIN(IWORK),0),6,1.0)
         CALL SETRAT(MEMBER(IASYIN(IWORK),0),7,WORK)
```

```
          IF(IPOS.GT.ISIZOF(IASPLN).OR.ISIZOF(IASPLN).EQ.0)THEN
          GO TO 99
          ENDIF

          GO TO 25

      ENDIF

 99   CALL DRAW (IASPLN)
      INSTOR = ISIZOF(IASPLN)
      CALL DISIV (108,6,9,INSTOR,3)

      RETURN
      END


      SUBROUTINE WIPCLS

$INCLUDE COM.ICL

      REAL LEVEL

      CHARACTER DAYOUT*10
      CHARACTER ACAB*10

      ISK = ISIZOF(IOUT(1)) + ISIZOF(IOUT(2))

      IF(IPER.LT.10)THEN
              WRITE(ACAB,2)'STK0',IPER,IDAY
  2           FORMAT(A4,I1,I1)
      ELSE
              WRITE(ACAB,4)'STK',IPER,IDAY
  4           FORMAT(A3,I2,I1)
      ENDIF

      READ(ACAB,6) DAYOUT
  6   FORMAT(A6)

      IF(ISK.LT.1)GO TO 150

      OPEN(4,FILE=DAYOUT,STATUS='NEW',ERR=10)
      GO TO 12

 10   CALL DISTXT(1,-1,0,'Cannot open '//DAYOUT)
      GO TO 800
 12   DO 100 K = 1,2

      IUPDAT = ISIZOF(IOUT(K))
      IF(IUPDAT.LT.1)GO TO 100

      DO 50 I = 1,IUPDAT
               J = INT(RAT(MEMBER(IOUT(K),1),1))
              JJ = INT(RAT(MEMBER(IOUT(K),1),4))
              RR =     RAT(MEMBER(IOUT(K),1),4)
              CALL ARPV(IWIP(K),1,-RR)
```

```
            CALL ARPV(ISTK(K),1,RR)
            CALL ARPV(IBUFST(J),1,RR)
            CALL ARPV(ISTOCK,1,RR)
            LEVEL= RPV(IBUFST(J),1)
            CALL ABSROB(SLEV(J),LEVEL)
      WRITE(4,22)'!P!',TPV(NCNVRT(J),1),CFI(JJ,13),'AS',
     & INT(RAT(MEMBER(IOUT(K),1),2)),'    ','FL18A','1'
 22         FORMAT(A3,A20,A13,A2,I6,A4,A6,A8)

            IF(TIM().GT.RPV(IPLAN,3))THEN
               TYPE = REAL(J)
               SIZE = REAL(JJ)
               RELE = RAT(MEMBER(IOUT(K),1),5)/480
               COMP = TIM()/480
               FLOW = COMP - RELE
               DUE  = RAT(MEMBER(IOUT(K),1),8)/480
               TVAR = DUE - COMP
               CALL ABSROB(IFLOTM(K),FLOW)
               CALL ABSROB(IBASIZ(K),SIZE)
               CALL ABSROB(ILATE(K),TVAR)
               CALL ARPV(IPLAN,(6+K),SIZE)

               IF(K.EQ.1)THEN
                     CALL ABSROB(BLED(INT(TYPE)),FLOW)
                     CALL ABSROB(BSIZE(INT(TYPE)),SIZE)
                     CALL ABSROB(BLAT(INT(TYPE)),TVAR)
               ELSE
                     CALL ABSROB(SLED(INT(TYPE)),FLOW)
                     CALL ABSROB(SSIZE(INT(TYPE)),SIZE)
                     CALL ABSROB(SLAT(INT(TYPE)),TVAR)
               ENDIF
        ENDIF

      IF(K.EQ.1)THEN
         INUMB = 10
      ELSE
         INUMB = 10
      ENDIF

            DO 25 IK = 1,INUMB
                  CALL SETRAT(MEMBER(IOUT(K),1),IK,0.0(
 25         CONTINUE

            IF(K.EQ.1)THEN
                  LISNAM = IBATWD
            ELSEIF(K.EQ.2)THEN
                  LISNAM = IPRODW
            ENDIF

            CALL ADD(MEMBER(IOUT(K),1),LISNAM,0)
            CALL REMOVE(IOUT(K),1)

 50   CONTINUE
100   CONTINUE
      GO TO 200
```

```
  150     OPEN(4,FILE=DAYOUT,STATUS='NEW',ERR=10)
          WRITE(4,155) '!P!','NO WIP OUTPUT'
  155     FORMAT(A3,A20)
  200     CLOSE(4)
  800     RETURN
          END

          SUBROUTINE RSCHED

$INCLUDE COM.ICL

          INTEGER SPRTNM(100),SIORD(100),SNWYER(100),
     &  SNWDAY(100),SOLYER(100),
     &  SOLDAY(100),BPRTNM(100),BIORD(100),BNWYER(100),BNWDAY(100)
     &  BOLDAY(100),BOLYER(100)
          CHARACTER TX1*20,TX2*1,TX3*3,SMESGE(100)*1,
     &  BMESGE(100)*1,DESC*4
          CHARACTER ACTFIL*10

          CALL DISTXT(1,-1,0,'Re-scheduling.')

          IB=0
          IS=0

          IF(IPER.LT.10)THEN
                  ACTFIL='ACTN0'//CFI(IPER,1)
          ELSE
                  ACTFIL='ACTN'//CFI(IPER,2)
          ENDIF

          OPEN(4,FILE=ACTFIL,STATUS='OLD',ERR=1)
          GO TO 5
    1     COMMD='Cannot open '//ACTFIL//'.'
          CALL COMPRT(COMMD)
          GO TO 999

    5     READ(4,10,ERR=990,END=15)TX1,IN1,IN2,TX2,
       &  IN3,IN4,I5,I6,IN7,TX3
   10     FORMAT(A20,2X,I10,I6,A1,I2,I3,I2,I3,I12,A3)

          IF(TX2.NE.'C')THEN
                  DO 11 IK = 1,255
                          IF(TX1.EQ.TPV(NCNVRT(IK),1)) GO TO 12
   11             CONTINUE
          IF(IK.GT.255)GO TO 5
   12             IF(IK.LT.100)THEN
                          IS=IS+1
                          SPRTNM(IS)=IK
                     SIORD(IS) =IN1
                     SMESGE(IS)=TX2
                     SNWYER(IS)=IN3
                     SNWDAY(IS)=IN4
                     SOLYER(IS)=IN5
                     SOLDAY(IS)=IN6
             ELSEIF(IK.LT.256)THEN
```

```
                    IB=IB+1
                    BPRTNM(IB)=IK
                    BIORD(IB) =IN1
                    BMESGE(IB)=TX2
                    BNWYER(IB)=IN3
                    BNWDAY(IB)=IN4
                    BOLYER(IB)=IN5
                    BOLDAY(IB)=IN6
            ENDIF
         ENDIF
         GO TO 5
 15      CLOSE(4)
         IE =IB+IS

         DO 40 IKI =1,26
         DESC = DSCOF(ISLIST(IKI))
         IF(ISIZOF(ISLIST(IKI)).EQ.0)GO TO 40
         IDI = ISIZOF(ISLIST(IKI))

         DO 35 IPOS =1, IDI

         IF(DSCOF(MEMBER(ISLIST(IKI),IPOS)).EQ.'     ')GO TO 3!
                 DO 30 KII = 1,IS

         IF(INT(RAT(MEMBER(ISLIST(IKI),IPOS),2)).EQ.SIORD(KII)
&              .AND.SMESGE(KII).EQ.'D')THEN
                    IYEARI = SNWYER(KII)
                    IDAYI  = SNWDAY(KII)
                    DUE    = REQIRD(IYEARI,IDAYI)

                    CALL SETRAT(MEMBER(ISLIST(IKI),IPOS),8,DUI

         ELSEIF(INT(RAT(MEMBER(ISLIST(IKI),IPOS),2)).EQ.SIORD
&       .AND.SMESGE(KII).EQ.'E')THEN
                    IYEARI = SNWYER(KII)
                    IDAYI  = SNWDAY(KII)
                    DUE    = REQIRD(IYEARI,IDAYI)

                    CALL SETRAT(MEMBER(ISLIST(IKI),IPOS),8,DUI
         ENDIF

  30             CONTINUE
  35     CONTINUE
  40     CONTINUE

         DO 240 IKI =1,136
         DESC = DSCOF(IPLIST(IKI))
         IF(ISIZOF(IPLIST(IKI)).EQ.0)GO TO 240
         IDI = ISIZOF(IPLIST(IKI))

         DO 235 IPOS =1,IDI

         DO 230 KII = 1,IB
         IF(INT(RAT(MEMBER(IPLIST(IKI),IPOS),2)).EQ.BIORD(KII)
   &                     BMESGE(KII).EQ.'D')THEN
```

```
                              IYEARI = BNWYER(KII)
                              IDAYI  = BNWDAY(KII)
                              DUE    = REQIRD(IYEARI,IDAYI)

                    CALL SETRAT(MEMBER(IPLIST(IKI),IPOS),8,DUE)

          ELSEIF(INT(RAT(MEMBER(IPLIST(IKI),IPOS),2)).EQ.BIORD
     &        .AND.BMESGE(KII).EQ.'E')THEN
                         IYEARI = BNWYER(KII)
                         IDAYI  = BNWDAY(KII)
                         DUE    = REQIRD(IYEARI,IDAYI)

                         CALL SETRAT(MEMBER(IPLIST(IKI),IPOS),8,DUI
          ENDIF
  230     CONTINUE
  235     CONTINUE
  240     CONTINUE
          GO TO 999
  990     CALL DISTXT(1,-1,0,'Error reading '//ACTFIL)
          GO TO 999
  999     RETURN
          END

          SUBROUTINE RUNBAT

$INCLUDE COM.ICL

          REAL ORDNUM,BQUAN,TYPE
          REAL LAUN,DUE,RELESE,REQIRD
          REAL RINC
          CHARACTER ORDERS*9,DUM*2

          IF(IPER.LT.10)THEN
             CALL DISTXT(1,-1,0,'Opening ORDER0'//CFI(IPER,1))
             ORDERS = 'ORDER0'//CFI(IPER,1)
          ELSE
             CALL DISTXT(1,-1,0,'Opening ORDER'//CFI(IPER,2))
             ORDERS = 'ORDER'//CFI(IPER,2)
          ENDIF

          OPEN(4,FILE=ORDERS,STATUS='OLD',ERR=3)
          GO TO 5

    3     COMMD='Can not open '//ORDERS//'.'
          CALL COMPRT(COMMD)

          GO TO 800

    5     READ(4,7,END=100,ERR=150) DUM,JORDER,PRTNUM,
     &        IQUAN,IYEARO,IDAYO,IYEARI,IDAYI

    7     FORMAT(A2,I10,A20,I8,4X,I2,I3,I2,I3)

          BQUAN=REAL(IQUAN)
          BBAT = 1.0
```

```
      DO 10 IK = 1,255
         IF(PRTNUM.EQ.TPV(NCNVRT(IK),1)) GO TO 15
10    CONTINUE

      COMMD = 'Part number '//PRTNUM//' not recognised !!!'
      OPEN(5,FILE='LST:',STATUS='NEW',ERR=5)
      WRITE(5,11)COMMD
11    FORMAT(A80)
      CLOSE(5)
      GO TO 5

15    IF(IK.LE.99)THEN
              INAME(1) =  IPRODW
              INAME(2) =  IASPLN
      ELSE
              INAME(1) =  IBATWD
              INAME(2) =  IBAOUT(20)
      ENDIF

      IF(IK.LE.99)THEN
         TRB = RPV(ITRBAT,2)
         TTRB = 1.5*RPV(ITRBAT,2)
      ELSE
         TRB = RPV(ITRBAT,1)
         TTRB = 1.5*RPV(ITRBAT,1)
      ENDIF

      LAUN    = RELESE(IYEARO,IDAYO)
      DUE     = REQIRD(IYEARI,IDAYI)
      TYPE    = REAL(IK)
      VAL2    = BQUAN
      ORDNUM  = REAL(JORDER)

 30   IF(VAL2.GT.TTRB)THEN
         CALL SETRAT(MEMBER(INAME(1),1),1,TYPE)
         CALL SETRAT(MEMBER(INAME(1),1),2,ORDNUM)
         CALL SETRAT(MEMBER(INAME(1),1),3,BBAT)
         CALL SETRAT(MEMBER(INAME(1),1),4,TRB)
         CALL SETRAT(MEMBER(INAME(1),1),5,LAUN)
         CALL SETRAT(MEMBER(INAME(1),1),7,20.0)
         CALL SETRAT(MEMBER(INAME(1),1),8,DUE)

         IF(IPRIOR.EQ.1)THEN
  CALL SETRAT(MEMBER(INAME(1),1),10,DUE/IAT(ITOPS,INT(TYPE))
         ENDIF

         IF(TYPE.LT.100.0)THEN
 CALL SETDSC(MEMBER(INAME(1),1),'S'//CFI(INT(TYPE),2))
         ELSE
 CALL SETDSC(MEMBER(INAME(1),1),CFI(INT(TYPE),3))
         ENDIF

         CALL ADD(MEMBER(INAME(1),1),INAME(2),0)
         CALL REMOVE(INAME(1),1)
```

```
          IF(IK.GE.100)THEN
          RINC=RAT(MEMBER(INAME(2),0),5)-TIME()

          IF(RINC.LE.0.0)THEN
                         RINC=0.0
          ENDIF
          CALL SCHEDL('TRNSPT',RINC,MEMBER(INAME(2),0))
          ENDIF
          IF(ISIZOF(INAME(2)).GT.1)THEN
                  ISERC = ISIZOF(INAME(2)) - 1
                  DO 40 IKJ = ISERC,1,-1

          IF(RAT(MEMBER(INAME(2),IKJ),5).GT.
     &    RAT(MEMBER(INAME(2),IKJ+1),5))THEN
          CALL TRACE('IASPLN SWAPPED = '//CFI(IKJ,3))
          CALL SWAP(MEMBER(INAME(2),IKJ+1),
     &    INAME(2),IKJ,IKJ+1)
          ENDIF
40        CONTINUE
          ENDIF

          VAL2 = VAL2-TRB
          BBAT = BBAT + 1.0
          GO TO 30
       ELSE
          CALL SETRAT(MEMBER(INAME(1),1),1,TYPE)
          CALL SETRAT(MEMBER(INAME(1),1),2,ORDNUM)
          CALL SETRAT(MEMBER(INAME(1),1),3,BBAT)
          CALL SETRAT(MEMBER(INAME(1),1),4,VAL2)
          CALL SETRAT(MEMBER(INAME(1),1),5,LAUN)
          CALL SETRAT(MEMBER(INAME(1),1),7,20.0)
          CALL SETRAT(MEMBER(INAME(1),1),8,DUE)

    IF(IPRIOR.EQ.1)THEN
    CALL SETRAT(MEMBER(INAME(1),1),10,DUE/IAT(ITOPS,INT(TYPI
    ENDIF

    IF(TYPE.LT.100.0)THEN
     CALL SETDSC(MEMBER(INAME(1),1),'S'//CFI(INT(TYPE),2))
    ELSE
     CALL SETDSC(MEMBER(INAME(1),1),CFI(INT(TYPE),3))
    ENDIF

          CALL ADD(MEMBER(INAME(1),1),INAME(2),0)
          CALL REMOVE(INAME(1),1)

          IF(IK.GE.100)THEN
                  RINC=RAT(MEMBER(INAME(2),0),5)-TIME()
                  IF(RINC.LE.0.0)THEN
                         RINC=0.0
                  ENDIF

    CALL SCHEDL('TRNSPT',RINC,MEMBER(INAME(2),0))
```

```
      ENDIF

      IF(ISIZOF(INAME(2)).GT.1)THEN
                    ISERC = ISIZOF(INAME(2)) - 1
                    DO 50 IKJ = ISERC,1,-1

      IF(RAT(MEMBER(INAME(2),IKJ),5).GT.
 &    RAT(MEMBER(INAME(2),IKJ+1),5))THEN
      CALL TRACE('IASPLN SWAPPED = '//CFI(IKJ,3))
      CALL SWAP(MEMBER(INAME(2),IKJ+1),
 &    INAME(2),IKJ,IKJ+1)
      ENDIF
  50        CONTINUE
            ENDIF

       ENDIF

C.......Get next WIP order.
        GO TO 5
  100   CLOSE(4)
        GO TO 800
  150   CALL DISTXT(1,-1,0,'ERROR reading '//ORDERS//'.')
  800   RETURN
        END

        FUNCTION RELESE(IYEARO,IDAYO)

$INCLUDE COM.ICL

        IF(IYEAR.EQ.IYEARO.AND.IYEAR.EQ.87)THEN
           RELESE = REAL(480*(IPV(ICAL87,IDAYO)))-480.0
        ELSEIF(IYEAR.EQ.IYEARO.AND.IYEAR.EQ.88)THEN
           RELESE = REAL(480*(IPV(ICAL88,IDAYO)))-480.0
        ELSEIF(IYEAR.EQ.IYEARO.AND.IYEAR.EQ.89)THEN
           RELESE = REAL(480*(IPV(ICAL89,IDAYO)))-480.0
        ELSEIF(IYEAR.LT.IYEARO.AND.IYEAR.EQ.87)THEN
           RELESE = REAL((251+IDAYO)*480) -480.0
        ELSEIF(IYEAR.LT.IYEARO.AND.IYEAR.EQ.88)THEN
           RELESE = REAL((500+IDAYO)*480) -480.0
        ELSEIF(IYEAR.LT.IYEARO.AND.IYEAR.EQ.89)THEN
           RELESE = REAL((748+IDAYO)*480) -480.0
        ELSE
           RELESE=0.1
        ENDIF

        IF(RELESE.LT.0.0.OR.RELESE.LT.TIME())THEN
           RELESE = TIME()
        ENDIF

        RETURN
        END

        FUNCTION REQIRD(IYEARI,IDAYI)

$INCLUDE COM.ICL
```

```
IF(IYEAR.EQ.IYEARI.AND.IYEAR.EQ.87)THEN
   REQIRD = REAL(480*(IPV(ICAL87,IDAYI)))-480.0
ELSEIF(IYEAR.EQ.IYEARI.AND.IYEAR.EQ.88)THEN
   REQIRD = REAL(480*(IPV(ICAL88,IDAYI)))-480.0
ELSEIF(IYEAR.EQ.IYEARI.AND.IYEAR.EQ.89)THEN
   REQIRD = REAL(480*(IPV(ICAL89,IDAYI)))-480.0
ELSEIF(IYEAR.LT.IYEARI.AND.IYEAR.EQ.87)THEN
   REQIRD = REAL((251+IDAYI)*480)-480.0
ELSEIF(IYEAR.LT.IYEARI.AND.IYEAR.EQ.88)THEN
   REQIRD = REAL((500+IDAYI)*480)-480.0
ELSEIF(IYEAR.LT.IYEARI.AND.IYEAR.EQ.89)THEN
   REQIRD = REAL((748+IDAYI)*480)-480.0
ELSE
   REQIRD=0.1
ENDIF
IF(REQIRD.LT.0.0)THEN
   REQIRD = 0.0
ENDIF
RETURN
END
```

Experement end point include file ENEDE.ICL

```
      SUBROUTINE ENDED

$INCLUDE COM.ICL

      CHARACTER*6 TT
      REAL STKOUT

      CALL DISTXT(1,-1,0,'SAVING RESULTS')

C.......Output results to disk

      JJ = IPV(NAME,1)
      TT = 'STAT'//CFI(IPER,2)

      OPEN(4,FILE=TT,STATUS='NEW')
      WRITE(4,7) '      '
      WRITE(4,7) '      '
    7 FORMAT(A)
    8 FORMAT(6F10.3)
    6 FORMAT(8F7.3)


      WRITE(4,7) ' EXPERIMENT:     '
      WRITE(4,7) ' ---------------      '
      WRITE(4,7) '      '
      WRITE(4,7) '      '
      WRITE(4,7) ' PARAMETER VALUES:-     '
      WRITE(4,7) ' -----------------     '
      WRITE(4,7) '      '
      WRITE(4,7)  '      PCBT/B    SYST/B       PCBTT       S:
                     LOAD'
      WRITE(4,7)  '        ------    ------      -----       --
                     ----'

      AA = RPV(ITRBAT,1)
      BB = RPV(ITRBAT,2)
      CC = RPV(ITRANS,1)
      DD = RPV(ITRANS,2)
      EE = RPV(IPLAN,5)

      WRITE(4,9) AA,BB,CC,DD,EE
    9 FORMAT(F10.2,F10.2,F10.2,F10.2,F10.2)
      CALL TRACE ('RESULTS FILE HEADER WRITTEN')

      WRITE(4,7) '      '
      WRITE(4,7) '      '
      WRITE(4,7) '  EXPERIMENTAL RESULTS FOR: '//TT
      WRITE(4,7) '  ------------------------------ '

      WRITE(4,7) '      '
      WRITE(4,7) '      '

      CALL ANALTS(IBASIZ(1),R1,R2,RMIN,RMAX,RMEAN,R3,R4,DUI
```

```
      CALL ANLTS(IBASIZ(2),RR1,RR2,RRMIN,RRMAX,
&        RRMEAN,RR3,RR4,DUR)

      WRITE(4,7) '      '
      WRITE(4,7) 'SUMMARIZED:    PCB BATCH SIZE       SYS B
                 SIZE  '
      WRITE(4,7)               '-----------     ---------
                 -------------- '
      WRITE(4,7) '      '
      WRITE(4,7)  '      MIN        MEAN       MAX MIN
                     MAX'
      WRITE(4,7)   '          ---        ----       ---
                 ----       ---'

      WRITE(4,8) RMIN,RMEAN,RMAX,RRMIN,RRMEAN,RRMAX

      CALL ANALTS(INPROG(1),R1,R2,RMIN,RMAX,RMEAN,R3,R4,DU
      CALL ANALTS(INPROG(2),RR1,RR2,RRMIN,RRMAX,
                 RRMEAN,RR3,RR4,DUR)

      WRITE(4,7) '      '
      WRITE(4,7) 'SUMMARIZED:     PCB WIP                SYS
      WRITE(4,7) '-----------     -------------- '
      WRITE(4,7) '      '
      WRITE(4,7)   '       MIN         MEAN        MAX
                           MEAN        MAX'
      WRITE(4,7)     '         ---         ----        ---
                           ----        ---'

      WRITE(4,8) RMIN,RMEAN,RMAX,RRMIN,RRMEAN,RRMAX

      CALL ANALTS(IFLOTM(1),R1,R2,RMIN,RMAX,RMEAN,R3,R4,DU
      CALL ANALTS(IFLOTM(2),RR1,RR2,RRMIN,
                 RRMAX,RRMEAN,RR3,RR4,DUR)

      WRITE(4,7) '      '
      WRITE(4,7) 'SUMMARIZED:    PCB FLOW                 SYS
      WRITE(4,7) '-----------    --------                 ----
      WRITE(4,7) '      '
      WRITE(4,7)  '    MIN    MEAN     MAX    STD   MIN
                     MAX     STD'
      WRITE(4,7)   '     ---    ----     ---    ---   ---
                     ---     ---'

      WRITE(4,6) RMIN,RMEAN,RMAX,R3,RRMIN,RRMEAN,RRMAX,RR3

      CALL ANALTS(ILATE(1),R1,R2,RMIN,RMAX,
                 RMEAN,R3,R4,DUR)
      CALL ANALTS(ILATE(2),RR1,RR2,RRMIN,
                 RRMAX,RRMEAN,RR3,RR4,DUR)

      WRITE(4,7) '      '
      WRITE(4,7)  'SUMMARIZED:    PCB D/D ACCURACY
                 D/D ACCURACY '
      WRITE(4,7)              '-----------    -----------
```

```
                 ---------------- '
      WRITE(4,7) '      '
      WRITE(4,7)  ' MIN     MEAN     MAX     STD    MIN MEAN
                  STD'
      WRITE(4,7)   '   ---     ----      ---     ---    ---    .
                  ---     ---'

      WRITE(4,6) RMIN,RMEAN,RMAX,R3,RRMIN,RRMEAN,RRMAX,RR3

      WRITE(4,7) '      '
      WRITE(4,7) '          W/C UTILIZATION DATA  '
      WRITE(4,7) '          --------------------  '
      WRITE(4,7) '      '
      WRITE(4,7) '      W/C     MIN     MEAN      MAX  '
      WRITE(4,7) '      ---     ---     ----      ---  '


      DO 11 II = 1,11


              CALL ANALTS(IUTILZ(II),R1,R2,RMIN,
                     RMAX,RMEAN,R3,R4,DUR)
              WRITE(4,10) II,RMIN,RMEAN,RMAX
10            FORMAT(I8,F8.2,F8.2,F8.2)

11    CONTINUE


      WRITE(4,7) '      '
      WRITE(4,7) '          W/C INPUT QUEUE DATA  '
      WRITE(4,7) '          --------------------  '
      WRITE(4,7) '      '
      WRITE(4,7) '      W/C     MIN     MEAN      MAX  '
      WRITE(4,7) '      ---     ---     ----      ---  '


14    DO 13 II = 1,11

              CALL ANALTS(INQUE(II),R1,R2,RMIN,
                     RMAX,RMEAN,R3,R4,DUR)

              WRITE(4,12) II,RMIN,RMEAN,RMAX
12            FORMAT(I8,F8.2,F8.2,F8.2)
13    CONTINUE

      WRITE(4,7) '      '
      WRITE(4,7) '     INDIVIDUAL WIP & STOCK DATA:- '
      WRITE(4,7) '     ---------------------------    '
      WRITE(4,7) '      '
      WRITE(4,7)  '              BATCH SIZES             I
                  TIMES
  &     LATENESS  '
      WRITE(4,7)         '                              -------.
                  ----------
```

```
&     ------------'
      WRITE(4,7) '       '
      WRITE(4,7)  '      TYPE      MIN    MEAN       MAX
                       MEAN      MAX    MIN    MEAN MAX '
      WRITE(4,7)     '       ----     ---    ----      ---
                            ----     ---
&     ---    ----       --- '
      DO 20 II = 1,99

      CALL ANALTS(SLED(II),R1,R2,RMIN,RMAX,RMEAN,R3,R4,DUR)
      CALL ANALTS(SSIZE(II),S1,S2,SMIN,SMAX,SMEAN,S3,S4,DUI
      CALL ANALTS(SLAT(II),S1,S2,RRMIN,RRMAX,RRMEAN,S3,S4,I

      NNMIN = INT(RRMIN)
      NNMEAN = INT(RRMEAN)
      NNMAX = INT(RRMAX)

      IF(SMEAN.EQ.0.0)GOTO 20

      WRITE(4,15) II,SMIN,SMEAN,SMAX,RMIN,
&   RMEAN,RMAX,NNMIN,NNMEAN,NNMAX
   15 FORMAT(I8,F8.2,F8.2,F8.2,F8.2,
             F8.2,F8.2,1X,I4,5X,I4,5X,I4)

   20   CONTINUE

      DO 27 II = 100,255

      CALL ANALTS(BLED(II),R1,R2,RMIN,RMAX,RMEAN,R3,R4,DUR)
      CALL ANALTS(BSIZE(II),S1,S2,SMIN,SMAX,SMEAN,S3,S4,DUI
      CALL ANALTS(BLAT(II),S1,S2,RRMIN,RRMAX,RRMEAN,S3,S4,I
      NNMIN = INT(RRMIN)
      NNMEAN = INT(RRMEAN)
      NNMAX = INT(RRMAX)

      IF(SMEAN.EQ.0.0)GOTO 27

      WRITE(4,25) II,SMIN,SMEAN,SMAX,RMIN,RMEAN,
                       RMAX,NNMIN,NNMEAN,NNMAX
   25   FORMAT(I8,F8.2,F8.2,F8.2,F8.2,F8.2,
                       F8.2,1X,I4,5X,I4,5X,I4)

   27   CONTINUE

      WRITE(4,7) '       '
      WRITE(4,7) '       '
      WRITE(4,7) 'SYSTEM LAUNCH ATTEMPTS:                      S'
                                 LEVEL INFO'M:'
      WRITE(4,7) '       '
      WRITE(4,7) '                LAUNCH '
      WRITE(4,7)  '      TYPE     ATTEMPTS   % STOCKOUT
                                 MEAN       MAX'
      WRITE(4,7)   '      ----      ------    ----------
                                 ----      ---'
```

```
      DO 125 II = 1,99

         CALL ANALTS(SLEV(II),S1,S2,RMIN,RMAX,RMEAN,S3,S4,I
         ILNC =  IPV(IACESS(II),1)
         ISTO =  IPV(IACESS(II),2)

         IF(ILNC.EQ.0)THEN
                 STKOUT = 0
                 GOTO 122
         ENDIF

         STKOUT =  (REAL(ISTO)/REAL(ILNC))
         STKOUT  = STKOUT*100

122      NNMIN = INT(RMIN)
         NNMEAN = INT(RMEAN)
         NNMAX = INT(RMAX)

         IF(NNMEAN.EQ.0)GOTO 125

         WRITE(4,123) II,ILNC,STKOUT,NNMIN,NNMEAN,NNMAX
123      FORMAT(5X,I3,7X,I4,6X,F7.2,6X,I4,5X,I4,5X,I4)

125   CONTINUE

      WRITE(4,7) '      '
      WRITE(4,7) '      '
      WRITE(4,7) ' INDIVIDUAL PCB STOCK LEVEL INFO'M:    '
      WRITE(4,7) '      '

      DO 126 II = 100,255
      CALL ANALTS(SLEV(II),S1,S2,RRMIN,RRMAX,RRMEAN,S3,S4,I

         NNMIN = INT(RRMIN)
         NNMEAN = INT(RRMEAN)
         NNMAX = INT(RRMAX)

         IF(NNMEAN.EQ.0)GOTO 126

         WRITE(4,127) II,NNMIN,NNMEAN,NNMAX
127      FORMAT(5X,I3,30X,I4,5X,I4,5X,I4)

126      CONTINUE

      CLOSE(4)

      IF(IPER.GT.39)CALL SHUT

      RETURN
      END
```

Comminication Routines, COMMS.ICL

```
      SUBROUTINE WIPOUT

$INCLUDE COM.ICL

      CHARACTER DAYOUT*10
      CHARACTER ACAB*10

      IF(IBABEL.EQ.0)RETURN
      COMMD = 'HELL TEST/TEST'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)
      COMMD = 'ASP/PINNACLE'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)
 10   COMMD = 'R $TMSOBJ/ASP010'
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

      IREC = 0

      IF(IYEAR.EQ.87.AND.IPER.LT.2)THEN
         NIJ=1
      ELSEIF(IYEAR.EQ.87)THEN
         NIJ=IPV(IPER87,IPER)
      ELSEIF(IYEAR.EQ.88.AND.IPER.EQ.1)THEN
         NIJ=IPV(IPER87,52)
      ELSEIF(IYEAR.EQ.88.AND.IPER.GT.1)THEN
         NIJ=IPV(IPER88,IPER)
      ELSEIF(IYEAR.EQ.89.AND.IPER.EQ.1)THEN
         NIJ=IPV(IPER88,51)
      ELSEIF(IYEAR.EQ.89.AND.IPER.GT.1)THEN
         NIJ=IPV(IPER89,IPER)
      ENDIF

      DO 150 LJK = 1,NIJ
      IF(IPER.LT.10)THEN
         WRITE(ACAB,20)'STK0',IPER,LJK
 20            FORMAT(A4,I1,I1)
      ELSE
         WRITE(ACAB,25)'STK',IPER,LJK
 25            FORMAT(A3,I2,I1)
      ENDIF

      READ(ACAB,30) DAYOUT
 30   FORMAT(A6)
      OPEN(4,FILE=DAYOUT,STATUS='OLD',ERR=135)
 129  READ(4,130,END=140) TOUT
 130  FORMAT(A80)
      COMMD = TOUT
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)
      IREC = IREC + 1
      GO TO 129
```

```
  135    CALL DISTXT(1,-1,0,'Cannot open '//DAYOUT)
         GO TO 191
  145    CLOSE(4)
  150    CONTINUE

C.......Test for correct number of records.

         IF(IREC.LT.10)THEN
            WRITE(ACAB,160)'!Z!000',IREC
  160             FORMAT(A6,I1)
         ELSEIF(IREC.LT.100)THEN
                  WRITE(ACAB,170)'!Z!00',IREC
  170             FORMAT(A5,I2)
         ELSEIF(IREC.LT.1000)THEN
            WRITE(ACAB,180)'!Z!0',IREC
  180             FORMAT(A4,I3)
         ENDIF

         READ(ACAB,185)COMMD
  185    FORMAT(A7)

         CALL COMPRT(COMMD)
         CALL COMUP(COMMD)

C.......Test for correct number of records.

         COMMD = '!E!'
         CALL COMPRT(COMMD)
         CALL COMUP(COMMD)

         GO TO 191

  190    CALL DISTXT(1,-1,0,'cannot write to '//DAYOUT)
  191    RETURN
         END

         SUBROUTINE NXTMPS

$INCLUDE COM.ICL

         CHARACTER MPSUP*9
         CHARACTER ACAB*10

         IF(IBABEL.EQ.0)RETURN

C.......Determine next MPS.

         IF(IPER.LT.10)THEN
            MPSUP = 'MPS0'//CFI(IPER,1)
         ELSE
            MPSUP = 'MPS'//CFI(IPER,2)
         ENDIF

C.......Call MPS update programme
```

```
      COMMD = 'HELL TEST/TEST'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'ASP/PINNACLE'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'R $TMSOBJ/ASP020'
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

C.......Output MPS update to TMS.

      IREC=0
      OPEN(4,FILE=MPSUP,STATUS='OLD',ERR=235)
  229 READ(4,230,END=240) TOUT
  230 FORMAT(A80)

      CALL INQUAN(TOUT)
      IREC=IREC+1
      COMMD = TOUT
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

      GO TO 229
  235 CALL DISTXT(1,-1,0,'Cannot open '//MPSUP//'....')

      GO TO 246
  240 CALL DISTXT(1,-1,0,MPSUP//' upload completed....')

  245 CLOSE(4)

C.......Test for correct number of records.

      IF(IREC.LT.10)THEN
         WRITE(ACAB,160)'!Z!000',IREC
  160       FORMAT(A6,I1)
      ELSEIF(IREC.LT.100)THEN
         WRITE(ACAB,170)'!Z!00',IREC
  170       FORMAT(A5,I2)
      ELSEIF(IREC.LT.1000)THEN
         WRITE(ACAB,180)'!Z!0',IREC
  180       FORMAT(A4,I3)
      ENDIF

      READ(ACAB,185)COMMD
  185 FORMAT(A7)

      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

C.......Test for correct number of records.

      COMMD = '!E!'
```

```
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

C.......NOTE: !E! automatically triggers an MRP run.

  246 RETURN
      END

      SUBROUTINE INQUAN(TOUT)

$INCLUDE COM.ICL

      CHARACTER INCDEM*80
      CHARACTER AAA*3
      CHARACTER AAJJ*20
      WRITE(INCDEM,10)TOUT
   10 FORMAT(A80)

      READ(INCDEM,20)AAA,AAJJ,IPERID,IQUAN
   20 FORMAT(A3,A20,I2,1X,I5)

      IQUAN=INT(REAL(IQUAN)*RPV(IPLAN,5))

      WRITE(INCDEM,30)AAA,AAJJ,IPERID,IQUAN
   30 FORMAT(A3,A20,I2,1X,I5)

      READ(INCDEM,40)TOUT
   40 FORMAT(A80)

      RETURN
      END


      SUBROUTINE SALES

$INCLUDE COM.ICL

      CHARACTER SALEUP*9
      CHARACTER ACAB*10

      IF(IBABEL.EQ.0)RETURN

C.......Determine next SALES.

      IF(IPER.LT.10)THEN
              SALEUP = 'SALE0'//CFI(IPER,1)
      ELSE
              SALEUP = 'SALE'//CFI(IPER,2)
      ENDIF

C.......Call SALES update programme

      COMMD = 'HELL TEST/TEST'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)
```

```
      COMMD = 'ASP/PINNACLE'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'R $TMSOBJ/ASP030'
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

C.......Output SALES update to TMS.

      IREC=0
      OPEN(4,FILE=SALEUP,STATUS='OLD',ERR=235)

  229 READ(4,230,END=240) TOUT
  230 FORMAT(A80)

      CALL INCSAL(TOUT)

      IREC=IREC+1
      COMMD = TOUT
      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

      GO TO 229
  235 CALL DISTXT(1,-1,0,'Cannot open '//SALEUP//'....')

      GO TO 246
  240 CALL DISTXT(1,-1,0,SALEUP//' upload completed....')

  245 CLOSE(4)

C.......Test for correct number of records.

      IF(IREC.LT.10)THEN
         WRITE(ACAB,160)'!Z!000',IREC
  160          FORMAT(A6,I1)
      ELSEIF(IREC.LT.100)THEN
               WRITE(ACAB,170)'!Z!00',IREC
  170          FORMAT(A5,I2)
      ELSEIF(IREC.LT.1000)THEN
               WRITE(ACAB,180)'!Z!0',IREC
  180          FORMAT(A4,I3)
      ENDIF

      READ(ACAB,185)COMMD
  185 FORMAT(A7)

      CALL COMPRT(COMMD)
      CALL COMUP(COMMD)

C.......Test for correct number of records.

      COMMD = '!E!'
      CALL COMPRT(COMMD)
```

```
      CALL COMUP(COMMD)

C.......NOTE: !E! automatically triggers an MRP run.

  246 RETURN
      END

      SUBROUTINE INCSAL(TOUT)

$INCLUDE COM.ICL

      CHARACTER INCDEM*80
      CHARACTER AAA*3
      CHARACTER AAJJ*20
      CHARACTER DUM*2

      WRITE(INCDEM,10)TOUT
   10 FORMAT(A80)

      READ(INCDEM,20)AAA,AAJJ,DUM,IQUAN
   20 FORMAT(A3,A20,A2,I9)

      IQUAN=INT(REAL(IQUAN)*RPV(IPLAN,5))

      WRITE(INCDEM,30)AAA,AAJJ,DUM,IQUAN
   30 FORMAT(A3,A20,A2,I9)

      READ(INCDEM,40)TOUT
   40 FORMAT(A80)

      RETURN
      END

      SUBROUTINE MSGDWN

$INCLUDE COM.ICL

      CHARACTER ACAB*10
      CHARACTER ACTFIL*10

      IF(IBABEL.EQ.0)RETURN

C.......Download Action messages to disc from TMS.

      IF(IPER.LT.10)THEN
              ACTFIL='ACTN0'//CFI(IPER,1)
      ELSE
              ACTFIL='ACTN'//CFI(IPER,2)
      ENDIF

      OPEN(4,FILE=ACTFIL,STATUS='NEW',ERR=35)

C.......Call order update programme

      COMMD = 'HELL TEST/TEST'
```

```
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'ASP/PINNACLE'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'R $TMSOBJ/ASP050'
      CALL COMPRT(COMMD)
      CALL COMDWN(COMMD)

C.......Send ready prompt to the Borroughs

      IREC = 0

   20 COMMD = '!C!'
      CALL COMDWN(COMMD)
      ILEN = LEN(COMMD) - 2

         DO 25 NN = 1,ILEN
              IF(COMMD(NN:NN+2).EQ.'!Z!') GO TO 40
   25         CONTINUE

      WRITE(4,30) COMMD(4:)
   30 FORMAT(A64)

      IREC = IREC + 1
          GO TO 20
   35 CALL DISTXT(1,-1,0,'Cannot open '//ACTFIL)

      GO TO 99
   40 CLOSE(4)

C.......Test for correct number of records.
      IF(IREC.LT.10)THEN
         WRITE(ACAB,160)'!Z!000',IREC
  160         FORMAT(A6,I1)
      ELSEIF(IREC.LT.100)THEN
         WRITE(ACAB,170)'!Z!00',IREC
  170         FORMAT(A5,I2)
      ELSEIF(IREC.LT.1000)THEN
         WRITE(ACAB,180)'!Z!0',IREC
  180         FORMAT(A4,I3)
      ENDIF

      READ(ACAB,185)COMMD
  185 FORMAT(A7)

      CALL COMPRT(COMMD)
      CALL COMDWN(COMMD)

   99 RETURN
      END
```

```
      SUBROUTINE WIPDWN

$INCLUDE COM.ICL

      CHARACTER ORDERS*9
      CHARACTER ACAB*10

      IF(IBABEL.EQ.0)RETURN

C.......Download WIP orders to disc from TMS.
C.......Determine next SALES.

      IF(IPER.LT.10)THEN
              ORDERS = 'ORDER0'//CFI(IPER,1)
      ELSE
              ORDERS = 'ORDER'//CFI(IPER,2)
      ENDIF

      OPEN(4,FILE=ORDERS,STATUS='NEW',ERR=35)

C.......Call order update programme

      COMMD = 'HELL TEST/TEST'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'ASP/PINNACLE'
      CALL COMPRT(COMMD)
      CALL COMOPN(COMMD)

      COMMD = 'R $TMSOBJ/ASP040'
      CALL COMPRT(COMMD)
      CALL COMDWN(COMMD)

C.......Send ready prompt to the Borroughs

      IREC = 0

   20 COMMD = '!C!'

      CALL COMDWN(COMMD)

      ILEN = LEN(COMMD) - 2

              DO 25 NN = 1,ILEN
                      IF(COMMD(NN:NN+2).EQ.'!Z!') GO TO 40
   25         CONTINUE

      WRITE(4,30) COMMD(4:57)
   30 FORMAT(A54)

      IREC = IREC + 1

      GO TO 20
```

```
   35   CALL DISTXT(1,-1,0,'Cannot open '//ORDERS//'....')

        GO TO 99

   40   CLOSE(4)

C.......Test for correct number of records.

        IF(IREC.LT.10)THEN
                WRITE(ACAB,160)'!Z!000',IREC
  160           FORMAT(A6,I1)
        ELSEIF(IREC.LT.100)THEN
                WRITE(ACAB,170)'!Z!00',IREC
  170           FORMAT(A5,I2)
        ELSEIF(IREC.LT.1000)THEN
                WRITE(ACAB,180)'!Z!0',IREC
  180           FORMAT(A4,I3)
        ENDIF

        READ(ACAB,185)COMMD
  185   FORMAT(A7)

        CALL COMPRT(COMMD)
        CALL COMDWN(COMMD)

        CALL DISTXT(1,-1,0,'WIP order download completed....'

   99   RETURN
        END


        SUBROUTINE COMOPN(COMMD)

$INCLUDE COM.ICL

        CHARACTER TXT*100
        TXT = COMMD
        DO 5 L = LEN(TXT),1,-1
        IF(TXT(L:L).NE.' ') GO TO 8
    5   CONTINUE
    8   TXT = TXT(:L)

C.......Send outgoing text.

        CALL DISTXT(1,-1,0,TXT)
    9   CALL SENDTX(TXT)

C.......Use  outgoing  text  to determine how to
C.......read incoming text.

        IF(TXT(:4).EQ.'HELL')THEN
   10           CALL GETTEX(TXT,ILEN)
                ILEN = ILEN - 3
                DO 15 NN = 1,ILEN
```

```
                   IF(TXT(NN:NN+3).EQ.'#ENT')THEN
                            GO TO 100
                   ELSEIF(TXT(NN:NN+3).EQ.'#LOG')THEN
                            TXT='HELL TEST/TEST'
                            GO TO 9
                   ENDIF
   15           CONTINUE
                GO TO 10
                 ELSEIF(TXT(:4).EQ.'ASP/')THEN
   20           CALL GETTEX(TXT,ILEN)
                ILEN = ILEN - 2
                DO 25 NN = 1,ILEN
                   IF(TXT(NN:NN+2).EQ.'#SE') GO TO 100
   25           CONTINUE
                GO TO 20
             ENDIF
  100   COMMD = TXT
        CALL COMPRT(COMMD)

        RETURN
        END


        SUBROUTINE COMDWN(COMMD)

$INCLUDE COM.ICL

        CHARACTER TXT*100
        TXT = COMMD

C.......Send outgoing text.

        CALL DISTXT(1,-1,0,COMMD(:60))
        CALL SENDTX(TXT)

C.......Use  outgoing  text  to determine how to
C.......read incoming text.

        IF(TXT(:9).EQ.'R $TMSOBJ')THEN
   30           CALL GETTEX(TXT,ILEN)
                ILEN = ILEN - 2
           DO 35 NN = 1,ILEN
                   IF(TXT(NN:NN+2).EQ.'!C!') GO TO 100
   35           CONTINUE
                GO TO 30
        ELSEIF(TXT(:3).EQ.'!C!')THEN
   40           CALL GETTEX(TXT,ILEN)
                ILEN = ILEN - 2
                DO 45 NN = 1,ILEN
        IF(TXT(NN:NN+2).EQ.'!P!'.OR.TXT(NN:NN+2).EQ.'!Z!')THI
        GO TO 100
        ENDIF
   45           CONTINUE
                GO TO 40
```

```
      ELSEIF(TXT(:3).EQ.'!Z!')THEN
50            CALL GETTEX(TXT,ILEN)
              ILEN = ILEN - 2
      DO 55 NN = 1,ILEN
              IF(TXT(NN:NN+2).EQ.'#ET')THEN
              GO TO 100
              ENDIF
55            CONTINUE
      GO TO 50

      ENDIF

100   COMMD = TXT
      CALL DISTXT(1,-1,0,COMMD(:60))

      RETURN
      END


      SUBROUTINE COMUP(COMMD)

$INCLUDE COM.ICL

      CHARACTER TXT*100
      TXT = COMMD
C.......Send outgoing text.
      CALL DISTXT(1,-1,0,COMMD(:60))
      CALL SENDTX(TXT)
C.......Use  outgoing  text  to determine how to  read  incoı
C.......text.

      IF(TXT(:9).EQ.'R $TMSOBJ')THEN
30            CALL GETTEX(TXT,ILEN)
              ILEN = ILEN - 2
              DO 35 NN = 1,ILEN
              IF(TXT(NN:NN+2).EQ.'!C!') GO TO 100
35            CONTINUE
              GO TO 30
      ELSEIF(TXT(:3).EQ.'!P!')THEN
40            CALL GETTEX(TXT,ILEN)
              ILEN = ILEN - 2
              DO 45 NN = 1,ILEN
              IF(TXT(NN:NN+2).EQ.'!C!')THEN
                 GO TO 100
              ENDIF
45            CONTINUE
              GO TO 40
      ELSEIF(TXT(:3).EQ.'!Z!')THEN
50            CALL GETTEX(TXT,ILEN)
              ILEN = ILEN - 2
              DO 55 NN = 1,ILEN
                 IF(TXT(NN:NN+2).EQ.'!Z!')THEN
                        GO TO 100
                 ENDIF
55            CONTINUE
```

```
             GO TO 50
      ELSEIF(TXT(:3).EQ.'!E!')THEN
  60         CALL GETTEX(TXT,ILEN)
             ILEN = ILEN - 2
             DO 65 NN = 1,ILEN
                IF(TXT(NN:NN+2).EQ.'#ET')THEN
                       GO TO 100
                ENDIF
  65         CONTINUE
             GO TO 60
      ENDIF
 100  COMMD = TXT
      CALL DISTXT(1,-1,0,COMMD(:60))

      RETURN
      END


      SUBROUTINE COMPRT(COMMD)

$INCLUDE COM.ICL

      CHARACTER TXT*100

      TXT = COMMD

C.......Send outgoing text to the printer, for validation.

      OPEN(5,FILE='LST:',STATUS='NEW',ERR=99)
           WRITE(5,10)TXT
  10  FORMAT(A80)
      CLOSE(5)
      GO TO 100
  99  CALL DISTXT(1,-1,0,'ERROR opening LST:')
 100  RETURN
      END

      SUBROUTINE SENDTX(TXT)

      INTEGER*2 IARR(300)

      CHARACTER TXT*(*)

      DO 10 L = LEN(TXT),1,-1
         IF(TXT(L:L).NE.' ') GO TO 20
  10  CONTINUE
      L = 0
  20  JJ = 0
      DO 30 II = 1,L
              JJ = JJ + 1
             IARR(JJ) = ICHAR(TXT(II:II))
  30  CONTINUE
      IARR(JJ+1) = 13
      IARR(JJ+2) = 0
      CALL TALK01(IARR)
```

```
      RETURN
      END


      SUBROUTINE GETTEX(TXT,ILEN)

      INTEGER*2 IARR(300)
      CHARACTER TXT*(*)

      CALL TALK00(IARR)

      DO 20 II = 1,300
              IF(IARR(II).EQ.0) THEN
                      ILEN = II - 1
                      TXT = TXT(:ILEN)
                      RETURN
              ENDIF
      TXT(II:II) = CHAR(IARR(II))
20    CONTINUE

      RETURN
      END


      SUBROUTINE TALK00(IARR)

      INTEGER*2 IARR(*)

      IC = 0
10    I = ITKGET()
      IF(I.EQ.0) GO TO 10
      IF(I.GT.127) I = I - 128
      IF(I.LT.32) THEN
              IF(I.EQ.3.AND.IC.GT.0)THEN
                      IC=IC+1
                      IARR(IC)=0
                      RETURN
              ENDIF
              GO TO 10
      ENDIF
      IF(I.LT.0)THEN
              I = I + 128
              I = MOD(I,128)
      ENDIF
      IC = IC + 1
      IARR(IC) = I
      GO TO 10
      END

      SUBROUTINE TALK01(IARR)

      INTEGER*2 IARR(*)
      INTEGER*1 JJ
```

```
      IC = 0
10    IC = IC + 1
      JJ = IARR(IC)
      IF(JJ.EQ.0) GO TO 20
      CALL ITKPUT(JJ)
      IF(JJ.EQ.13) JJ = 10
      GO TO 10
20    RETURN
      END
```

```
Record event include file RECORD.ICL
        SUBROUTINE RECORD

$INCLUDE COM.ICL
        IUT = 0
        DO 10 II = 1,3
           IUT = ISIZOF(IMCPRO(II,1))
           RR   = REAL(IUT)
           RMAC = REAL(IPV(NMAC(1),II))
           RUT  = 100.0*(RR/RMAC)
           CALL ABSROB(IUTILZ(II),RUT)
   10   CONTINUE
        II = 4
        IUT = 0
        DO 20 JJ = 1,9
           IUT = IUT + ISIZOF(IMCPRO(II,JJ))
   20   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
        RUT  = 100.0*(RR/RMAC)
        CALL ABSROB(IUTILZ(II),RUT)
        II = 5
        IUT = 0
        DO 30 JJ = 1,3
           IUT = IUT + ISIZOF(IMCPRO(II,JJ))
   30   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
        RUT  = 100.0*(RR/RMAC)
        CALL ABSROB(IUTILZ(II),RUT)
        II = 6
        IUT = 0
        DO 40 JJ = 1,6
           IUT = IUT + ISIZOF(IMCPRO(II,JJ))
   40   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
        RUT  = 100.0*(RR/RMAC)
        CALL ABSROB(IUTILZ(II),RUT)
        II = 7
        IUT = 0
        DO 50 JJ = 1,2
           IUT = IUT + ISIZOF(IMCPRO(II,JJ))
   50   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
        RUT  = 100.0*(RR/RMAC)
        CALL ABSROB(IUTILZ(II),RUT)
        II = 8
        IUT = 0
        DO 60 JJ = 1,2
           IUT = IUT + ISIZOF(IMCPRO(II,JJ))
   60   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
```

```
      RUT  = 100.0*(RR/RMAC)
      CALL ABSROB(IUTILZ(II),RUT)
      II = 9
      IUT = 0
      DO 70 JJ = 1,20
         IUT = IUT + ISIZOF(IMCPRO(II,JJ))
 70   CONTINUE
         RR   = REAL(IUT)
         RMAC = REAL(IPV(NMAC(1),II))
         RUT  = 100.0*(RR/RMAC)
         CALL ABSROB(IUTILZ(II),RUT)
         II = 10
         IUT = 0
      DO 80 JJ = 1,3
         IUT = IUT + ISIZOF(IMCPRO(II,JJ))
80   CONTINUE
     RR   = REAL(IUT)
     RMAC = REAL(IPV(NMAC(1),II))
     RUT  = 100.0*(RR/RMAC)
     CALL ABSROB(IUTILZ(II),RUT)
     II = 11
     IUT = 0
     DO 90 JJ = 1,8
        IUT = IUT + ISIZOF(IMCPRO(II,JJ))
 90   CONTINUE
        RR   = REAL(IUT)
        RMAC = REAL(IPV(NMAC(1),II))
        RUT  = 100.0*(RR/RMAC)
        CALL ABSROB(IUTILZ(II),RUT)
        CALL ABSROB(INPROG(1),RPV(IWIP(1),1))
        CALL ABSROB(INPROG(2),RPV(IWIP(2),1))
      DO 110 JJ =1,11
         AA=0
         KK = ISIZOF(IBATIN(JJ))
         IF(KK.GT.0)THEN
            DO 100 LL =1,KK
               AA = AA+RAT(MEMBER(IBATIN(JJ),LL),4)
100         CONTINUE
         CALL ABSROB(INQUE(JJ),AA)
         ELSE
         CALL ABSROB(INQUE(JJ),0.0)
         ENDIF
110   CONTINUE
      BB=0
      DO 130 JJ =1,11
         KK = ISIZOF(IBAOUT(JJ))
         IF(KK.GT.0)THEN
                 DO 120 LL =1,KK
                         BB = BB+RAT(MEMBER(IBAOUT(JJ),LL),
120                  CONTINUE
         ENDIF
130   CONTINUE
      CALL ABSROB(INMOVE(1),BB)
      CALL SCHEDL('RECORD',RPV(IPLAN,4),IMACH(20,1))
      CALL DONTC
```

```
      RETURN
      END
Statistics Initialization.

      DO 5 J = 1,11
         IF(J.LT.10)THEN
                 CALL MAKETS(INQUE(J),'QU '//CFI(J,1))
         ELSE
                 CALL MAKETS(INQUE(J),'QU'//CFI(J,2))
         ENDIF
 5    CONTINUE

      DO 10 J = 1,11
         IF(J.LT.10)THEN
                 CALL MAKETS(IUTILZ(J),'Ut '//CFI(J,1))
         ELSE
                 CALL MAKETS(IUTILZ(J),'Ut'//CFI(J,2))
         ENDIF
10    CONTINUE

      CALL MAKETS(IFLOTM(1),'BFLO')
      CALL MAKETS(IFLOTM(2),'SFLO')
      CALL MAKETS(IBASIZ(1),'BSIZ')
      CALL MAKETS(IBASIZ(2),'SSIZ')
      CALL MAKETS(ILATE(1),'BLAT')
      CALL MAKETS(ILATE(2),'SLAT')

      DO 7 ITY = 1,99
      CALL MAKETS(SLAT(ITY),'L '//CFI(ITY,2))
 7    CONTINUE

      DO 8 ITT = 100,255
      CALL MAKETS(BLAT(ITT),'L'//CFI(ITT,3))
 8    CONTINUE
      CALL MAKETS(INMOVE(1),'BMOV')
      CALL MAKETS(INMOVE(2),'SMOV')
      CALL MAKETS(INPROG(1),'BWIP')
      CALL MAKETS(INPROG(2),'SWIP')
      DO 11 JJ = 1,99
      CALL MAKETS(SLED(JJ),'SF'//CFI(JJ,2))
11    CONTINUE

      DO 12 JJ = 100,255
      CALL MAKETS(BLED(JJ),'F'//CFI(JJ,3))
12    CONTINUE
      DO 13 JJ = 1,99
      CALL MAKETS(SSIZE(JJ),'SS'//CFI(JJ,2))
13    CONTINUE

      DO 14 JJ = 100,255
      CALL MAKETS(BSIZE(JJ),'S'//CFI(JJ,3))
14    CONTINUE
      DO 25 JJ = 1,99
      CALL MAKEPT(IACESS(JJ),'A'//CFI(JJ,2),'I',3)
25    CONTINUE
```

```
      DO 16 JJ = 1,255
      CALL MAKETS(SLEV(JJ),'L'//CFI(JJ,3))
16    CONTINUE
```

APPENDIX vi THE FCL MODEL USER GUIDE

USER GUIDE CONTENTS

INTRODUCTION

User guide overview

The following text presents a guide to the maintenance and use of the FCL Production System model which was built in collaboration with Aston University and Fulcrum Communications Ltd. (FCL).

The guide has been designed to serve two main purposes. First as a learning aid for those new to the model and secondly as a reference guide for the more experienced user.

It is assumed that the reader is familiar with the following:

a). The TMS Production Control system .
b). The Simulation package,OPTIK1 & 11 .
c). The syntax of Fortran-77.
d). The syntax of the Workflow language on TMS .
e). The CPM/68K word processor,Mince on the Pinnacle.
f). The 3-Phase approach to computer simulation modelling.

Reference to any of the above subjects can be found in the respective manuals . A description of the 3-Phase approach to simulation modelling as proposed by K. Tocher, can be found in the book, computer simulation in management science, by Pidd(1985).

A knowledge of the existing set-up at FCL in terms of the current production control system and the manufacturing facility is also assumed.

MODEL OVERVIEW

The model it's self can be conveniently divided in to two parts as follows :

The Production Control System Model

The actual Production Control system (TMS), running on the Boroughs B5900 mainframe computer at FCL was used here to provide the generation of suggested orders and actions for the Manufacturing Simulator.

The TMS model consists of; dedicated modules which provide production control, special routines for communication with the Pinnacle and special routines for updating the TMS data base. The code for these model elements is written in both Cobol and Algol.

The operation of the TMS model is complemented by the existence of two Utility programs;

i). GEMCOS. This provides the interface between the user and the TMS database. In normal use a suite of user friendly screens are offered by GEMCOS to enable access to and amendment of the database.

ii). CANDE. This provides a facility which lies somewhere between a word processor and an operating system. It enables the more experienced user to access and modify TMS programs, create procedural batch files (these are called workflows for which there is a specific workflow language) and communicate with other users.

The Manufacturing Simulator

The second part is a deterministic simulation model of the manufacturing facility at FCL, running on the TDI Pinnacle microcomputer. The model represents each of the machine/process and labour elements used in the manufacture, assembly and test of products and their components.

The Simulator consists of routines for; logical definition of all elements of the manufacturing facility, Pictorial definition of those elements used in the screen display, mathematical definition of the manufacturing facility rules and continually updating the model state during an experimental run. Also built into the simulator is a dedicated communications package which controls information flow between the Pinnacle and the B5900.

The Manufacturing Simulator has been written using a visual interactive simulation package called OPTIK. The package contains numerous library files. These enable the programmer to; define events , entities and lists , determine the visual representation of the model and set up OPTIK user packet arrays for information storage.

The self contained OPTIK library files(or OPTIK routines) are linked together logically using the programming language, Fortran. The Fortran code defines the simulation flow, invokes user defined routines and declares the variable parameters.

The two sub-models are linked together by a specially designed communications package. Figure 8.7 shows how the hardware for the model has been configured.

The model has been designed as a package so that when experiments are being conducted there is no need for any programming. The experimental parameters and control policies do however need to be input via data files. This procedure, although tedious for certain experiments, is not difficult. A full description of data file creation is included in this guide.

It is suggested that experiments involving the B5900 should be undertaken at times of minimum load and only with the express permission of the computer operations manager.

SOFTWARE REQUIREMENTS

The Manufacturing Simulator files

The current manufacturing simulation package is called FCFSIM (Fulcrum Communications Factory Simulator). The following files and programs must exist on the designated disk area for successful use of the model. It is assumed that all the software associated with the operating system the fortran compiler the OPTIK package and the Mince word processor already exist:

Model code file

```
D:FCFSIM.FOR -  simulation model code
```

The INCLUDE files associated with FCFSIM.FOR are ;

```
D:COMMS.ICL -  contains all common variables
D:ENDED.ICL -  end of experiment report generation
D:NEWDAY.ICL-  controls procedure between simulated days
D:COM.ICL   -  system communication software
D:FMSINT.ICL-  defines the manufacturing area
D:ASYINT.ICL-  defines the assembly area
D:STATS.ICL -  defines the statistical variables
D:RECORD.ICL-  observes the system variables
D:PINIT.ICL -  defines the pictorial representation
D:GRAPH.ICL -  generates the visual effects of the model
```

The executable simulation file

```
G:FCFSIM.68K - program to control the simulation model
```

The Database files associated with FCFSIM.68K are ;

```
E:PCB.DAT    -  PCB process and set times at each W/C
E:SYS.DAT    -  assembly process times at each W/C
E:SYSPIC.DAT-  product structure for each assembly
E:TRANSLT.DAT- model to TMS partnumber translator
E:CALEN87.DAT- the 1987 shop calendar
```

The Communication data files are ;

```
E:MPS*       -  contains periodic MPS changes
E:SALE*      -  contains periodic product sales
E:MPS00      -  contains the initial MPS demands
E:STKUP      -  contains all the system stock policies
```

The experimental parameter file is ;

```
E:EX10       -  contains experimental run parameters
```

The model utility programs include ;

D:FILE.68K - used for creating and amending data files
D:MPSUP.68K - uploads the file MPS00 to the B5900
D:PARMUP.68K- uploads the file STKUP to the B5900

The following are periodic communication data files which are created by their associated Fortran programs .

E:ORDER* - contains periodic WIP orders from TMS
E:ACTION* - contains periodic actions from TMS
E:STK* - contains daily completed stock data

The Test Database files

The following files and programs must be resident on the designated packs in the Test Database . It is assumed that all appropriate TMS module files , appropriate cobol programs and GEMCOS are all available to the Test Database:

The executable programs(on PACKT,under BEC) include ;

TMSOBJ/ASP010 - uploads stock records to TMS
TMSOBJ/ASP020 - uploads MPS changes to TMS
TMSOBJ/ASP025 - uploads stock parameters to TMS
TMSOBJ/ASP030 - uploads sales transactions to TMS
TMSOBJ/ASP040 - downloads orders to Pinnacle
TMSOBJ/ASP050 - downloads action messages to Pinnacle

TMSOBJ/ASP100 - creates the orders extract file
TMSOBJ/ASP200 - updates WIP stock transactions on TMS
TMSOBJ/ASP300 - rolls MPS and calendar forward on TMS
TMSOBJ/ASP400 - updates MPS transactions on TMS
TMSOBJ/ASP500 - creates the actions extract file
TMSOBJ/ASP601 - control program for number of MRP runs
TMSOBJ/ASP700 - updates sales transactions on TMS

The workflow files(on PACKT,under TEST) include ;

RUN/ASTONMRP- controls the TMS experimental cycle
RUN/ASTONSTK- updates the database with stock parameters
RUN/ASTONMPS- updates the database with initial MPS

The parameter files(on PACKT,under TEST) include ;

TMSPRM/MRP2RUN- contains MRP run parameters
TMSPRM/ASPCNTX- controls number of MRP runs
TMSPRM/ASP100 - determines number of days in given period

The following files are temporary files only . They are created by their associated cobol programs , and then deleted after the appropriate information has been used.

The initialisation files(on PACKT,under TEST) include ;

TMSASP/STKPRM - contains stock parameter information
TMSASP/MPSUP - contains MPS demand information

The data files(on PACKT,under TEST) include ;

TMSASP/ORDERS - contains periodic WIP order data
TMSASP/EXCEPTIONS - contains periodic action messages
TMSASP/STOCK - contains periodic completed stock data
TMSASP/SALES - contains periodic product sales data
TMSASP/MPSUP - contains periodic MPS demand information

The schemactic diagrams in figures 8.5 and 8.6 show the total FCL model run cycle from the view of the manufacturing simulator and the TMS model respectively.

The TMS model

SYSTEM TEST database

The Boroughs B5900 mainframe computer provides a Test Database and mechanism for generating WIP orders and action messages for the manufacturing simulator, running on the Pinnacle. The completed WIP orders and finished goods stock along with any changes to the MPS are processed and used to update the Test database before the next periods output is generated.

The Test database, namely SYSTEM TEST has been created using the following TMS modules ;

1). Engineering Data Base (EDC)
2). Materials Requirement Planning (MRP)
3). Stock Control (STK)
4). Work in progress (WIP)

Parts on SYSTEM TEST are modelled to PCB level only. Each part in SYSTEM TEST is represented by a definition of part details , it's partial (down to PCB level) bill of materials and a full description of stock policy and stock parameters. All relevant information has been manually transferred from the A9 Boroughs mainframe to the B5900.

Part route,work centre definition, job process time and machine set-up time information has not been included in SYSTEM TEST. This information (collected from various sources including the A9 and operation process charts) resides on the Pinnacle database and is used to define the flow of parts through the manufacturing simulator.

The TMS model emulates the real production control function by running a number of specially written cobol programs under workflow control. The following section briefly describes the specification for each of these programs.

Workflow control programs

The workflow control routines contribute to the working of the FCL model by providing the necessary update and TMS processing facilities.

RUN/ASTONSTK

This routine must be run prior to the start of any experiment. It updates the TMS data base with the selected stock policies for each part. All stocking policies are previously entered into a file on the Pinnacle and uploaded to TMS using the model utility programs . The data used to update the stock control module, for each part in the TMS model is as follows ;

i) Minimum order quantity
ii) Lead time
iii) Order policy
iv) Pan size
v) EOQ/Time bucket
vi) Minimum batch size

RUN/ASTONMPS

This routine must be run prior to the start of any experiment. It updates the TMS data base with the initial MPS demands for each top level part. The file containing this data is originally created on the Pinnacle and uploaded to TMS using the model utility programs. The production schedule is created in the TMS model by using the following data ;

i) Part type
ii) Period number
iii) Quantity

RUN/ASTONMRP

This routine forms the main section of the TMS model and as such controls all the activities necessary to generate suggested orders and action messages for the simulation model of the manufacturing facility.

After updating the experimental counter the MRP process is run. The requirements for a valid MRP run are ;

i) The MRP parameter file containing all the necessary information about the current run. The parameters include ;

- MRP run start date
- MRP run cut off date
- MRP run mode (Regenerative or Net Change)

ii) A valid MPS
iii) Valid stock policies
iv) A complete BOM for each part
v) Uniquely defined part numbers

The outputs from the MRP process are used to download the appropriate information(ie. suggested orders and action messages) to the simulation model. The suggested orders for each part is defined by the following data ;

i) Part type
ii) Order quantity
iii) Start date
iv) Due date

Each action message uses the following parameters to download its information ;

i) Part type
ii) Action message (expedite or delay)
iii) Original start date
iv) Original due date
v) New start date
vi) New due date

At this point, the work flow waits until the simulation model has completed a whole week of simulated manufacture, using the information downloaded from the TMS model as well as the data already resident on the Pinnacle database. The work flow automatically continues when the stock control module is updated with the completed stock information from the manufacturing model.

Any parts which may have been sold during the week or any changes which may have occurred in the MPS are then uploaded from the Pinnacle to the TMS model.

The production schedule is then rolled over by one week to the next period. If at this point the experimental counter has not exceeded the number of simulated periods required, then the MRP process is repeated until such time that it does.

Test database control programs

Extraction of orders from TMS - Program ASP100

This program extracts the suggested orders from TMS. A parameter file, TMSPRM/ASP100 is used to determine which orders are to be extracted.

Format of parameter file ;

ASP100 .... 6 characters
Number of days .... 3 characters

The number of days are added to the MRP parameter date to determine which orders are to be released.

The program will select and release suggested orders by start date. At the same time it will issue all the allocations for those orders to reduce stock quantities of the components.

Each order released will be written to an extract file "TMSASP/ORDERS" in the following format ;

Order number .... 12 characters
Part number .... 20 characters

Order quantity .... 12 characters
Start date .... 5 characters (julian, YYDDD)
Due date .... 5 characters (julian, YYDDD)

Update stock transactions on TMS - Program ASP200

This program updates orders and stock from information previously uploaded from the Pinnacle.

The program will update the released orders by order number from the stock transactions sent from the pinnacle system. Order quantities will be reduced by the quantity sent from the pinnacle and stock will increase.

Orders will assume to be completed once the quantity uploaded is equal or greater than the original order quantity.

Each stock record will be uploaded from the Pinnacle and written to a file "TMSASP/STOCK" in the following format ;

Part number .... 20 characters
Order quantity .... 13 characters
Order number .... 12 characters
Store location .... 6 characters (default assumed)
Bin location .... 8 characters (default assumed)

The file 'TMSASP/STOCK' will be removed from the system once updating is complete regardless of whether any errors existed.

MRP period Roll-over - Program ASP300

This program rolls the MPS forward by one period and sets the MRP parameter record to the date of the new period.

Quantities in the previous period are lost to the system.

Updating MPS transactions on TMS - Program ASP400

This program updates the MPS with information uploaded from the Pinnacle.

The program will select the MPS for the period and Part number requested and overwrite the MPS quantity with the quantity sent from the Pinnacle.

MPS changes are uploaded from the Pinnacle into a file "TMSASP/MPSUP" in the following format ;

Part number .... 20 characters
Period number .... 2 characters

Quantity .... 13 characters

The file 'TMSASP/MPSUP' will be removed from the system once updating is complete regardless of whether any errors existed.

Extraction of action messages from TMS - Program ASP500

This program extracts the suggested actions from TMS. The program will select all action messages; "Expedite", "Cancel" and "Delay" from the TMS action messages file.

Each action message will be written to an extract file "TMSASP/EXCEPTIONS" in the following format ;

```
Part number ........... 20 characters
Order number .......... 12 characters
Order line number ..... 6 characters
Action message ........ 1 character
                        E=expedite,D=delay & C=cancel
New date .............. 5 characters (julian, YYDDD)
current date .......... 5 characters (julian, YYDDD)
Order quantity ........ 12 characters
Planner code .......... 2 characters(not used)
Source ................ 1 character(not used)
```

MRP run control program - Program ASP600

This program controls the number of times the MRP cycle is to run.

It requires the parameter file, "TMSPRM/ASPCNTX" to be in the following format ;

```
ASP600 ................... 6 characters
Number of MRP runs ....... 6 characters
```

Update sales transactions on TMS - Program ASP200

This program updates customer sales transactions from information previously uploaded from the Pinnacle.

The program will update the finished stock of parts from the sales transactions sent from the pinnacle system. Part stock quantities will be reduced by the quantity sent from the pinnacle.

Each stock record will be uploaded from the Pinnacle and written to a file "TMSASP/FINSTOCK" in the following format;

Part number .... 20 characters

Order quantity .... 13 characters
Store location .... 6 characters (default assumed)
Bin location .... 8 characters (default assumed)

The file 'TMSASP/FINSTOCK' will be removed from the system once updating is complete regardless of whether any errors existed.

TMS communication programs

All programs must be called from the Pinnacle system using the following statements ;

```
HELL TEST/TEST                ** CANDE user code    **

ASP/PINNACLE                  ** access code        **

R $TMSOBJ/programname         ** program run command **
```

All programs will respond with "!C!" to inform the Pinnacle that they are ready. The programname will either perform upload functions or download functions.

Upload programs

Uploading programs are used to send information from the Pinnacle to TMS.

To send information the following is required ;

!P! followed by information being sent

The program will respond with !C!

When all records are sent the sending program should send a !Z! with a record count.

The receiving program will answer with !Z! with it's record count.

After the counts have been compared and checked for errors the sending program will send !E! to end the program.

The upload programs are ;

ASP010 - used to transfer stock records
ASP020 - used to transfer MPS updates
ASP030 - used to transfer sales records

Download programs

Downloading programs are used to send information from TMS to the Pinnacle .

The receiving program responds with a !C! from the original sign on message.

The sending program sends ;

!P! followed by information to send

The receiving program will respond with !C!

When all records are sent the sending program should send a !Z! with a record count.

The receiving program will answer with !Z! with it's record count.

After the counts have been compared and checked for errors the sending program will send !E! to end the program.

The download programs are ;

ASP040 - used to transfer orders
ASP050 - used to transfer action messages

Download programs remove the files downloaded when !E! or !Z! are received from the receiving program.

The Manufacturing Simulator

The simulation model

The event based simulation system, OPTIK has been chosen to represent the manufacturing system at FCL. The detailed factory simulation model resides on a TDI Pinnacle which is a high speed super microcomputer. The simulation process is monitored on an Intercolor microcomputer screen. A graphical representation of both the PCB manufacturing department and the assembly area have been included.

The simulation model itself is capable of including the whole product range with their individual routes, operation times and set-up times. The information is stored as specially written OPTIK data packets which can be accessed at any time from the Pinnacle database. In addition each of the work centres and machine processes are modelled separately. Each work centre has its own unique mode of operation, including setter/operator restrictions and available processing resource.

Owing to the similarities of job processing characteristics in the real system, the model has been written using general event routines. This method allows the simulation of individual processes without the overhead of excessive program code.

The following summarises features of the model event routines(see appendix v. the program source code listing) and describes how they contribute to the working of the FCL model. It is assumed that the word 'machine' refers to any manufacturing activity performed in the manufacturing facility.

The conditional event routines

SUBROUTINE BEGSET

Determines whether or not a Setting process can begin at a particular machine by continually testing the following four conditions for each of the workcentres ;

i) Check for waiting jobs in the workcentre queue

ii) Check for availability of a suitable machine to process the job.

iii) Check for availability of setter resource with correct skill

iv) Check for availability of operator resource with correct skill

If all the conditions are satisfied then the BEGSET routine can be executed. Appropriate changes are made to the machine, setter and operator attributes to reflect the fact that they are now tied up with a specific job. The appropriate machine attributes are also changed to show which job is currently being processed and what operation number this constitutes.

Once the next job for processing has been determined, the Pinnacle database is accessed in order to retrieve the set time allocated to that part number at the current machining operation. This data is used to schedule the time at which the setting process will end and the subsequent machining operation begin, since a job at this stage does not need to wait in a queue. The event scheduled for the end of the set time is BEGPRO.

SUBROUTINE BEGASY

Determines whether or not an assembly process can begin at a particular work area by continually testing the following three conditions for each of the workcentres ;

i) Check for waiting assembly jobs in the workcentre queue.

ii) Check for availability of a suitable work area to assemble the job.

iii) Check for availability of operator resource with correct skill

Notice that since there are no machines or processes to be organised in the assembly department of FCL, a setter resource is not required.

If all the conditions are satisfied then the BEGASY routine can be executed. Appropriate changes are made to the working area and operator attributes to reflect the fact that they are now tied up with a specific job (ie the next one in the queue). The appropriate work area attributes are also changed to show which job is currently being assembled and what operation number this constitutes.

Once the next job for assembly has been determined, the Pinnacle database is accessed in order to retrieve the assembly time allocated to that part number at the current work area. The total assembly time is calculated by multiplying The part assembly time with the transfer batch size. This data is used to schedule the time at which the assembly process will end. The event scheduled for the end of the assembly time is ENASSY.

The bound event routines

SUBROUTINE TRNSPT

This routine is responsible for all PCB job transportations between stores and workcentre and between each workcentre. The routine is initially scheduled for the time when each job is to be launched in to the system (this is also the method used to start off the simulation run). Subsequent scheduling of TRNSPT is done by the bound event, ENDPRO.

The batch entity to be transported is determined by interrogating information carried over from the particular event which scheduled the current TRNSPT. This information holds details about the previous workcentre number, part type and next operation number. The next workcentre number is retrieved from the Pinnacle database (PCB route details) using the part type and the next operation number for precise location of the data. The appropriate batch entity attribute is reset to this new workcentre for subsequent route information.

Based on this data, TRNSPT then ascertains whether or not the previous operation was the last one. If the previous workcentre indicates that this was the last operation, the batch is moved to the completed parts stores. If however, the job requires more operations, then the batch is transferred to the next workcentre queue and waits there until it can be processed. This will occur when all the conditions of BEGSET are satisfied for this particular job.

SUBROUTINE BEGPRO

The BEGPRO routine is responsible for representing the machine process for all PCB jobs. The batch entity to be processed is determined by interrogating information carried over from the scheduling event BEGSET. This information includes details about the next workcentre, and the machine which has just been set up for the current part type.

Appropriate changes are made to the machine, setter and operator attributes to reflect the fact that the current machine is in the state of processing a specific job.

The part process time is retrieved from the Pinnacle database (operation times) using the part type and the next operation number for precise location of the data. The total processing time is calculated by multiplying The part process time with the transfer batch size. This data is used to schedule the time at which the machine process will end. The event scheduled for the end of the assembly time is ENDPRO.

SUBROUTINE ENDPRO

At the completion of each process (ie when the simulated time reaches that of the total process time determined in BEGPRO) the event ENDPRO is executed.

Appropriate changes are made to the batch, machine and operator attributes to reflect the fact that the current machine is in the state of being idle and make resource available for future jobs.

The routine then schedules the event TRNSPT for a time in the future(PCB transport time) dictated by the PCB transport time parameter determined at the commencement of the simulation run.

SUBROUTINE ENASSY

At the completion of each assembly operation (ie when the simulated time reaches that of the total assembly time determined in BEGASY) the event ENASSY is executed.

Appropriate changes are made to the batch, work area and operator attributes to reflect the fact that the current work is in the state of being idle and make resource available for future assembly jobs.

The routine finally schedules the event ASTRAN for a time in the future(assembly transport time) dictated by the assembly transport time parameter determined at the commencement of the simulation run.

SUBROUTINE ASTRAN

This routine is responsible for all assembly transportations between stores and work area and between each work area. The routine is initially scheduled for the time when each job is to be launched in to the system. Subsequent scheduling of ASTRAN is done by the bound event, ENASSY.

The batch entity to be transported is determined by interrogating information carried over from the particular event which scheduled the current ASTRAN. This information holds details about the previous workcentre number, part type and next operation number. The next workcentre number is retrieved from the Pinnacle database (assembly route details) using the part type and the next operation number for precise location of the data. The appropriate batch entity attribute is reset to this new workcentre for subsequent route information.

Based on this data, ASTRAN then ascertains whether or not the previous operation was the last one. If the previous workcentre indicates that this was the last operation, the batch is moved to the completed part stores. If however, the job requires more operations, then the batch is transferred to the next workcentre queue and waits there until it can be processed. This will occur when all the conditions of BEGASY are satisfied for this particular job.

Simulation model procedure routines

The event ENDDAY in the shop model determines the activity of the model between simulated days by controlling the execution of a number of subroutines. The procedural flow of each of these various subroutines is shown in figure 8.4 The following describes this event and it's associated routines ;

SUBROUTINE ENDDAY

At the end of each simulated day this event is scheduled. It's primary purpose is to carry out all the appropriate procedures which occur at this time, namely ;

i) The Updating of all the appropriate stock levels and writing each modification to the disc on the Pinnacle using the routine called WIPCLS.

ii) The collection of all stock level statistics

iii) The reinitialization of all idle batch entities for future use in the model when creating new orders.

At this point ENDDAY checks if this is the end of the period as well as the end of the day. If the check is confirmed then the subroutine ENDPER is called. If however, the check results in a negative response, (ie end of day only), then the subroutine STTDAY is called. The later subroutine performs those activities which occur at the start of the day, ie;

i) Increment the day counter, and

ii) The launch of all assembly batches which are due for now and have sufficient component stocks.

SUBROUTINE ENDPER

Activities that occur at the end of each period are included in the subroutine ENDPER. This routine begins by reading all the completed stock information from the Pinnacle and uploading the appropriate part type with the completed quantity to the TMS model database using the subroutine, WIPOUT.

Each experimental period has a separate MPS change file associated with it. Each file is made up of; part type, period concerned and new MPS quantity. After successful upload of the stock data, the MPS is modified in the TMS model by uploading the appropriate file which contains all the changes which will have occurred during the period and up to the point of the next MRP run. This upload is achieved by invoking the subroutine, NXTMPS.

Any sales which are due to occur in each period are held in individual files. Each file holds the part types and quantities that are sold in each respective experimental period and are uploaded to the TMS model in the subroutine, SALES . This information is also used to reduce finished stock levels in the MM model as well as the TMS model.

SUBROUTINE STPER

The subroutine, STPER is called at the completion of an MRP run on the TMS model. All activities that occur at the commencement of each period are included in subroutine STPER. Before the communication routines are invoked the following tasks must be performed ;

i) Increment the period counter to the next period.

ii) Determine the number of days in the next period using the appropriate shop calendar. In this way the model takes into account public holidays and FCL shut downs.

iii) Schedule the event ENDDAY to occur in 480 minutes from now.

All the suggested orders and action messages are downloaded from TMS and written to specific periodic files using subroutines, WIPDWN and MSGDWN respectively.

All suggested orders are converted into planned orders and used to make up the manufacturing model's production plans for the appropriate departments. The subroutine, RUNBAT generates the production plan for the manufacturing department, whereas RUNSYS is used to generate the assembly production plan. Both plans are made up of batch entities.

These production plans are organised into launch date order and the appropriate scheduling of the events TRNSPT and ASTRAN for each planned order is done. Again all transportation times are dictated by a variable parameter which is determined at the beginning of each experimental run.

The action messages downloaded from TMS are used to delay and expedite suggested orders which have not yet been launched in the MM model. This rescheduling mechanism is incorporated in the subroutine, RSCHED.

General simulation model routines

SUBROUTINE RECORD

At specific points in time during the simulation run the bound event, RECORD is scheduled. This routine is responsible for collecting the statistical observations of the model. Each observation is time weighted in order to minimise any bias caused by early transient behaviour. The regularity of the event RECORD is determined by a variable parameter which is entered at the beginning of the experimental run. The following model elements are observed;

- i) Batch size
- ii) Shop floor WIP
- iii) Job flow time
- iv) Due date accuracy
- v) Workcentre utilisation
- vi) Workcentre input queue
- vii) Inventory stock

SUBROUTINE ENDED

The final routine to be executed in any experimental run is the bound event, ENDED. It is scheduled to occur at a specified time in the future by a variable parameter which is entered at the beginning of the experimental run. The routine performs statistical analysis of all the observed data collected by RECORD. The minimum, mean and maximum values of each data item are then written to a file on the Pinnacle and used as the basis for measuring the performance of the FCL model under various policy and parameter combinations.

EXPERIMENTAL PROCEDURE

Experimental overview

The FCL model has been designed so that experiments can be conducted under two different modes , namely :

1). Interactive experimental mode

2). Non-interactive experimental mode

Interactive experiments are those which involve communication with the B5900 during model run time. After a simulated period, completed WIP orders, product sale and MPS changes are used to update the TMS database. The MRP process is executed, after which time, suggested orders and actions are downloaded to the Pinnacle. The simulation model will not commence until all downloaded information has been received correctly.

Non-interactive experiments are those which involve no communication with the B5900 during model run time. This mode of operation does however assume that the periodic WIP orders and actions data files have been created by a previously run interactive experiment. These downloaded files are individually specified with the period number they represent at the time of creation .

The interactive approach is obviously a slower means of experimentation since each simulated period requires the execution of the MRP process. The approach does however, allow the whole range of possible control policy combinations to be evaluated. These policies are therefore variable at MPS, MRP and shop levels. With the non-interactive approach to experimentation the MPS and MRP control policies are predetermined and held fixed whilst the shop policies are allowed to vary. The time advantage gained relative to the interactive approach enables many more experiments to be conducted involving the analysis of how the shop floor performs to given outputs from TMS.

Setting up the Manufacturing simulator - Utility program

Before the commencement of any experiment a number of data files must exist in the designated disk area and with the correct field format. These files include;

```
E:PCB.DAT     -  PCB process and set times at each W/C
E:SYS.DAT     -  assembly process times at each W/C
E:SYSPIC.DAT -  product structure for each assembly
E:TRANSLT.DAT-  model to TMS partnumber translator
E:MPS*        -  contains periodic MPS changes
E:SALE*       -  contains periodic product sales
E:MPS00       -  contains the initial MPS demands
E:STKUP       -  contains all the system stock policies
```

These files can all be created and amended using the utility fortran program, 'FILE' . Running this program from the CPM/68K operating system on the Pinnacle will present the user with the following menu :

```
DATA FILE CREATION/AMENDMENT PROGRAM

B.... Board operation times , set-up times and routes
A.... Assembly/System operation times and routes
P.... Part pick-list creation
C.... File conversion routine(Direct to Sequential)
M.... MPS update files
S.... Product sales files
T.... Partnumber translator
U.... Stock parameter upload file
H.... Parameters for lead time calculation on TMS
E.... EXIT
```

This screen invites the user to select one of the above options.These options will now be described in the order in which they appear on the menu screen.

Board operation times , set-up times and routes

All operation times , set-up times and route details for PCBs are contained in the file named PCB.DAT. Before writing this file a certain amount of preparatory work is required . This involves identifying the route for each PCB, collecting MTM process times for each PCB at each Work station and estimating the set times for a given PCB at each work station. Each PCB represented by the model is written in PCB.DAT using the following format:

1st record .....Part model number
2nd record .....No. of operations

3rd record .....Work station No.,process time and set time

The last work station number for every PCB is '99'. This represents a dummy work centre and lets the simulation model know that the particular PCB in question has completed all the specified operations.

After selecting option B the user is guided through a question and answer session for each PCB. This takes the following form ;

NEXT PCB MODEL No. ==> 1

Are process times in Hours(Y/N)....

Here the user needs to decide whether the following times will be in hours or minutes. If the letter 'N' is selected minutes will be assumed.

Size of board 1=Small, 2=Medium & 3=Large....

At the time of writing there was no formal specification for work station set-up times . Since the model requires set-up time information a method of estimation this data was necessary. For each new database the PCBs are categorised as small , medium or large. These groupings are based on Physical PCB size and number of components per board. This then forms the basis of the work station set-time data. For example a PCB categorised as large may require an estimated set-up time of 50 minutes whereas a small PCB may only need 10 minutes at the same workcentre. These estimated times are stored and automatically written to the database.

NUMBER OF OPERATIONS(0 to end) ?

After letting the routine know the total number of operations required to produce the current PCB the work station number and process time of each operation is typed in.

DETAILS FOR OP No.

Work station number ?...........

Process time ?.................

The PCB database routine assumes that the first PCB model number will be '100'. Each subsequent PCB will automatically be identified by an integer number, greater than 100 in numerical order.

Assembly/System operation times and routes

All operation times and route details for products and assemblies are contained in the file named SYS.DAT.Again the part process times and routes details are are derived from existing sources. Each part represented by the model is written in SYS.DAT using the following format:

1st record .....Part model number and No. of operations
2nd record .....Work station No. and process time

The last work station number for every product or assembly is '99'. This represents a dummy work centre and lets the simulation model know that the particular part in question has completed all the specified operations.

After selecting option A the user is guided through a question and answer session for each part. This takes the following form ;

NEXT SYSTEM MODEL No. ==> 1

Are process times in Hours(Y/N)....

Here again the user needs to decide whether the following times will be in hours or minutes.

NUMBER OF OPERATIONS(0 to end) ?

After letting the routine know the total number of operations required to produce the current part the work station number and process time of each operation is typed in.

DETAILS FOR OP No.

Work station number ?..........

Process time ?................

This database routine assumes that the first part model number will be '1'. Each subsequent product or assembly will automatically be identified by an integer number, greater than 1 in numerical order up to and including '99'.

Part pick-list creation

The product structure of all products and assemblies are contained in the file named SYSPIC.DAT. Each structure represented by the model is written in SYSPIC.DAT using the following format:

1st record .....Part model number
2nd record .....Total number of components and total number of different components
3rd record .....Dummy(not used)
4th record .....Dummy(not used)
5th record .....Component model number and number off

After selecting option A the user is guided through a question and answer session for each part. This takes the following form ;

NEXT SYSTEM MODEL No. ==> 1

TOTAL NUMBER OF COMPONENTS OFF(0 to end) ?

NUMBER OF COMPONENTS ?

The user is first asked to input the component details for the part. This information can be taken from the part BOM specification on TMS. Following this initial information the model number and number off for each component is typed in using the following prompts ;

DETAILS FOR COMPONENT No. ==> 1

PCB/SUB-ASSEMBLY No ?...........

HOW MANY OF THESE ARE USED ?.....

File conversion routine(Direct to Sequential)

The files, when created are written using the direct access format as specified in the Fortran 77 manual. The selection of option C enables a previously created direct access file to be converted to a sequential access file. This is necessary since the manufacturing simulator will only read the later. After selecting option C the user is simply invited to specify the name of the direct access file for conversion and the new name of the sequential access file. The following prompts are presented;

CONVERT DIRECT ACCESS FILE (ie O:Filename) ?

TO SEQUENTIAL ACCESS FILE (ie O:Filename) ?

MPS update files

This option enables the user to build the 'MPS changes' database. Before creating the MPS change files a certain amount of preparatory work is required. A copy of a past MPS can be obtained from the planning department at FCL along with all the subsequent changes made to it over a sufficiently long horizon(at least 3 months) . At this stage it is necessary to decide what MPS policy will be used for the experiment. For example , grouping the given MPS will require manually modifying the periods in which demands are scheduled to be met. Another example of an MPS policy would be to exclude any forecast demands and only MPS those items which make up a definite customer order.

Once the MPS policy has been agreed the user is then ready to begin creating the MPS change files. The files used to store all the MPS changes over a given horizon are prefixed 'MPS'.Each period the experimental run requires a separate file. The complete filename therefore includes the period number. For example all the changes that occurred in period 6 would be contained in the file with the filename, 'MPS06'. One file record is used for each MPS change. Each record is made up as follows ;

Communication key ..... 3 characters
Part model number ..... 20 characters
Period number.......... 2 characters
New quantity .... 13 characters

After selecting option M the user is guided through a question and answer session for creating each MPS change file and typing in the required information . for each MPS change the user will respond to the following questions ;

NEXT MPS CHANGE ==> 1

MODEL PART No or (E)xit ? ....

PERIOD NUMBER ?..............................

QUANTITY ? ....

Product sales files

This option enables the user to build the 'Product sales' database. Before creating these files a certain amount preparatory work is required. A copy a past MPS can be obtained from the planning department at FCL along with all the subsequent sale that were made over a sufficiently long horizon(at least 3 months) .

The files used to store all the product sales over a given horizon are prefixed 'SALE'.Each period the experimental run requires a separate file. The complete filename therefore includes the period number. For example all the sales that occurred in period 22 would be contained in the file with the filename ,'SALE22'.One file record is used for each customer sale. Each record is made up as follows ;

Communication key ..... 3 characters
Part model number ..... 20 characters
Quantity sold .... 13 characters

After selecting option the user is guided through a question and answer session for creating each sales file and typing in the required information . For each customer sale the user will respond to the following questions ;

NEXT SALE ==> 1

MODEL PART No or (E)xit ? ....

QUANTITY SOLD ? ....

Partnumber translator

The manufacturing simulator defines each assembly and sub-assembly as an integer number. The TMS partnumber is however, made up alphanumeric characters. The file created by this routine, TRANSLT.DAT is used to translate between model number and TMS number. Each record has the following format ;

Simulation model number ..... 4 characters
TMS partnumber .... 20 characters

After selecting option T the user is presented with the following question and answer session for each part represented by the model ;

TYPE IN THE TMS NUMBER FOR MODEL No. >> 1

TMS PART No(C/R to end) ?...............

The routine assumes that the first TMS number typed in will represent model number '1', the second model number '2' and so on. The model numbers are automatically written to the database.

Stock parameter upload file

The file created by this routine is perhaps one the most important ones as far as experimental design is concerned, since it contains all the TMS stock control policies and parameters. It allows the following policies to be determined for each part represented in the model ;

1). Minimum stock level
2). Lead time
3). Order policy
4). Pan size
5). EOQ/Time bucket
6). Minimum batch size

The file created is called STKUP.DAT.Each record represents the policies for each part and has the following format ;

Communication key ..... 3 characters
TMS partnumber .... 20 characters
Minimum stock .... 11 characters
Lead time .... 4 characters
Order policy .... 1 character
Pan size .... 7 characters
EOQ/Time bucket .... 11 characters
Minimum Batch size .... 11 characters

After selecting option U the user is presented with the following question and answer session for each part represented by the model ;

NEXT STOCK PARAMETERS FOR PART ==> 1

Minimum Stock level (-ive to end) ?.....

Lead time ? ....

Order policy (0-6) ?....................

Pan size ?..............................

EOQ/Time bucket ?.......................

Minimum batch size ?...................

The file STKUP.DAT is automatically uploaded to the B5900 by running the program, PARMUP on the pinnacle. Once loaded in to TMS (under a file called TMSASP/STKPARM) the records are used to update the TMS stock parameters.

Parameters for lead time calculation on TMS

The routine under option H gives the user the opportunity to calculate the lead time for each part number based on the following linear lead time policy formula ;

Lead time = total process time * process batch size
+ average transport time * No. operations-1
+ total machine set-up time
+ average queue time * No. operations

The file created is called TMSHRS and is used as an option when running the routine to create the stock parameter file,STKUP.DAT. The arguments for the above formula are selected using the following question and answer session ;

PCB Queue Element (days) ?.....

SYS Queue Element (days) ?.....

PCB Transport time (mins) ?.......

SYS Transport time (mins) ?.......

PCB mean quantity ?...............

SYS mean quantity ?...............

Exit from the file utility

Finally option E is used to exit the program, FILE.

Experiment parameter file

The final file which needs to be created is EX10, which holds the main experimental parameters for the manufacturing simulator. These are ;

Load Factor	Artificially modifies production plan in terms order size.
PCB Trans time ..	Average time for PCB to move between W/Cs
SYS Trans time ..	Average time for SYS to move between W/Cs
PCB Trans batch .	Cut off value for PCB process batch size
SYS Trans batch .	Cut off value for SYS process batch size
Priority No......	Number to determine priority rule (not in use)
Experiment len...	Number that the experiment will run

This file is simply created in the word processor ,Mince with record holding the above parameters. Each of these parameters has a field width of 9 characters.

Setting up the Test data base

The test database replicates the A9 database down to PCB lev Initial analysis showed that 90% of what was live at the time the study was undertaken would be sufficient for our purposes. The TMS model only requires four modules. These are ;

1). Engineering Data Base (EDC)
2). Materials Requirement Planning (MRP)
3). Stock Control (STK)
4). Work in progress (WIP)

The EDC module has Three main purposes in the TMS model .These are

i) Part definition
ii) Bill materials (BOM) specification
iii) Shop calendar maintenance

The MRP module has three main purposes in the TMS model .These are

i) The MRP calculation process
ii) Setting up the the MPS
iii) Definition the planning horizon

The STK module has three main purposes in the TMS model .These are

i) control stock allocations
ii) Setting up the part stock parameters
iii) Inventory control (stock status updating)

The major WIP module functions such as production plan creation and shop floor activity monitoring are provided by the manufacturing simulator. Therefore the only purpose served by the WIP module is the allocation of component stock.

Running an interactive experiment

It is assumed at this stage that both the manufacturing simulator and the TMS module have been set up according to the guide lines given in this user guide. In addition, for each new experiment, the test database must be reloaded from the backup disk on TMS. It is suggested that the user consults with the operations manager at FCL before this procedure is attempted.

Since an MRP run is required before the first manufacturing period can be simulated it is suggested that the TMS model cycle is started first. All programs and workflows are run under CANDE. Access to a CANDE session can be obtained by transmitting the following statements at the top left hand corner a TMS compatible terminal ;

CANDE

HELL TEST/TEST

ASP/PINNACLE

A successful 'log on' is completed when the system returns a work session number.From now on all statements and commands are under CANDE control.

Two TMS database updates are necessary before we actually run the TMS model cycle workflow, these represent ; the creation of the initial MPS and the setting up of the stock parameters and policies.

The initial MPS demands are created by transmitting the follo statement ;

ST RUN/ASTONMPS

This workflow uses the records in the previously created f: TMSASP/MPSUP to update the new MPS.

The new stock parameters which are stored in the file, TMSASP/S are updated similarly by transmitting the following statement ;

ST RUN/ASTONSTK

When the previous workflows have finished the TMS model cycle then be run by issuing the following statement ;

ST RUN/ASTONMRP

A continual assessment of the job summary will be displayed on the terminal until the model run is complete. This stage will be indicated by the last displayed statement which will declare ;

Aston experiment has completed OK !

We can now concentrate on starting the manufacturing simulator cy Before the program run however make sure the following steps been taken to ensure a successful model run ;

1). Check that the is printer on , online and has sufficient paper.

2). Check that the P1000 is on and has been reset.

3). Check the communications link by running the fortran program, FIX which allows statements to be transmitted and received through the P1000. Sending the command, 'WRU' should return a CANDE session number.

4). Erase all files on disk F because as the model progresses the model status at given periods will be saved on F:. This precaution is taken in case there is a need to restart the experiment.

The manufacturing simulation cycle is started by running the program, CBOTT on the pinnacle, by issuing the following statement from disk E;

```
G:CBOTT
```

After the simulation initialisation the user will be given the following prompt ;

Comms are ON: 1=ON, 0=OFF ?

This informs the user that the communications link is currently switched on. At this stage it possible to switch the communications off. For our purposes the response to this prompt will be '1' which will allow the model to run interactively(for non-interactive experiments respond with '0').

The manufacturing simulator will run until the experiment completion time(as specified in the parameter file, EX10). At the end of the experimental run control will be directed back to the operating system.

LIST OF REFERENCES

AUTHOR

YEAR

DESCRIPTION

Anshoff,H.I and Hayes,L.R

1972

"Role of models in corporate decision making"

Proceedings of IFORS 6th International Conference

Dublin, Ireland.

Aggarwal,D, Wyman,P and McCarl,B,A.

1973

"An investigation of a cost based rule for job shop scheduling"

International Journal of Production Research. 11:3pp241-261

Arumugan,V

1985

"Priority sequencing in a real world job shop"

School of Engineering, Federal University of Technology, Owerri, Nigeria

Aucamp,D.C

1984

"A caveat on using set-up costs in lot-sizing"

Southern Illinois University , Ewardsville,IL

Black,M

1962

"Models and Metaphors:Studies in Language an Philosophy"

Cornell University Press , Ithaca, NY.

AUTHOR

YEAR

DESCRIPTION

Blackburn, J.D. and Millen, R. A.

1979

"Improved Heuristics for Multi-stage Requirements Planning Systems."

Management Science, Vol.28 No.1 pp.44-56 1982.

Blackburn,J.D and Millen,R.A

1982

"The impact of a rolling schedule in a multi-level MRP system"

Journal of Operations Management Vol. 2, No. 2

Blackburn, J.H.Jnr.

1984

"Considerations in Choosing Shop Scheduling Algorithms." Production And Inventory Management, 4th quarter 1984, pp.107-125.

Blackburn, J.D. Krop, D.H. and Millen, R.A.

1985

"MRP System Nervousness: Causes and Cures." Engineering Costs and Production Economics. 9, pp.141-146

Bobko. P. B. and D. C. Whybark.

1985

"The Coefficient of Variation as a Factor in MRP Research.' Decision Sciences, Vol. 16, pp. 420 - 427. 1985.

Bott, K.N.

1981

"An Analysis of Factors Which Affect Load Variability and System Performance in a Multi-stage, Multi-Product Production System." Unpublished Doctoral Thesis, The Ohio State University.

AUTHOR

YEAR

DESCRIPTION

Blackburn,J.D and Kropp,D.H and Millen,R.A

1985

"MRP system nervousness:causes and cures"

Engineering Costs and Production Economics,vol. 9

Bott,K.N

1981

"Analysis of factor which affect load variability and system performance in a multistage , multi-product production system"

OH St University USA , diss.

Bridge,K

1987

"The development of a manufacturing design procedure"

A 1st year report for Phd. Aston University, UK.

Carlson, R.C., Jucker, J.V. and Kropp, D.H.

1979

"Less Nervous MRP Systems: A Dynamic Lot Sizing Approach," Management Science, Vol. 25, No. 8 (1979), pp. 754-761 .

Churchman,C,W

1957

"Introduction to operations research"

Wiley , New York .

Clarke,S.R

1988

"Aspects of the design and control of manufacturing systems subject to demand uncertainty"

Phd thesis, Aston University , Birmingham

AUTHOR

YEAR

DESCRIPTION

Clementson,A.T.

1982

"Extended Control Language and Simulation Language"

Cle. Com Ltd., Birmingham

Conway,R.W , Johnson,B.M , Maxwell,W.L

1959

"Some problems of digital systems simulation"

Management Science , vol. 6

Conway,R.W and Maxwell,W.L

1964

"Experimental investigation of priority assignment in a job shop"

Rand Corporation Memorandum.

Coyle,R.G.

1977

"Management System Dynamics"

Willey Interscience, London.

De Bodt, M.A. and Van Wassenhove, L. N.

1980

"Lot Sizes and Safety Stocks in MRP: A Case Study," Werkdokument 80-24, Industrial Beleid, Katholiek University, Leuven. 1980.

AUTHOR

YEAR

DESCRIPTION

De Bodt, M.A., Van Wassenhove, L.N. and Gelders, L.F.

1982

"Lot Sizing and Safety Stock Decisions in an MRP System With Demand Uncertainty," Engineering Costs and Production Economics, Vol. 6. pp. 67-75.

De Bodt,M.A and Van Wassenhove,L.N

1983

"Lot sizes and safety stocks in MRP :A case study"

Catholic University of London , Leuven.

Fisherman,G.S and Kiviat,P.J

1967

"DigitalComputer Simulation:Statistical Considerations"

Rand Corporation , Santa Monica , CA.

Forrester, J.W.

1961

"Industrial Dynamics", Published by MIT Press.

Forrester, J.W.

1971

"World Dynamics".

Goldratt, E.M.

1981

"The Unbalanced Plant"

Proc. APICS Annual Conf., 1981 pp. 36-51.

AUTHOR

YEAR

DESCRIPTION

Goodman,M.R

1974

"Study notes in system dynamics"

Wright-Allen, MA.

Gordon,G

1978

"System Simulation"

2nd edition, Prentis Hall,NJ.

Hermann,E.J

1967

"Validation problems in games and simulation"

Behavioural Sciences, vol. 12

Henriksen,J.O

1979

"The GPSS/H User's manual"

Wolverine software, Falls Church, Va.

Hoo-Gon Choi, Malstrum , E.M. and Classen, R.J

1984

"Computer simulation of lot sizing algorithms in the three-stage Multi-Echalon inventory system."

Journal of Operations Management , Vol 4 No. 8 pp 259-276

Hurrion,R.D

1977

"Visual interactive simulation-an aid to decision making"

C.I.E.B.R. discussion paper 76 , University of Warwick

AUTHOR

YEAR

DESCRIPTION

IBM Corp.

1967

"General Purpose Simulation System"

New York.

Kaimann,R.A

1969

"EOQ vs. dynamic programming - which one to use for inventory ordering"

Production and Inventory management, 10(4), pp66-74

Kanet,J.J CPIM

1982

"Towards understanding lead times in MRP systems"

The University of Georgia, Athens, GA

Keviat,P.J

1967

"GASPII: A FORTRAN based simulation language"

Industrial Engineering dept.,Arizona State University.

Kropp, D.H,Carlson,R.C and Jucker,J.V

1983

"Heuristic lot-sizing approaches for dealing with MRP system nervousness" .

Decision Sciences , vol. 14 , No.2

AUTHOR
YEAR
DESCRIPTION

Lee,T.S
1983
"Operations forecasting error evaluation in MRP production inventory systems"
University of Missouri, Columbia, USA ,Diss.

Love, D.M.
1980
"Aspects of Design for a Spare Parts Provisioning System".
Ph.D. Thesis, University of Aston in Birmingham, UK.

Mc Claren, B.J.
1977
"A Study of Multiple Level Lot Sizing Procedures for Materials Requirements Planning Systems,"
Ph.D Dissertation, Purdue University .

Melnyk, S.A.
1980
"Manufacturing Lead Time Estimation and the Implementation of an MRP system: A Computer Simulation Study,"
Unpublished Dissertation. University of Western Ontario.

Melnyk,S.A and Gonzalez,R.F
1985
"MRPII: The early returns are in"
Michigan State University , East Lansing, MI

AUTHOR

YEAR

DESCRIPTION

Minifie, J.Roberta and Heard, E.

1985

"On The Generalizability of MRP Simulation Results," Engineering Costs and Production Economics, Vol. 9 (1985) pp. 211-217.

Mihram,G.A

1971

"Simulation : Statistical foundation and methodology" Academic press , New York .

Mize, J.H.

1979,

"Simulation Modelling" Proceedings of the Industrial Engineering Conference, Fall 1979.

Muth,E.J

1977

"Transformation methods with applications to engineering and operations research"
Englewood Cliffs: Prentice Hall.

Nandakumar, G,

1985

"Lot-Sizing Techniques In A Mulyiproduct Multilevel Environment."
Production and Inventory Management, 1st quarter, 1985. pp.16-54.

AUTHOR

YEAR

DESCRIPTION

Naylor,T.H and Finger,J.M

1967

"Verification of computer simulation models"

Management Science , vol.14

Orlicky, J.

1975

"Materials Requirements Planning,"

McGraw-Hill, New York.

Panwalker, S.S. and Iskander, W.

1977

"A Survey of Scheduling Rules"

Operations Research, Vol. 25, No 1,

Pegles,C.C

1969

"Simulation and the optimal design of the production process"

International Journal of Production Research, Vol 7, No. 3

Pegels, C.C. and Naryan, P.S.

1976,

"Simulation as a Planning Technique for a Complex Machine Shop",

International Journal of Production Research, 1976, Vol 14, No 3, pp 387-400.

AUTHOR

YEAR

DESCRIPTION

Phillips,D.T

1977

"A Generalised manufacturing simulator"

Industrial Engineering Dept., Texas A & M University.

Pidd M.

1984

"Computer Simulation In Management Science."

John Wiley & Sons.

Popper,K.R.

1959

"The Logic of Scientific Discovery" .

Basic Books , New York .

Pritsker,A.A.B

1974

"The GASP IV Simulation Language"

New York.

Raiszadeh,M.E

1982

"Experimental evaluation of the performance of lot size strategies in a MRP system facing uncertain demand environment with different operating conditions"

Louisiana State University , USA , Dissertation.

Rapoport,A

1953

"Operational Philosophy"

Harper and Brothers , New York.

AUTHOR

YEAR

DESCRIPTION

Reichenbach,H

1951

"The rise of scientific philosophy"

University of California Press, Berkeley, US.

Rogers, S.D., Griffin, T., Ganate, D.P. and Davis, B.

1980

"Ongoing Online Production Simulation"

Proceedings of the 1980 Spring Annual Conference on Simulation.

Shannon,R.E

1975

"System Simulation"

Prentice-Hall,Inc., N.J.

Shubik,M

1960

"Simulation of the industry and the firm"

American Economic Review,50.

Spooner,P.D

1985

"A Simulation based interactive production control system"

Proceedings of the 1st International Conference on Simulation in Manufacturing,UK., pp65-73.

AUTHOR

YEAR

DESCRIPTION

St John, R.E.

1983

"The Cost Effects Of Inflated Planned Manufacturing Lead Times, In Materials Requirements Planning Systems" unpublished doctoral dissertation, Arizona State University.

St John, R.

1984

"The Evils Of Lot Sizing In MRP."
Production and Inventory Management, 4th Quarter pp 75-85.

St. John, R.

1985

"The Cost of Inflated Planned Lead Times in MRP Systems"
Journal of Operations Management, Vol.5, No.2, Feb 1985, pp. 119-128.

Tabata,K, Mori,K, Mitsumori,S, Ohshima,K, Ono,H and Inoue,K

1977

"PASS: An interactive online simulator for predictive production control"
Information control problems in manufacturing technology Conference, Tokyo.

AUTHOR

YEAR

DESCRIPTION

Tocher,K.D.

1963

"The art of simulation"

English University Press London.

Van Horn,R

1969

"Validation in the design of computer simulation experiments"

Duke University Press , Durham , pp 232-251 .

Van Horn,R

1971

"Validation of simulation results"

Management Science vol 17 , No 5 January , USA

Wagner, H.M. and Whitin T.

1985

"Dynamic Version of the Economic Lot-sizing Model"

Management Science, Vol. 5, No. 1 (1985), pp. 89-86.

Wemmerlov U.

1979

"Design Factors in MRP Systems: A Limited Survey"

Production and Inventory Management, 4th quarter pp. 15-35.

AUTHOR

YEAR

DESCRIPTION

Wemmerlov,U

1982

"A comparison of discrete, single state lot sizing heuristics with special emphasis on rules based on the marginal cost principle"

Engineering Cost and Production Economics, 7 pp 45-58

Wemmerlov U.

1983

"Comments On A Near Optimum Lot Sizing Model"

Production And Inventory Management, 3rd quarter pp.117-121.

Wemmerlov,U

1985

"Comments on 'Cost increases due to demand uncertainty in MRP lot sizing': A verification of ordering probabilities"

Decision Sciences , vol 16 , pp 410-419 .

Whybark,D.C and Williams,J.G

1976

"Material requirement planning under uncertainty"

Decision Sciences, 7, 595-606.