DESIGN & DEVELOPMENT OF A SIMULATION MODEL TO ANALYSE SCHEDULING RULES IN AN FMS IN A VIRTUAL MANUFACTURING ENVIRONMENT

A thesis presented in partial fulfilment of the requirements for the degree of Master of Technology in Manufacturing and Industrial Technology at Massey University

Ajay Bhushan Talla

1997

ABSTRACT

Due to the rapid changes in the needs of the customer for new products, the future manufacturing systems must cope with these changes. Hence, the need for the manufacturing systems to support these changes in the products with shorter lead times within a single manufacturing facility. The Virtual Manufacturing System (VMS) is one concept which can assist in meeting these demands. The VMS concept enables the manufacturing system designers to emulate and test the performance of the future manufacturing systems.

This research has given an overview of the new concepts of Virtual Manufacturing Systems and Virtual Manufacturing in general. A Virtual Reality Software tool has been used to realise the VMS concept. A Virtual Manufacturing Environment representing a Flexible Manufacturing System (FMS) has been modelled. A simulation control language is employed for developing simulation control logics and decision making control logics for the development of the FMS model.

The modelled FMS is implemented and tested through simulation experiments. The testing is done by analysing the traditional scheduling rules in a manufacturing facility. Average Machine Utilisation, Mean Flow Time, Average Queue Lengths and the System Production Rate are measured as the System Performance Measures for the evaluation of the scheduling rules.

This research has identified that the Virtual Manufacturing Software is a powerful tool which can identify optimum configurations and highlight potential problems before a final and expensive manufacturing system is established physically.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Dr. Saeid Nahavandi for his valuable suggestions and guidance during the progress of my research and the successful completion of the thesis. I would also like to express thanks to Prof. Don Barnes for his guidance during the write-up of the thesis.

My thanks to all the staff members and fellow postgraduates in the department of Production Technology for their help and support during the course of my research.

Also my thanks to my friends for providing me with lighter moments in my hour of stress.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

This chapter gives an introduction to the concept and definition of Virtual Manufacturing System (VMS) and the technologies related to VMS. It elaborates the scope of the research and research objectives. It also includes an overview of this thesis.

1.1 Concept of Virtual Manufacturing System (VMS)

In this research, the concept of Virtual manufacturing systems (VMS) is based on the concept developed by Masahiko Onosato et. al. (1993) which is explained as follows:

A manufacturing system in the real world could be decomposed into two different subsystems :

A real and physical system (RPS)
A real and informational system (RIS)

An RPS is composed of real entities that exist in real world. Entities such as, materials, parts, products, machine tools, jigs, sensors, controllers, and so on. When a manufacturing system is in operation, these components of the manufacturing system have physical behaviours and interactions such as motion, transfer and transformation.

An RIS involves many activities of information processing and decision making such as, designing, planning, scheduling, controlling, estimating and so on. It consists of computer activities as well as human activities of a real manufacturing system. Consider a real manufacturing system made up of an RPS and an RIS. Activities in the RIS are physically separated from the substantial entities in the RPS, and the RIS has relation to the RPS by exchanging information. The RPS sends status reports to the RIS by means of sensors, data terminals, and other communication channels. Control commands generated by the RIS are sent to the RPS so as to operate machines there. The RIS can effect and know the RPS only by means of control commands and the status reports respectively.

1.2 Definition of VMS

If all the activities of the RPS can be represented in a computer system that interprets control commands and returns status reports from and to the RIS, and if the responses from the system are equivalent to the RPS, then no one in the RIS would know whether the status reports are returned from the RPS or from the computer system. A computer system that simulates the responses of the RPS is called "Virtual and Physical System" (VPS).

Similarly if there is a computer system that can simulate the functions of the RIS in such a way that the machines in the RPS cannot distinguish whether control commands are coming from the RIS or the computer system. Then such a computer system which simulates an RIS and generates control commands for the RPS is called "Virtual and Informational System" (VIS)

The following four manufacturing system types can be formulated by having different combinations of the physical and informational systems :

1) RPS + RIS :	Real manufacturing systems
2) RPS + VIS :	Automated manufacturing systems
3) VPS + RIS :	Virtual manufacturing systems (physically)
4) VPS + VIS :	Virtual manufacturing systems (fully)

This research is based on the manufacturing system of type 4.

1.3 Technologies related to VMS

According to the various manufacturing systems as described above, CIM (Computer Integrated Manufacturing) is including the activities concerning computerisation and integration of RISs in real manufacturing systems. When an RIS is fully computerised and integrated by CIM, it is also considered as a VIS.

A Flexible Manufacturing System(FMS), which will be explained in detail in chapter 3, corresponds to an RPS. Since machines in FMSs can be highly automated by computers and have good controllability, it is easier to realise a VPS for FMSs than that for non-computerised manufacturing systems.

Virtual Reality (VR) is now attracting much attention of researchers and industries. Virtual reality is a technology by which a user can observe or operate objects in computers as if they were in the real world. So, VR is closely related to VPSs.

1.4 Scope of the Research

The basic objective of VMS is to model and simulate a VPS and a VIS and establish a communication link to the RPS and the RIS. Since this covers all the major areas of a manufacturing system, the scope of this research is limited to the development of a Simulation Model in a Virtual Manufacturing Environment, which emulates a Flexible Manufacturing System model i.e., VPS as mentioned in the above sections, which is explained in detail in Chapters 4 and 5. Later on this model is tested for the evaluation of scheduling rules in an FMS. As Virtual Reality is closely related to VPS, a Virtual Reality Software system is used in this research to achieve the research objectives.

1.5 Research Objectives

The objectives of this research include :

- Development of a simulation model of the Manufacturing Pilot Plant and the Industrial Engineering Workshop of the Department of Production Technology using Virtual Reality Software. The Manufacturing Pilot Plant is involved in manufacturing printed circuit boards and the Industrial Engineering Workshop produces plastic components and the injection moulds for those components.
- Development of a Simulation Model of an experimental FMS in a Virtual Manufacturing Environment using Virtual Reality Software.
- Testing of the model by analysing traditional scheduling rules in a manufacturing facility. This is done by analysing the effect of the scheduling rules on the System Performance. The system performance indices set for this research are, Mean Flow Time of Jobs, Average Machine Utilisation, Average Queue Lengths and the System Production Rate.

1.6 Overview of the Thesis

Chapter 2 covers an overview of the recent trends in Virtual Manufacturing and the related literature. Chapter 3 gives details about Flexible Manufacturing Systems (FMS), Scheduling of FMS and relevant literature, Chapter 4 explains in detail the methodology adopted for this research in achieving the research objectives. Chapter 5 covers the simulation experiments conducted in this research for analysing the scheduling rules in a manufacturing facility. Chapter 6 discusses the simulation results and the analysis of the results and finally Chapter 7 draws the conclusions from this research.

Chapter 2

RECENT TRENDS IN VIRTUAL MANUFACTURING

This chapter covers the concept of Virtual Manufacturing, the background to the research in Virtual Manufacturing & relevant literature, the vision of Virtual Manufacturing, and elaborates the benefits of Virtual Manufacturing and how it can be used. The Chapter concludes by giving an overview of the various applications of Virtual Reality in Engineering.

2.1 Virtual Manufacturing

The definition of Virtual Manufacturing (VM) by various researchers is as follows :

VM is an approach to effective modelling of necessary product behaviour and associated manufacturing processes by computer as precisely as possible, and to predict potential problems for product functionality and manufacturability before undertaking real manufacturing (Fumihiko Kimura, 1993). It is an approach to construct future generation manufacturing systems based on thorough modelling and simulation of all the necessary manufacturing activities by computer support technology.

VM is the use of desktop virtual reality techniques to develop a system which aids in computer-aided design of components and processes for manufacture (Gary Bayliss et. al.,1994). It offers excellent scope for creating and viewing three-dimensional engineering models, later to be passed to numerically-controlled machines for real manufacturing.

2.2 Research Background in Virtual Manufacturing

Virtual Manufacturing as expressed by Alex Pentland et.al. (1989), is the close integration of manufacturing, analysis and design. Most often it is observed that the design engineer designs the engineering components without quite knowing the manufacturing and the analysis problems for the components. VM provides the design engineer with all the manufacturing and analysis information at the initial design stage, thus giving the design engineer the flexibility to design more producible designs. For example, if the designer intends to generate a conceptual design of the engineering components, VM technology helps in testing the product functionality and testing/conducting the finite-element analysis at the initial design stage.

As it is becoming vitally important, in highly industrialised countries, to realise timely production of value-added high quality products in quick response to market requirements. Manufacturing systems are forced to cope with a small sized production of many variety of products. In such a situation, it is strongly required to achieve a tight integration of all the product manufacturing activities, such as product design, process planning, product planning and manufacturing preparation, etc.

The current design process for manufacturing systems development involves the development of multiple models of the eventual manufacturing system. Charles et al., (1994), proposed a new paradigm called Virtual Manufacturing Systems Engineering which integrates independent domain specific design tools into an integrated system for the design of large and complex manufacturing systems.

Virtual Manufacturing systems are integrated computer models that represent the precise and whole structure of manufacturing systems and simulate their physical and logical behaviour in operation (K. Iwata et. al., 1995)

Manufacturing technology has been well developed, but new requirements have appeared, which can be summarised as follows:

- *Flexibility for the change:* Market needs, products/production technology, human requirements and social environments etc. are rapidly and unexpectedly changing. Hence, traditional manufacturing is required to adapt to those changes efficiently.

- *Human centered organisation:* Instead of eliminating human to achieve automation, augmentation of human power and human integration are required to achieve high quality manufacturing.

- *Clean manufacturing:* The effects of manufacturing on the environment cannot be overlooked, and sound technology should be developed to measure such effect and to effectively control it.

Computer aided technology has been already widely used in every industry and it has become an indispensable tool for companies to keep their competitiveness in highly developed industrial environment. However, it is still not satisfactory for coping with the above requirements. Modelling capability of products and manufacturing processes is not comprehensive and prediction of product and process behaviour is difficult in unexpected situations. Systems are normally very complicated and not well modularised , and are difficult to be extended according to the change of requirements. Computer internal processing is not transparent to human users and it is not easy to effectively intervene when something wrong happens.

To overcome those deficiencies and to use computer power more effectively the following approach was proposed: (Kimura, 1993)

- systematic organisation of manufacturing knowledge based on relevant theory and accumulated knowledge.
- comprehensive modelling of engineering objects based on the above analysis.

- evaluation of design and manufacturing activities based on precise computer simulation before making actual manufacturing.
- elimination of inappropriate results by the above evaluation.
- maintenance of models in daily operation to achieve high-quality simulation

The above approach is collectively termed as Virtual Manufacturing. In this approach modelling of product behaviour and associated manufacturing processes plays a central role. Activity modelling is another key area for modelling. Activity models are to represent various activities, whether human or by computer, for product or production engineering. Compared with product and process models, activity models are not yet deeply investigated.

The Virtual Manufacturing approach is very effective for the above stated new requirements of manufacturing technology. It allows easy re-configuration and extension of a system by offering systematic and modularised knowledge of manufacturing, and its computer support activities are transparent for human understanding. By virtue of precise engineering verification, manufacturing side-effects can be predicted. Such model-based approaches have been pursued for many years under names such as geometric modelling, product modelling and process modelling etc. The individual modelling techniques have been used for specific application areas.

Integrated models have been also considered, but in many cases model construction is rather ad hoc, and are not generally applicable. By virtual manufacturing, it is possible to realise a comprehensive model representation framework by which all the models can be integrated as shown in fig. 2.1, and can be consistently manipulated. For instance, dynamic models can be related with shape models to get necessary shape data and to make object collision analysis. Models should be formalised in some form of computer software for actual usage. In order to realise flexibility required for future manufacturing systems, some kind of modular software architecture is essential. Constituent modules should be independent as shown in fig. 2.2, and be able to communicate with each other and these modules should be in a framework which is hierarchical in nature.

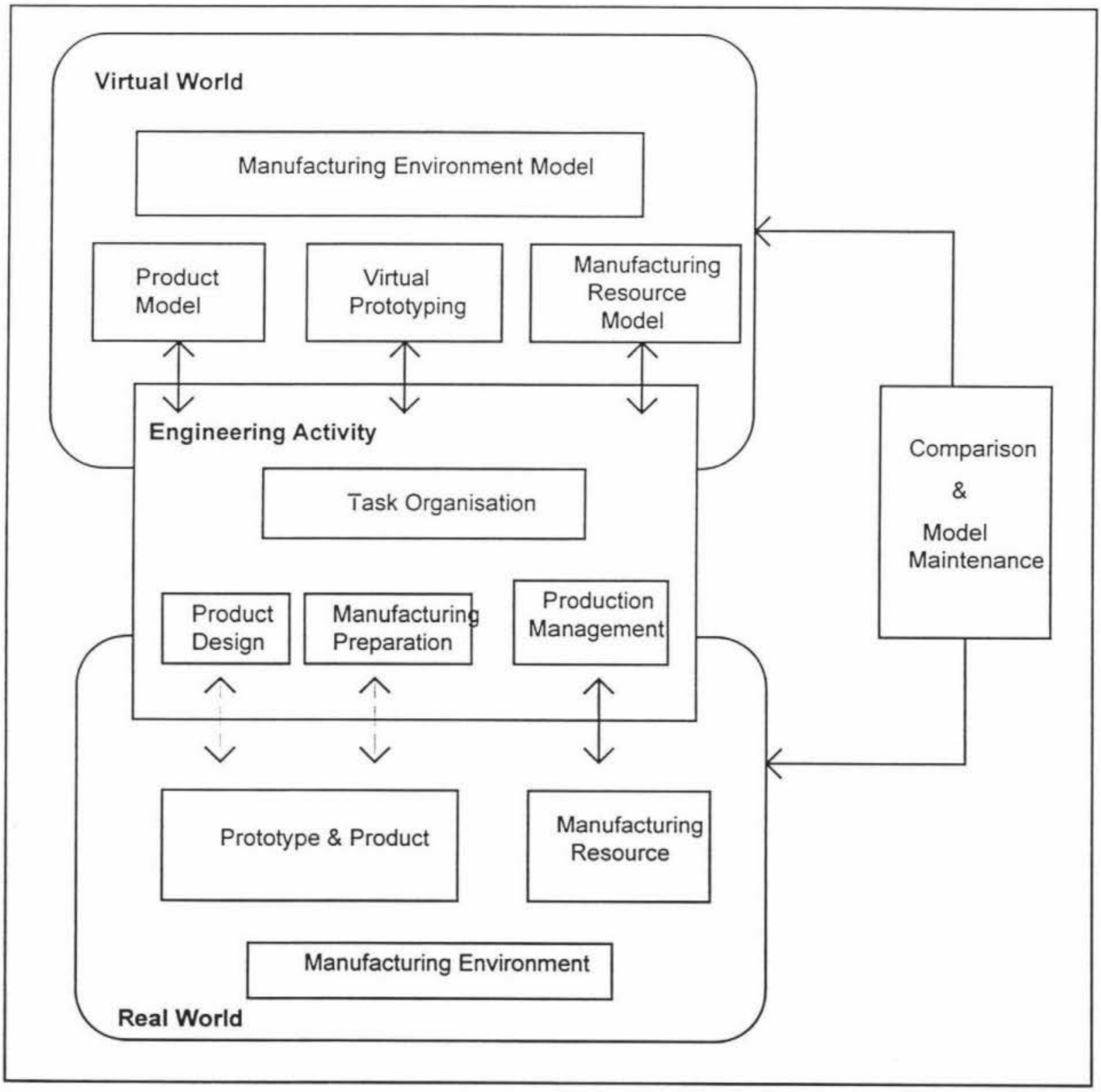


Fig. 2.1 Total Manufacturing Integration by Virtual Manufacturing Concept (Kimura, 1993)

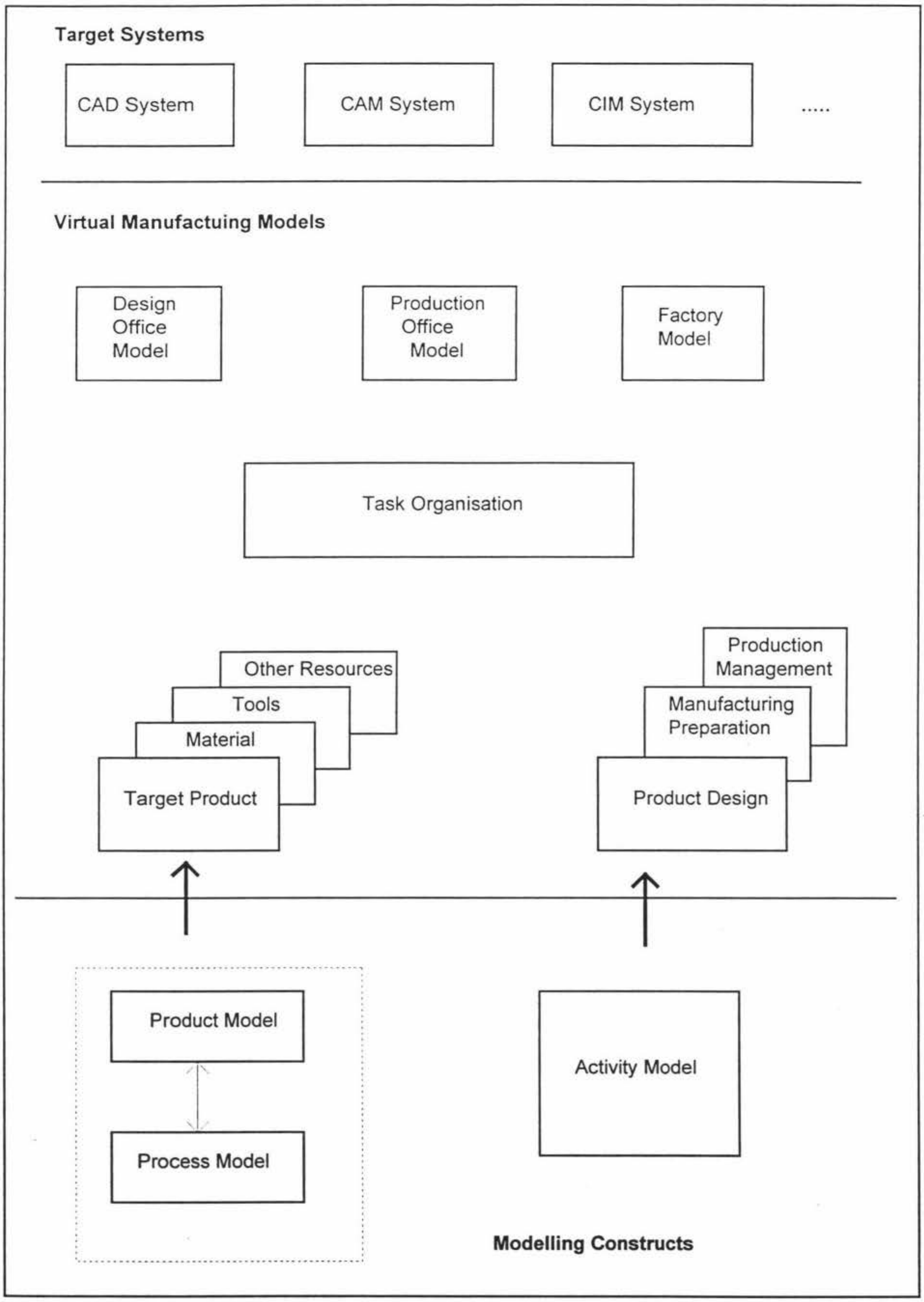


Fig. 2.2 Modelling Hierarchy for Virtual Manufacturing (Kimura, 1993)

Research and development of Virtual Interface Technology has been underway for the past 2-3 decades. Sudershan Jetley et. al. (1994), have explored the potential advantages and limitations of Virtual Interface Technology in manufacturing and related disciplines.

Virtual Reality (VR) is an advanced technology of spatial simulation of reality generated by a computer. A virtual world simulates the real world so that to the human senses, there is no distinguishable difference between the real and the simulation. Virtual Interface technology is a newly emerging and rapidly growing field.

Computer Aided Design and Manufacturing (CAD/CAM) can be simply defined as the use of computers to aid the design and manufacturing processes. CAD can be said to have two components, namely graphics and analysis. Its two broad applications are in the design of manufactured products and manufacturing processes. CAD process involves creating geometry and then performing functional analysis. It is possible to perform these functions using separate software packages or one comprehensive package such as CATIA, that has separate modules for these functions. Other examples of this software are the IDEAS software for product design and QUEST® for process design (Sudershan Jetley et. al., 1994).

Even the widespread adoption of computer aided design and analysis has not eliminated the need for prototypes. In fact, engineers today are producing more complex designs using computer aided engineering that need prototypes for validation. In most cases, prototyping becomes time consuming but the use of new rapid prototyping methods like stereolithography are helping to alleviate this problem. To remain competitive in today's market industry needs to shorten their product development cycle. This leads to the situation for the elimination of prototypes and this could possibly be achieved by VR.

Three-dimensional virtual interfaces can link humans to the CAD/CAM systems. Using these, instead of being forced to remain on the outside of the design looking at the visual monitors, it would be possible for the engineer to "move inside the design" and feel surrounded by the world or object which is being designed. The designer will be able to reach out and touch the virtual objects, which are actually illusions fabricated by a computer. The designer could move or change such as he or she reaches out to them. If such a virtual design was applied in the automotive industry for example, it would be possible to go inside a virtual automobile and design the instrument panel and drive the car before it was actually made. Once the design is completed and optimised in the virtual environment, other software applications would take over to create the database for geometry and manufacture of parts. Such computer models can also predict the performance of a proposed design accurately when combined with a comprehensive materials database.

Thus, today it is a challenge for industry to exploit this VR technology to further improve productivity and quality, as well as minimise adverse affects on the environment and working conditions. Virtual Manufacturing brings new concept to the table, even if they are not usable at the moment. VM provides the potential of ultimate flexibility in design and experimentation and therefore the means to solve many complex problems and cause shifts in many a paradigm or traditional thinking.

Virtual manufacturing software lets production engineers create life-like, full-action mock-ups of automated production systems on computer workstations, and then analyse and debug them before investing in capital equipment. Many steps in preparing for automated production - from designing tools and fixtures to programming factory floor equipment - can be performed long before production start-up and at far less cost than ever before. VM software fills the technology gap between design and manufacturing as shown in fig. 2.3. It gives engineers the technology they need to perform their complex tasks.

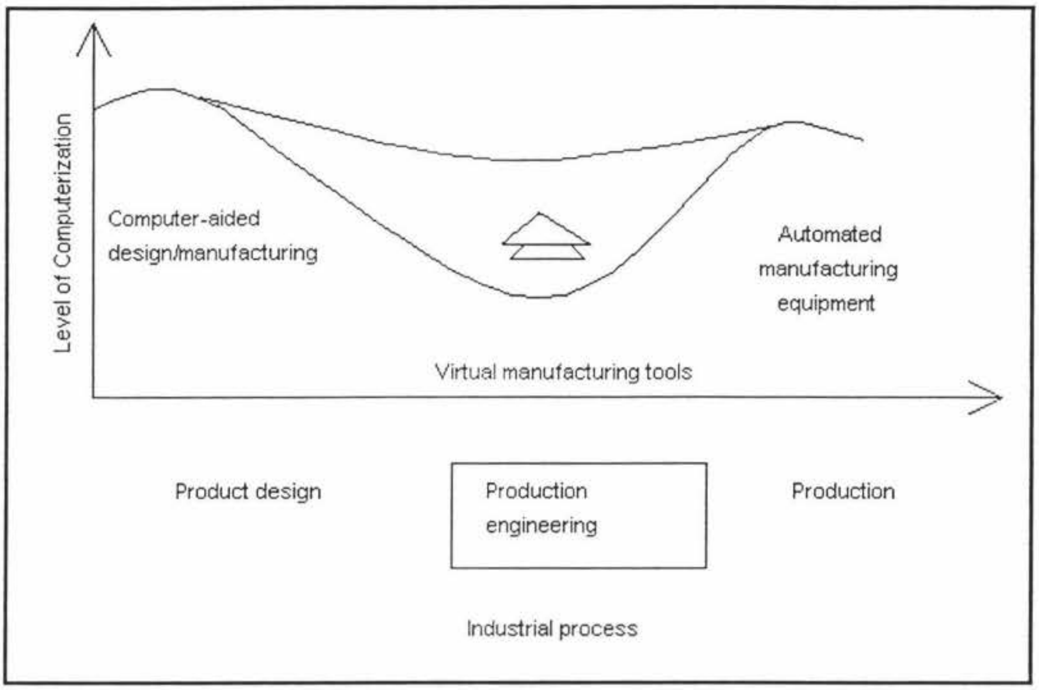

Level of Computerization
Computer-aided design/manufacturing
Automated manufacturing equipment
Virtual manufacturing tools
Product design
Production engineering
Production
Industrial process

Fig. 2.3 VM software fills the gap in automating the industrial process (Gilad Lederer, 1995)

2.3 Moving Virtual Manufacturing into the mainstream

Some manufacturers have long established VM tools in the industrial process. Others are rapidly closing the gap, migrating to VM tools as they did to CAD in 1980s. VM has evolved from simulation and offline programming tools to mainstream computer-aided-production engineering (CAPE) tools which are tightly integrated with the other computerised and automated tools in the industrial development cycle. ROBCAD, developed by Technomatix Technologies Ltd., Israel, is one such VM software tool which provides the capability to start manufacturing process design at the same time as production design (Gilad Liderer, 1995)

2.4 VM Vision

The vision of Virtual Manufacturing is to provide a capability to "Manufacture in the Computer". In essence, VM will ultimately provide a modelling and simulation environment so powerful that the fabrication/assembly of any product, including the associated manufacturing processes, can be simulated in the computer, as shown in fig. 2.4. This powerful capability would take into account all of the variables in the production environment from shop floor processes to enterprise transactions. In other words, VM will accommodate the visualisation of interacting production processes, process planning, scheduling, assembly planning, logistics from the line to the enterprise, and related impacting processes such as accounting, purchasing and management.

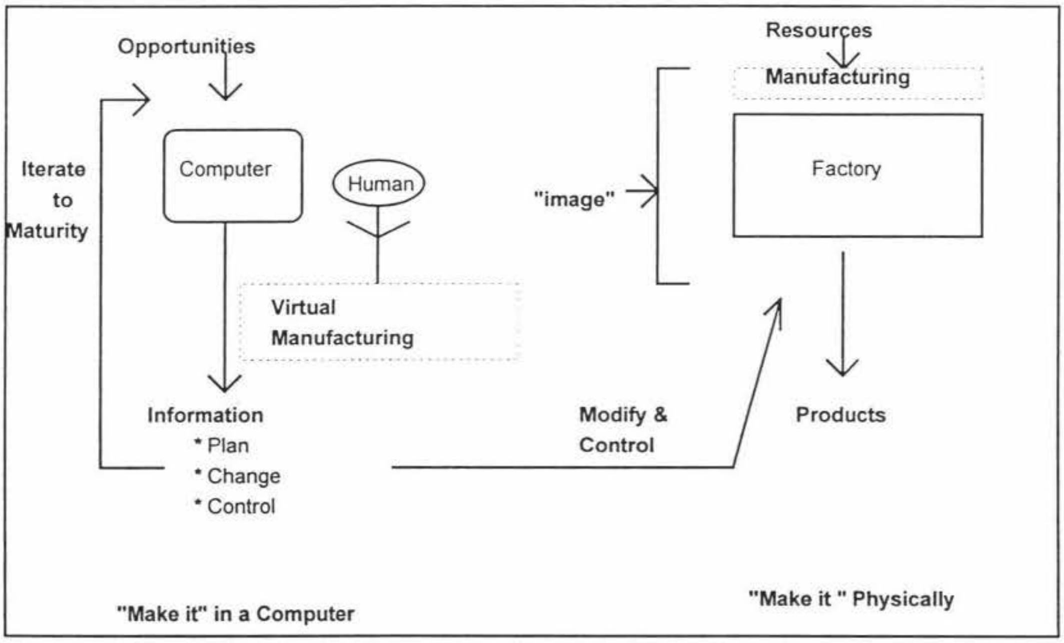


Fig. 2.4. VM Vision

Three paradigms emerged during the workshop conducted in Dayton,Ohio, 1994, on Virtual Manufacturing. For each of these paradigms, a definition of VM was proposed to capture the view of VM within that paradigm. For each of these definitions, the term "Manufacturing" should be construed in a broad sense to include not only production, but also suppliers, customers, and other processes that impact production.

2.4.1 Design-Centered VM: VM adds manufacturing information to the product development process with the intent of allowing simulation of many Manufacturing alternatives and the creation of many "soft" prototypes by "Manufacturing in the Computer"

A near-term definition: VM is the use of manufacturing-based simulations to optimize the design of product and processes for a specific manufacturing goal such as: design for assembly; quality; lean operations; and/or flexibility.

A longer-term definition: VM is the use of simulations of processes to evaluate many production scenarios at many levels of fidelity and scope to inform design and production decisions.

2.4.2 Production-Centered VM: VM adds simulation capability to manufacturing process models with the purpose of allowing inexpensive, fast evaluation of many processing alternatives.

A near-term definition: VM is the production-based converse of IPPD (Integrated product process development) which optimizes manufacturing processes, potentially down to the physics level. An example would be evolutionary re-engineering/optimisation of a fabrication facility.

A longer-term definition: VM adds analytical production simulation to other integration and analysis technologies to allow high confidence validation of new processes and paradigms. Examples would include revolutionary re-engineering of processes or factory, and/or introduction of virtual corporation paradigms.

2.4.3 Control-Centered VM: VM is the addition of simulation to control models and actual processes, allowing for seamless simulation for optimization during the actual production cycle.

In general, the workshop participants did not consider a "control-centered" use of VM a high priority.

In summary, Design-centered VM provides Manufacturing information to the designer at the design phase. Production-centered VM uses simulation during production planning to optimize lines/factories, including the evaluation of processing alternatives. Control-centered VM uses machine control models in simulations, the goal of which is process optimization during actual production.

2.5 Benefits of VM

VM provides the following four fundamental changes to manufacturing:

1. VM can be used to prove the production scenarios, resulting in "pre-production hardened systems" (i.e. systems which are developed and verified but never actually undergo actual production runs)

2. VM can support the generation of more reliable production costs and schedule because the models are based on actual processes, not just parametrics

3. Modelling & simulation (M&S) can significantly improve production flexibility, hence, reducing the "fixed costs"

4. Reliable predictions of costs, risk and schedule can substantially improve the decision making process of acquisition managers.

As a result of these changes to manufacturing, VM will contribute to realising the following benefits:

- Affordability - Reliable cost and process capability information that can impact key design and management decisions, and support balancing system performance with manufacturing cost, schedule and risk.

- Quality - More producible designs moving to the shop floor and higher quality work instructions to support production.

- Producibility - The ability to make the initial production that is trouble-free, high quality, involves no reworks, and meets requirements. Optimize the design of the manufacturing system in coordination with the product design.

- Flexibility - The ability to execute product changeovers rapidly, to mix production of different products, and to return to producing previously shelved products.

- Shorter Cycle Times - Increased effectiveness of IPPD (Integrated product process development) process and the ability to go directly into production without false starts.

- Responsiveness - The ability to respond to customer "what-ifs" about the impact of various funding profiles and delivery schedules with improved accuracy and timeliness.

- Customer Relations - Improved relations through the increased participation of the customer in the IPPD process, lower costs, better schedule performance, improved quality and greater responsiveness.

2.6 How VM will be used

The following categorisation shows the breadth of areas in which VM might be used.

Shop Floor - In the near term, shop floor people and concerns should have a greater influence on the design process, and manufacturing approaches that have been modelled and simulated above the shop floor will be brought out on the shop floor to validate the models and simulations. In the longer term, significant improvements to work instructions will be seen through the ready availability of graphics. Much better tooling will be available on the shop floor with features that make it easier for the worker to succeed via access to better instructions and illustrations to use error-free tool use. This will also make it easier to accommodate the envisioned drop in the average skill and education level of shop floor workers. The proofing of manufacturing designs & processes in the computer prior to commitment to hardware should sharply reduce the problems on the shop floor.

Capital Investment - Manufacturing models and simulation will and are having some influence on capital decisions currently, but this use is isolated to a few companies and not widespread within those companies. In the longer term, VM should be widely used in capital investments decisions since it should allow more credible comparisons of investments alternatives and should also provide history on the performance of past investments which is frequently hard to obtain in the current environment.

Supplier Management - The current VM impact on suppliers is probably small and the use of VM by suppliers themselves would probably be limited to the bigger companies because of the anticipated large investment required to install VM. The future impact on supplier management, however, is expected to be very significant. Make/buy decisions will be enhanced through easy access to better quality and more detailed information on costs, capacity, process capability and lead-times as part of the make/buy decision process. Cost control would also be enhanced because of the more accurate cost information available about suppliers. Major suppliers will have early involvement in product design and planning through the IPPD teaming approach that is likely to be an accelerating and long-lasting trend and will interact with VM in that context. Smaller suppliers will probably be positively impacted by getting much better and more stable product requirements information from their customers and the customers should be positively impacted by not having to invest so many resources in having to solve problems with their suppliers.

Product Design - In the near term, available and emerging modelling and simulation will enhance the effectiveness of systems integration in the design process, and as a result, improve the fit of the components, minimise interference between subsystems and, reduce the dependence on hard-mockups. In the longer-term, major improvements to the transition from the design to production are envisioned because of much stronger and more effective influence of process capacity and manufacturing cost information on the product designer as well as the ability to do many more design iterations prior to committing to hardware. The outcome of the information from VM and the design process, i.e., the computer based models and the simulations can be used in training the manufacturing workforce.

Cost Estimating - The move towards VM will necessitate finer-grained, more accurate cost information that can typically provide current cost accounting systems (and VM cannot succeed without this kind of information). This will, in turn, accelerate the current trend towards activity based accounting systems and other accounting system changes that allow detailed and accurate product costing. Currently, the industry is relying on "semi-expert" systems for product cost estimation, but these systems are not fully reliable. These systems are not very satisfactory and will be abandoned as the industry moves into VM and better data becomes available to support more accurate approaches. Future VM systems will provide accurate cost data throughout the design, development and production process. Cost estimating systems will become fully integrated with design and manufacturing databases and will access detailed process-level design feature related data.

Customer Interface - The interest and enthusiasm of the customer for VM could potentially lead to a temptation for companies to exaggerate the use and impact of VM in their dealings with the customer. In spite of this risk, near-term impacts are likely in more effective inclusion of the customer in the IPPD process; the inclusion of some requirements for VM in customer statements of work; and better responses to customer "what-if" questions about changes to budgets and delivery schedules. In the longer-term, VM will enhance the credibility of responses to "what-if" queries significantly and this, in turn, will have an important impact on program stability by allowing decisions about program budgets and delivery schedules. The customer's ability to participate in the IPPD process should be greatly improved. Uncertainty remains about what changes might evolve in customer oversight as a result of the enhanced visibility available.

Functional Interfaces - VM will potentially accelerate the current trend towards weaker functional distinctions within companies by promoting the widespread sharing of information and enhancing close inter-functional working relationships within the IPPD process. This trend in the longer term, should lead to weakening of the influence of functional departments within the companies and their customers as information sharing becomes even more widespread and effective, and as work efforts are more likely to be organised on product basis rather than being functionally oriented.

2.7 Virtual Reality in Manufacturing

Carolina Cruz-Neira (1993) defines virtual reality as, "immersive, interactive, multi-sensory, viewer-centered, three-dimensional computer generated environments and the combination of technologies required to build these environments."

Virtual Reality (VR) can be described as the science of integrating man with information. It consists of three-dimensional, immersive, interactive, computer generated environments. These environments can be models of real or imaginery worlds. VR is a set of computer technologies which, when combined, provide an interface to the computer with which the user can believe to be actually in a computer-generated world. Immersion and interaction are the two key capabilities of VR that distinguishes it from the traditional interface of the computer. It provides users with the feeling of occupying space in a computer graphic environment where they can interact with the virtual world using hands and legs to manipulate objects for some tasks and operate with natural gesture to look around the virtual environment.

VR is a powerful tool for human and computer interface. It can be characterised by two main features - real-time processing in order to react immediately to the action of the users, and close integration of humans by means of novel input/output and interaction devices like trackers, gloves and head-mounted displays.

VR is also a useful tool in the design process which consists of knowledge base, design, analysis, realization and communication. In addition, VR can increase the overall productivity of an individual. David Kahaner (1993) elaborates the use of VR by Japanese researchers in a wide range of applications such as, equipment operation, image processing, remote control of construction robots, and also designing molecular models. Since, VR's main objective is to generate virtual worlds as realistic as possible to the human perception, a lot of research is focussed on human factors and immersive technology (Mark Bolas, 1994).

A wide range of VR applications in manufacturing have been highlighted by Sandy Ressler (1994), such as the improvement of the design process for heavy equipment, automotive assembly, interaction with a virtual lathe. Gunter Wittenberg (1995) explains how Motorola is using the VR software for their flexible robotic assembly lines. VR software allows three-dimensional, computer-generated information to be created in real time. This offers more freedom and versatility than other means because it provides complete representation of information.

VR finds its application in defense as well, as a flight simulator for training pilots in a virtual world (Dean McCarty et. al., 1994). VR is also used by researchers for medical applications like, modelling the heart beat of humans, simulating the human anatomy and a Virtual Clinic that aids surgeons in improving their techniques (John Adam, 1995).

Based on these various applications of VR technology specially in the field of manufacturing has lead to the idea of using Virtual Reality Software called QUEST® in this research for modelling and simulation of a flexible manufacturing facility. Later these models are used for analysing the scheduling rules in a Flexible Manufacturing System.

The following Chapter 3 explains in detail the Scheduling activities of Flexible Manufacturing Systems and covers a review of relevant literature.

Chapter 3

SCHEDULING OF FMS : REVIEW OF RELEVANT LITERATURE

This Chapter gives a brief introduction to the structure of manufacturing systems, Flexible Manufacturing Systems (FMS), Scheduling activities in FMS, the application of simulation as an analysis tool for the evaluation of various scheduling rules in an FMS and the review of relevant literature.

3.1 Structure of Manufacturing Systems

The structure of manufacturing system based on the plant layout pattern is basically decided by the relationship between the number of products P and the production quantity Q, and classified in the following three ways (Katsundo Hitomi, 1994):

Product (or Flow-line or Production line) Layout: In the case of a large Q : P ratio, continuous mass production is justified. Production facilities and auxiliary services are located according to the process route for producing the product, generating the linear material flow.

Process (or Functional) Layout: In the case of a small Q : P ratio, or jobbing or small lot (batch) production, machines of like types are located together as work centers in one area of the plant. In this case the flow of materials is not smooth, resulting in low productivity.

Group Technology (GT) (or Cellular) Layout: In the case of an average Q : P ratio, when a great variety of products can be grouped into several families, these families are manufactured as lots of similar parts, and machines are arranged to meet this type of production, thereby resulting in higher productivity. This is like a hybrid layout taking the advantages of both Product and Process Layouts.

Group technology is a technique that increases a production lot size by grouping various parts and products with similar shape, dimensions, and/or process route. Production with this concept (cellular manufacturing) increases productivity. The combined advantages of both the group technology and the flow-line production system has led to the development of flexible manufacturing systems, which is explained in the following section.

3.2 Flexible Manufacturing Systems (FMS)

A flexible manufacturing system, through the careful combination of computer control, communications, manufacturing process and related equipment enables a section of the production-oriented aspects of an organisation to respond rapidly and economically, in an integrated manner, to significant changes in its operating environment. Such systems typically comprise: process equipment (like machine-tools, assembly stations, robots etc.) material handling equipment (like robots, conveyors, automated guided vehicles etc.,), a communication system and a sophisticated computer control system.

3.2.1 Flexibility Criteria in FMS

Flexibility is the main characteristic of FMS. Automated manufacturing systems can achieve their greatest potential when they are designed to be flexible (P J O'Grady, 1986).

This flexibility can take a number of forms, including:

- Volume flexibility: The ability to handle changes in the production volume of a part.
- Re-routeing flexibility: The ability to have a number of routes through the system for each part in order to enable, for example, machine breakdown to be dealt with.

- Part flexibility: The ability to handle a wide variety of parts including the ability to adapt a system to handle a new part.

3.2.2 Classification of FMS

The following are the schemes based on which FMS are classified (Raouf & Ben Daya, 1995) :

Sequential FMS - A system which processes one batch at a time and is quickly reset for processing new batches.

Random FMS - A system in which the various types of products it is designed for can be processed in random order. The system can simultaneously process two or more different products.

Dedicated FMS - A system which is designed to produce continuously a limited number of products. However, this is not as rigid as a transfer line.

Modular FMS - A flexible manufacturing cell which has been expanded on a step-by-step basis into an FMS.

3.3 Hierarchical Planning Structure of FMS

The FMS planning stage can be classified into three levels as, Strategic, Tactical and Operational levels (Van Looveren et al., 1986). The hierarchical planning framework for these levels is shown in fig. 3.1.

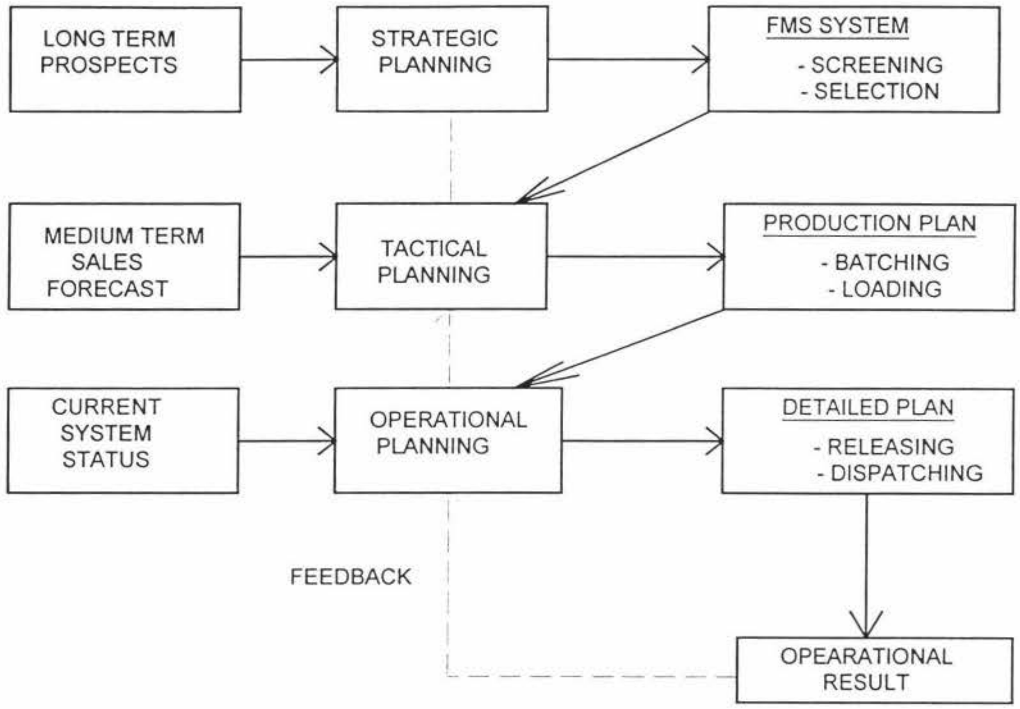


Fig. 3.1 HIERARCHICAL PLANNING STRUCTURE OF FMS (Van Looveren et. al., 1986)

3.3.1 Strategic Level : The strategic level is basically the responsibility of top level management and deals with long term decision making. This long term decision making includes the design of the manufacturing system taking into consideration the technical and economical feasibility. This can be achieved by following the **screening** and **selection** processes. These processes will assist in finalising the system layout and the selection of machine tools and the material handling system based on the forecast of the products.

3.3.2 Tactical Level : Once the system is installed, the tactical stage follows wherein, the **batching** process and the **loading** process are taken into consideration for planning the manufacturing system. The batching problem is concerned with parts and part related issues such as due dates, pallets and fixtures. The objective of batching process is to organise production such that orders are finished in time. This is achieved by splitting the production requirement into batches based on part types and part mix. Once the batching process is over, the resource allocation, i.e., machines and tools, has to be done in order to manufacture each of the part types. This is done by the loading process.

3.3.3 Operational Level : This level is concerned with the detailed decision making required for real time operation of the FMS. This is done by the **release** process and the **dispatching** process. The release process controls the flow of workpieces into the system by taking into consideration the resource allocation to each part type done at the tactical level and also the system disturbances such as, machine breakdown and transportation system breakdown. It also takes into consideration, the availability of pallets and fixtures, any changes from desired production rates. The dispatching process encompasses many decisions such as the selection of machines, buffers, transportation system, if there are several alternatives available.

This research is concerned with the dispatching process at the operational level of the FMS.

3.4 Scheduling of Manufacturing Systems

Scheduling is concerned with determining when and using what, resources, jobs or orders in a factory when they are competing for the resources like machine tools, jigs & fixtures, tools, materials, operators etc. The importance of scheduling is a consequence of the current trend in the manufacturing sector to increase product variety which results in decreasing batch size. Hence, the current manufacturing trend attempts to meet the requirements of the market with more flexibility of the production, the main idea behind the JIT philosophy.

Scheduling rules in a manufacturing facility can be classified as (Choi & Malstrom, 1988) :

a) Part selection rules : These rules are formulated to select a job from a set of jobs waiting to be processed. The selection of the job is done based on due dates and processing times.

b) Machine selection rules : These rules allow for the selection of a machine if alternative machines are available for processing the same operation.

The rules can be grouped into three categories (S.S.Panwalker et. al,1977):

Processing time based rules :

Shortest Imminent operation (SI); Shortest Processing Time (SPT); Least Work Remaining (LWR); Fewest Operations Remaining (FOR);

Due Date based rules:

Earliest Due Date (EDD); Smallest Slack per Remaining Operation (SL/OP);

General rules:

Random (R); First-In-First-Out (FIFO);

3.4.1 Simulation as Analysis Tool

Simulation is a widely used method by various researchers to analyse the performance of various dispatching rules in an FMS. It is used as a what-if analysis tool. According to Becker and Parr (1994), simulation provides the following advantages in the context of scheduling :

- Most scheduling systems need assumptions about the future in order to reach decisions. The alternative scenarios implied by different decisions are elaborated by simulation.

- Simulation enables the evaluation of various dispatching rules with respect to the decision objectives, before implementing them at the plant.

- A simulation of the manufacturing system is a testbed for the scheduler.

Montazeri and Wassenhove (1990) have used user-oriented discrete-event simulator for analysing scheduling rules of an FMS. Szu-Yung and Richard (1989) have used discrete-event simulation for on-line control and scheduling of an FMS in a dynamic fashion. Goyal et. al., (1995) have used simulation as the analysis tool for evaluating scheduling rules in an FMS. Yash Gupta et. al., (1989) have given an overview of all the scheduling rules in an FMS and highlighted the fact that part scheduling problem has the fundamental implications on the overall performance of the system. Choi and Malstorm (1988) have used a physical simulator to evaluate the performance of scheduling rules in an FMS.

Robert O'Keefe and Kasirajan (1992) have suggested that Machine Selection Rules have received less attention than Part Selection Rules (dispatching rules). Hence, the objective of this research has more emphasis on the evaluation of the performance of Machine Selection Rules than Part Selection Rules on the overall system performance.

The following Chapter 4 emphasises the methodology being adopted in this research to model an experimental FMS using a Virtual Reality Discrete-Event Simulation tool, followed by the evaluation of the above mentioned rules.

Chapter 4

METHODOLOGY

This chapter discusses the design of Production System for this research and the selection of the simulation software package for modelling the Production System. It also gives an overview of the QUEST® simulation package and the Software Development for achieving the research objectives.

4.1 Design of Production System

Conventionally manufacturing can be divided into three major categories:
(P.J. O'Grady, 1986)

1. Flow or mass production
2. Batch manufacture
3. Jobbing manufacture

Flow or mass production is concerned with producing a limited range of products in high volume (for example, car assembly).
Batch manufacture deals with a much larger product range than flow manufacture, but the products tend to have lower volumes and repeat orders are expected.
Jobbing manufacture produces 'one-offs', that is, there is no expectation that there will be repeat orders for the products. Jobbing manufacture is characterised by a high product-type range but a low volume.

The Production System modelled in this research is of Batch Manufacturing type, as the current trend of manufacturing is 'flexible manufacturing' which is production of a large variety of products in small batches or volumes (Nigel R. Greenwood, 1988).

4.2 Concept of "push type" and "pull type" Production Ordering Systems

(Muramatsu et al., 1995)

In the "push type" production ordering system, the ordered quantity in each stage is determined by forecasted demand. Forecasted demand is the length of cumulated lead time from one stage to the final assembly line, and of feedback information of product or in-process inventory in each stage. In this system, the ordered quantity of each production stage is ordered by a central controller. Thus it may also be called as "centralised ordering system". Material flows are controlled just as if they are "pushed out" from the raw materials stage toward the final stage.

In "pull type" production ordering systems, the ordered quantities in each stage are determined by actual quantities consumed by the immediate next stage. Here no central controller is needed. Thus it may also be called a "decentralised ordering system". Material flows are controlled just as if they were "pulled" into the final product stage from the stages preceding final production.

Due to the limitations of the Simulation Software used in this research, "push type" production ordering system is taken into consideration in designing the Production System.

4.3 Selection of Simulation Software Package

For any simulation study, a simulation package must be chosen carefully so that it can accurately represent the system being simulated (Felix T.S. Chan, 1995). Also, the simulation package should be suitable to those who are going to be working with the model.

The following criteria should be considered for the selection of the simulation package (Grant et. al., 1986):

- Modeller and end user (training, knowledge, skill)
- Ease of construction of models.
- Ease of creating graphics.
- Graphics and animation capabilities.
- Interaction with the models.
- Hardware requirements (computer, display type, speed)
- Cost (initial and software upgrades).
- Simulation speed.
- Simulation reports and statistics

As QUEST® simulation package provides all the above mentioned capabilities, it has been chosen for conducting the simulation study in this research.

4.4 QUEST® Simulation Package Overview

The QUeueing Event Simulation Tool, QUEST®, is an interactive, 3-D simulation tool aimed at all batch process manufacturing, whatever the product, and is capable of producing textured virtual reality walk throughs. This tool is capable of providing high quality, 3-D graphics of geometrically and dimentionally correct models. This means that the user can come out with the solutions for designing optimal plant layout by simulating various 'what-if' production scenarios.

In this research this tool is used to model a hypothetical FMS and analyse the scheduling rules in an FMS.

4.4.1 Modelling

QUEST® includes a rich resource library of geometric models which, combined with a Visual File Interface, enables users to quickly and accurately model:

- Workcells and Processes
- Labor

- Conveyors, Power & Free
- AGVs, Forklifts, Cranes
- Buffers, AS/RS (Storage System)

QUEST® has an integrated CAD package using which the user can build other devices and also can import from IGES, DXF and other formats.

4.4.2 Simulation Control Language

QUEST® models directly emulate real-world system behaviours through distributed logics that can be associated with each resource, including:

- Route, Sequence
- Composite Processes with Requirements/Selection Rules
- Buffering Policies
- Push Production Attributes
- Request-based Decision-making

The most commonly needed behaviour logics are easily selected from comprehensive logic menus.

For unique problems, QUEST®'s **Simulation Control Language (SCL)** can be used. This high-level, structured language provides distributed processing with access to all the system variables. SCL allows users to define custom behaviours and gain unlimited control over the simulation.

4.4.3 Batch Control Language

QUEST®'s open architecture allows the advanced user to perform batch simulation runs to automatically collect and tabulate data using the Batch Control Language (BCL). Replications and parameter optimisation are easily controlled with simple batch command files; advanced applications can be developed to drive QUEST® from any program on a UNIX network using TCP/IP socket protocols.

4.5 QUEST® Menu System

QUEST® has a Three-tier Menu System which is placed on the QUEST® window in the following manner:

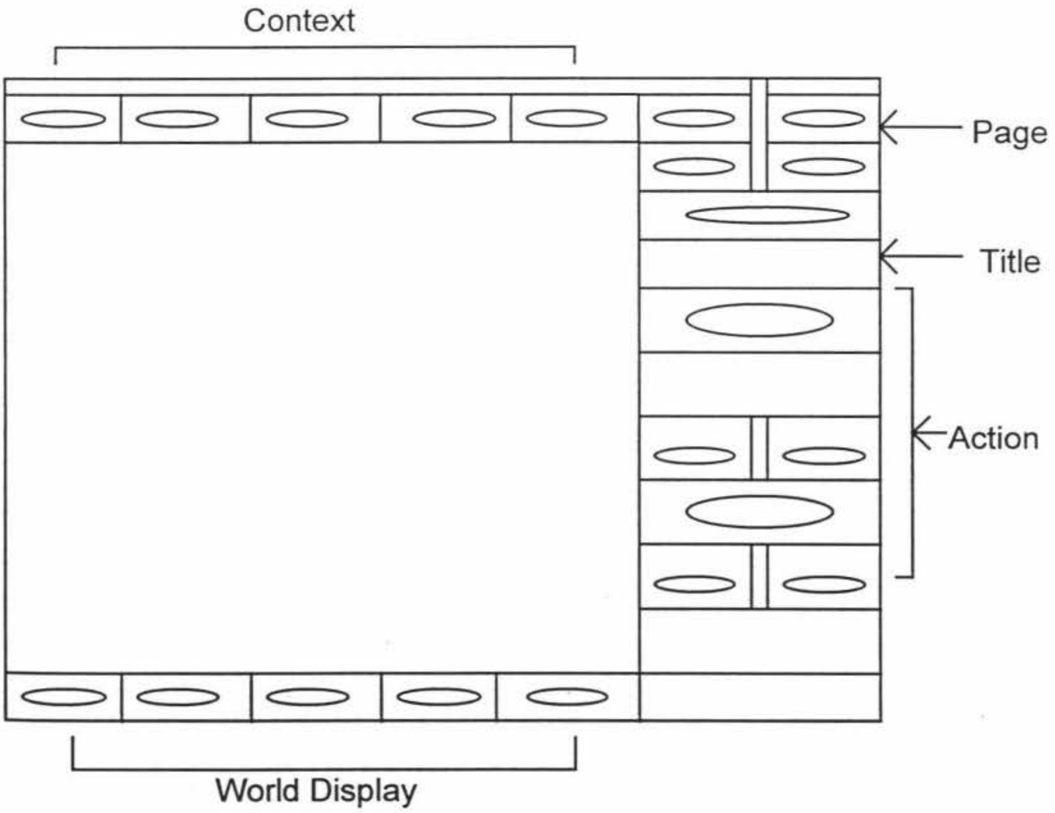


Fig. 4.1 Display of QUEST® Menu screen

4.5.1 Context

Contexts are the main division of functions in the QUEST® Menu System. The Contexts are arranged at the top of the QUEST® screen. Each Context has a group of PAGES associated with it. The Contexts in QUEST® are:

CAD	Kin	Model	MHS	Run	Stats	Draw	User	Analysis	Sys

CAD Context helps the user in building Workcells or other Devices to be modelled. *Kin Context* helps the user to create Kinematic Devices or Workcells, thus giving kinematics to Workcells with various degrees-of- freedom.

Model Context provides with the facilities to the user to create a model of a factory or any manufacturing facility with all the resources such as Workcells, Buffers and also Labor. These resources can be called from the built-in library or can be created by the user and stored in user's library and later retrieved when required.

MHS Context helps the user to model, a Material Handling System such as Conveyors, Power & Free system , AGV's and Cranes etc., for any manufacturing facility.

Run Context is used to execute the simulation of any model for the user defined time.

Stats Context provides the user with all the statistical data of all the resources in the model such as the Busy Time, Idle Time, Blocked Time etc.,

Draw Context allows the user to create 2D drawings consisting of lines and text, and plot these to any scale.

User Context allows the user to create user-defined action Buttons and assign user-defined SCL Macros to them. When these Buttons are clicked they perform the user-defined tasks.

Analysis Context allows the user to do some dimensional analysis on the layout of any modelled manufacturing facility i.e., it helps in finding out the correct dimensions of all the resources modelled and also the distance between each of the resources in the model.

Sys Context allows the user to set the System parameters before creating a model. The Parameters such as setting the grid size of the floor over which the model is built.

4.5.2 Worlds

Contexts also have a distinction associated with them regarding "worlds". The CAD context is a separate world, called the CAD world. The KIN, MODEL, MHS, RUN, STATS contexts operate in the Model World. The DRAW context is a separate world called the Draw World. The remaining Contexts (USER, ANALYSIS, and SYS) are not world specific and may be entered and used in all the worlds. Certain functions in these Contexts may be disabled depending on which world is active at the time the Context is entered.

4.5.3 Page

Pages are the secondary division of functions. Each Page provides access to a group of Buttons. Each Context has between two and nine Pages.

4.5.4 Action Buttons

Action Buttons are the basic functions that perform tasks.

Action Buttons are of the following type:

Regular - They perform the action assigned to them i.e., when activated by clicking the mouse on these Buttons, they execute the Procedures, Routines or Macros associated with them.

Toggle - They turn a state flag alternatively True and False.

Confirm - These Buttons ask for a second mouse click of confirmation before executing the function assigned to them.

4.5.5 Logic Flow for various resources in QUEST®

Figures 4.2, 4.3, 4.4 show the flow of logic for the Model, Workcell, and the Resources. Resources in QUEST® include Buffers, Material Handling Systems like Conveyors and Power & Free, and Workcells. The flow of logic is same for all the Resources except Workcell which is shown in figure 4.3.

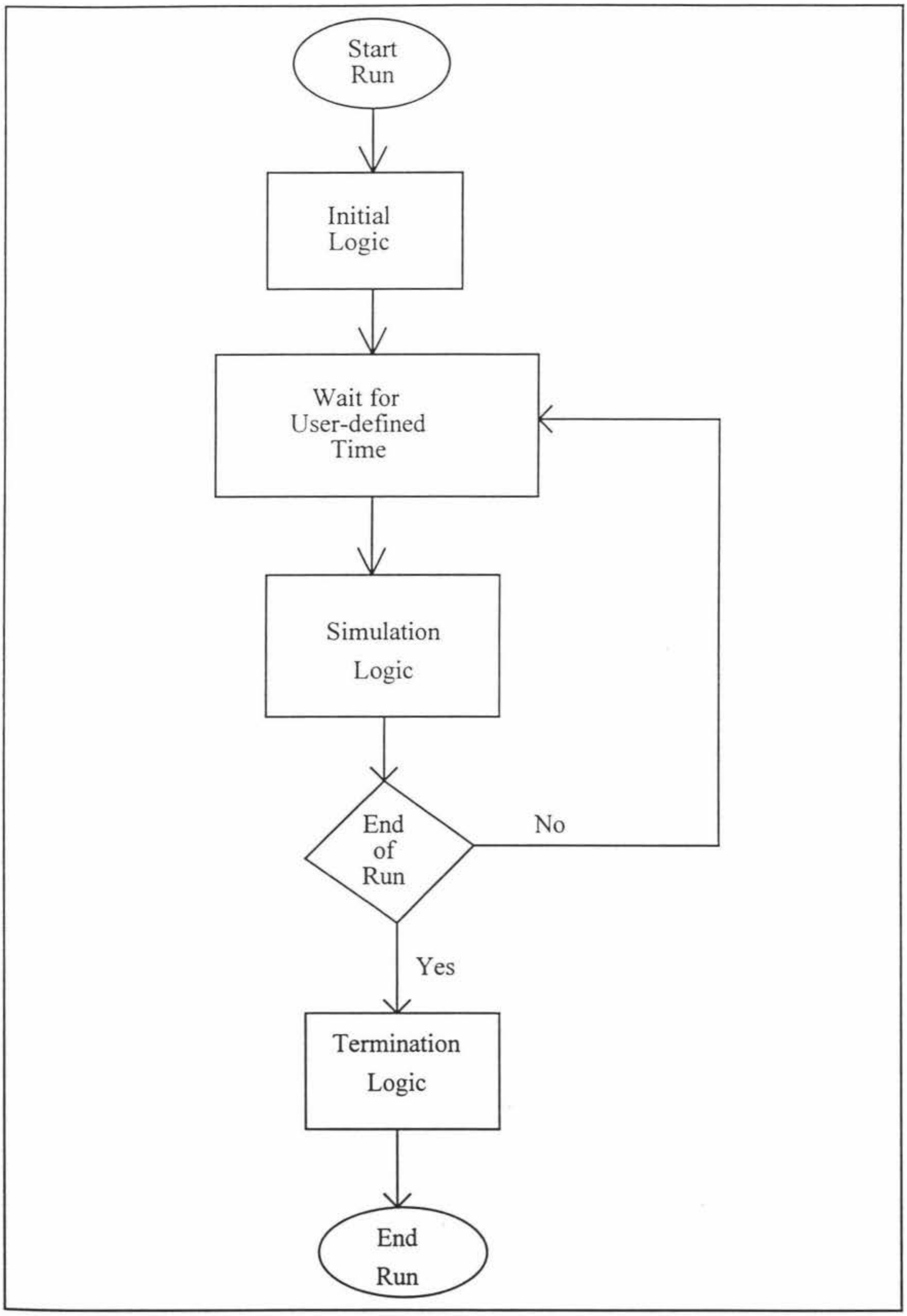


Fig. 4.2 Model Logic Flow

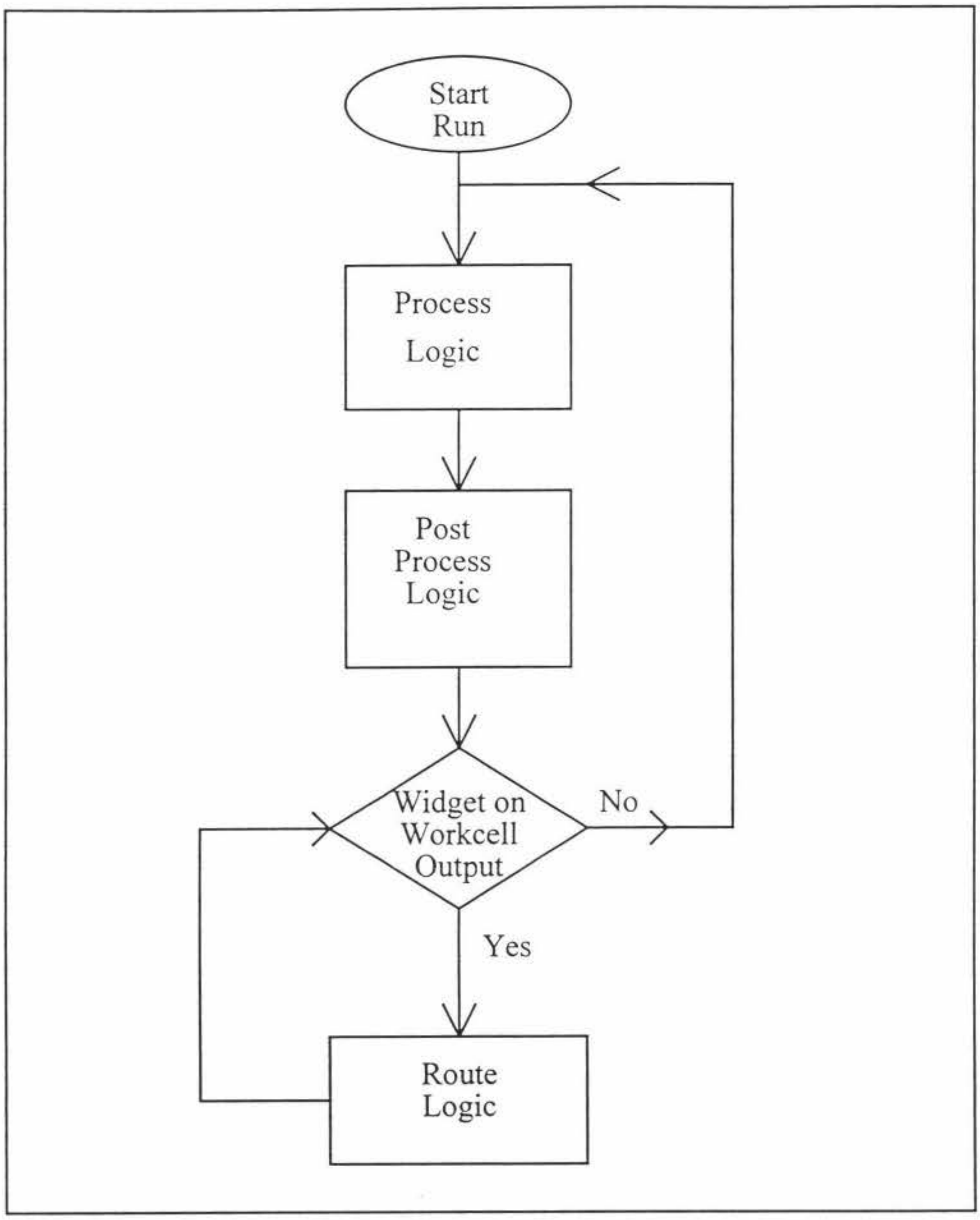


Fig. 4.3 Workcell Logic Flow

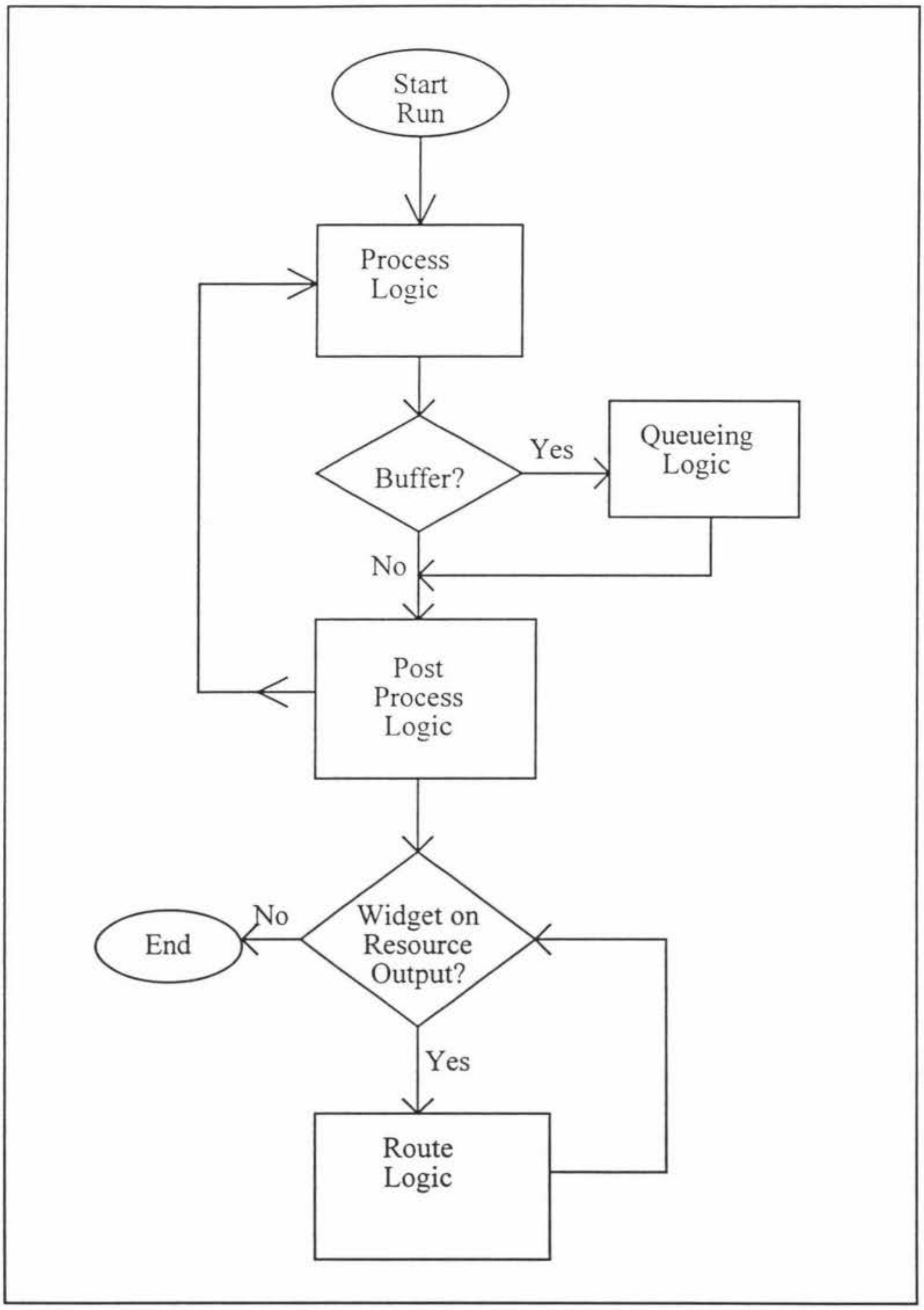


Fig 4.4 Resource Logic Flow

4.6 Definitions of Logics used in QUEST®

QUEST® has eight different types of SCL logics that are called at specific times during a simulation. The eight types are defined below and when they are called during a simulation is shown in figures 4.1, 4.2, 4.3. These logics help in decision-making and control of simulation for any model.

4.6.1 Initialisation Logic

This logic is called at the beginning of model execution. It is useful for system initialisation and the creation of shared global variables. It is empty by default.

4.6.2 Simulation Logic

Simulation Logic is called at a user-defined time interval throughout the course of the simulation run. It is useful for data gathering, data analysis, or to make systemwide changes to a model based on a set schedule. It is empty by default.

4.6.3 Termination Logic

This logic is called at the end of a simulation run. It is useful for taking data collected during the run and writing it to a file in a specified format. It is empty by default.

4.6.4 Process Logic

Every resource in a model has a Process Logic. Once the Initialisation Logic is complete for the model, every resource starts its Process logic. Process Logic is used to model everything from the assembly of two parts to the movement of a widget or job on a conveyor. Once a Process Logic has been completed, it will wait until the Route Logic for the resource has been completed then it will execute the Process Logic again.

4.6.5 Queueing Logic

Queueing Logic is executed whenever there is a widget or job sitting on the output of a buffer. This logic arranges the jobs in the order as defined by the user.

4.6.6 Post Process Logic

This Logic is meant as a location for the users to write SCL, make logical changes to a system, or perform other functions without being burdened with a set task to complete. Post Process Logic is empty by default.

4.6.7 Route Logic

Route Logic is executed whenever a widget exists on the output of a resource. The only ways to get a widget to the output of a resource is to use a PASS, PRODUCE, or DO_PROCESS statement. This logic will decide the next resource to which the widget should go in order to be processed further.

4.6.8 Decision Logic

Decision Logic is executed whenever a widget encounters a Decision Point on a MHS (Material Handling System). Decision logic will end with either a TRANSFER statement to remove the widget from the MHS or a resume_travel statement to continue moving on the MHS.

4.7 Model creation using QUEST®

QUEST® software uses object-oriented graphical user interface and it is very easy to model any manufacturing facility. With its ability to produce graphical models, it gives good visual effects and easy understandability of the models.

The construction of the model is made up of two types of entities, those that remain fixed in the system, such as, machines, and those that tend to move through the system, such as parts or products. The fixed entity is known as a "resource" and the moving entities are called "widgets". The **"widgets"** are defined as the entities that represent the parts or products or the information flowing through the system. Widgets receive services from Resources or are processed by them. The widgets enter the system and interact with the objects in pre-defined ways. The movement of widgets from resource to resource is defined by paths called "flows" and these paths are governed by Route Logics. A widget can be given a series of alternative paths along which to travel depending on model conditions and logics.

To create the model the following steps are involved:

1. **Resource creation:** The resources created in QUEST® are Sources, Sinks, Workcells, Buffers, Material Handling Systems like Conveyors and AGVs.

   Sources are the objects which generate widgets. The widgets generation may be defined by using a number of statistical distribution options such as, normal, exponential, triangular, Weibull, Piossons etc., Also it may be defined by a file-based information which is user-defined, i.e., the user defines times in a file based on which the Source generates widgets. The various attributes to be defined by the user to create a Source are shown in fig. 4.8

   Sinks are the objects which collect the information regarding the finished parts or products or the information flow after the completion of simulation. The various attributes given by the user to a sink are shown in fig. 4.9

   Workcells are the objects where the actual processing is done on the widgets. The processing times may be defined by using a number of statistical distribution options such as, exponential, normal, triangular, Weibull and file-based information. The various attributes to be defined by the user to create a Workcell are shown in fig. 4.5

Material Handling Systems (MHS) such as Conveyors, AGVs and Power & Free are objects which are used in the model for transporting the widgets from one resource to another. The various attributes to be defined by the user for the creation of MHS are shown in fig. 4.6

Buffers are the objects where the widgets, which are waiting to be processed, are queued in order to be processed by a Workcell. The various attributes to be defined for the creation of Buffers are shown in fig. 4.7

The geometries of all the resources mentioned above are selected from built-in libraries or are selected from user-defined libraries which contain user-defined objects. Once selected they are placed on the screen as per the user requirements.

2. **Widget Routeing:** It is the paths between the resources which are defined for a specific widget type. A widget may represent a part, product or information flow.

3. **Widget Interaction:** This is defined as the interactions between a given widget type and a given resource. These interactions are accomplished by the connections established between various resources in the model.

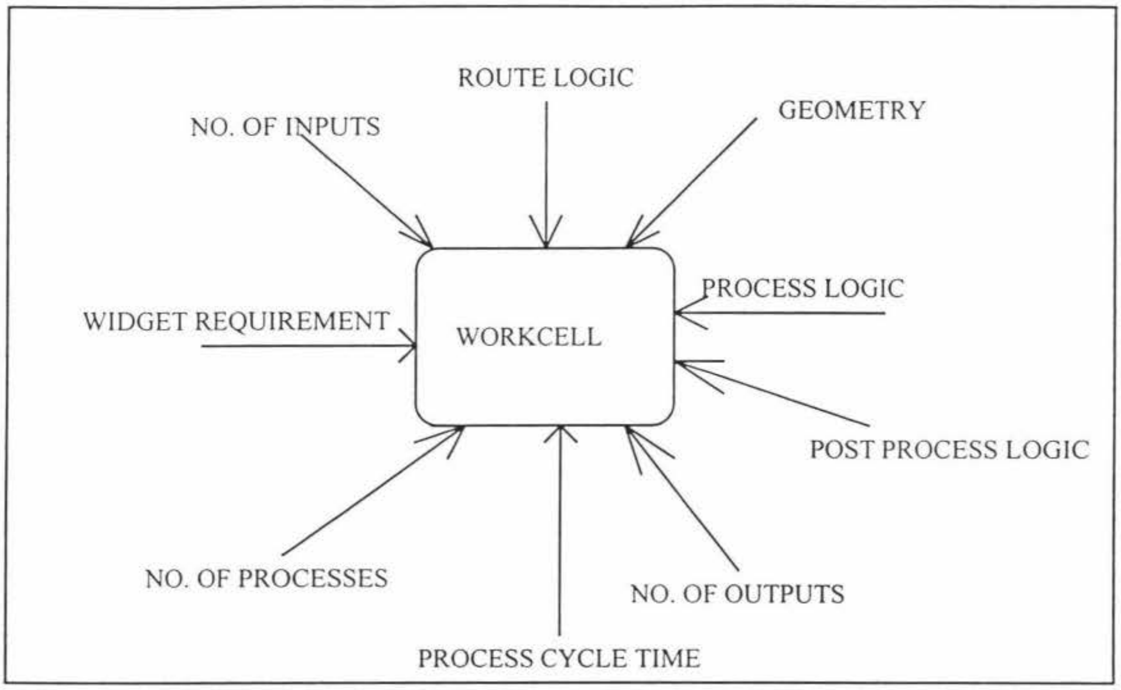


Fig. 4.5 Various attributes defined for a Workcell

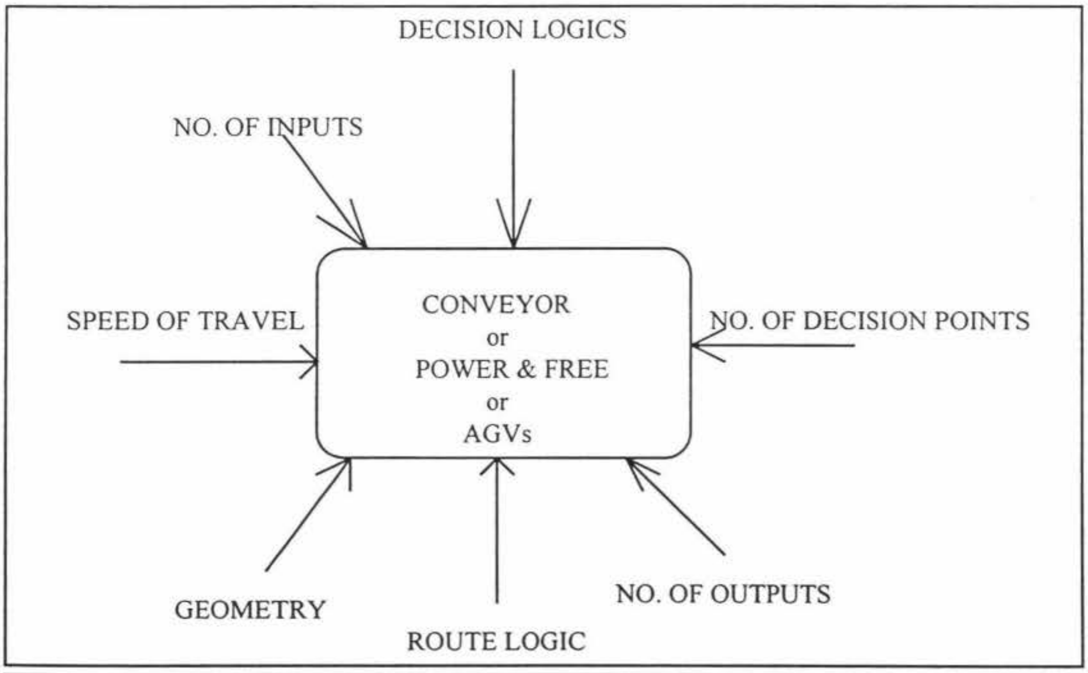


Fig. 4.6 Various attributes defined for a Material Handling System

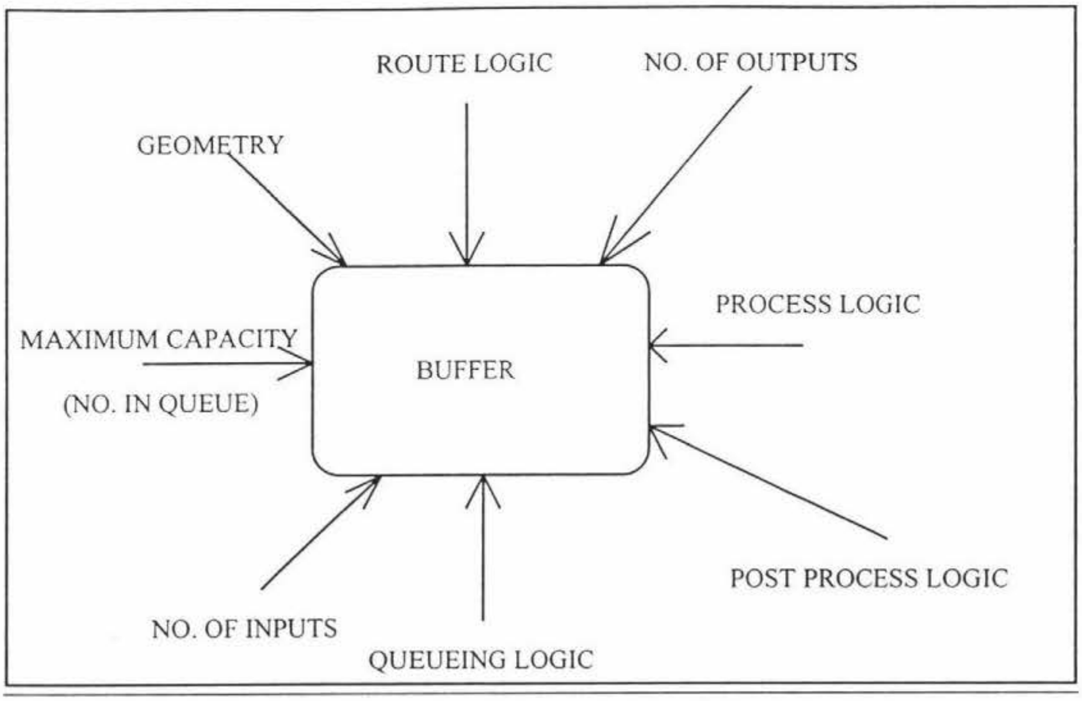


Fig. 4.7 Various attributes defined for a Buffer

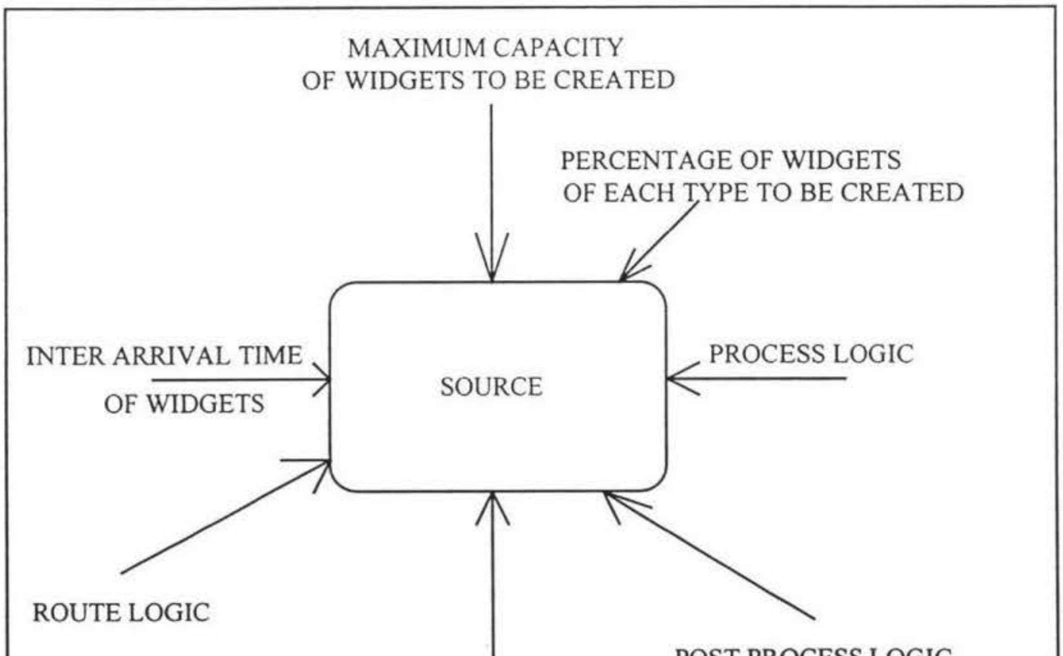


Fig. 4.8 Various attributes defined for a Source

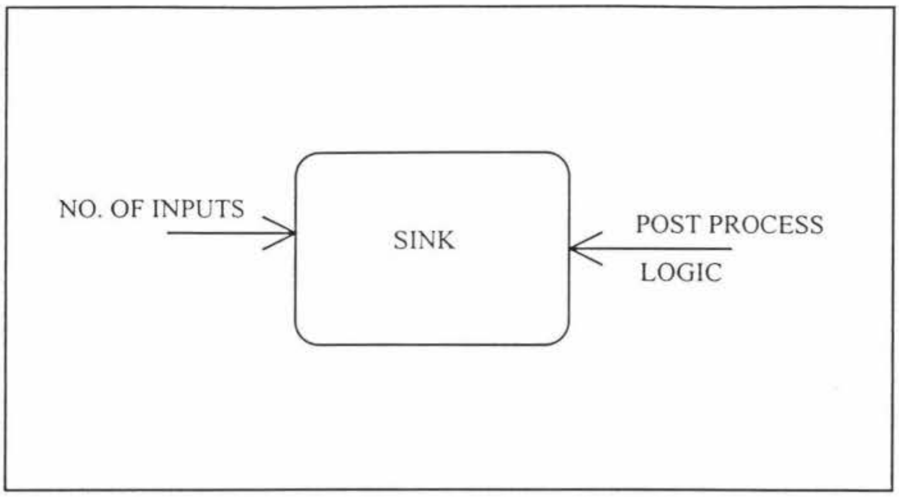


Fig. 4.9 Various attributes defined for a Sink

4.8 Software Development

Various procedures are written in SCL (Simulation Control Language) for the development of models to achieve the main objectives of this research. The program code is listed in Appendix A.

The procedures are written for the Process Logics of Workcells, Route Logics for all the resources in the models and the Queuing Logics for the Buffers. These logics are written in order to model and simulate the scheduling activities in a manufacturing facility, which is explained in detail in Chapter 5.

Fig. 4.10 represents the flow charts for the model control using the Part Selection and Machine Selection algorithms and fig. 4.11 shows the flow chart for the model control using the Machine Breakdown algorithm.

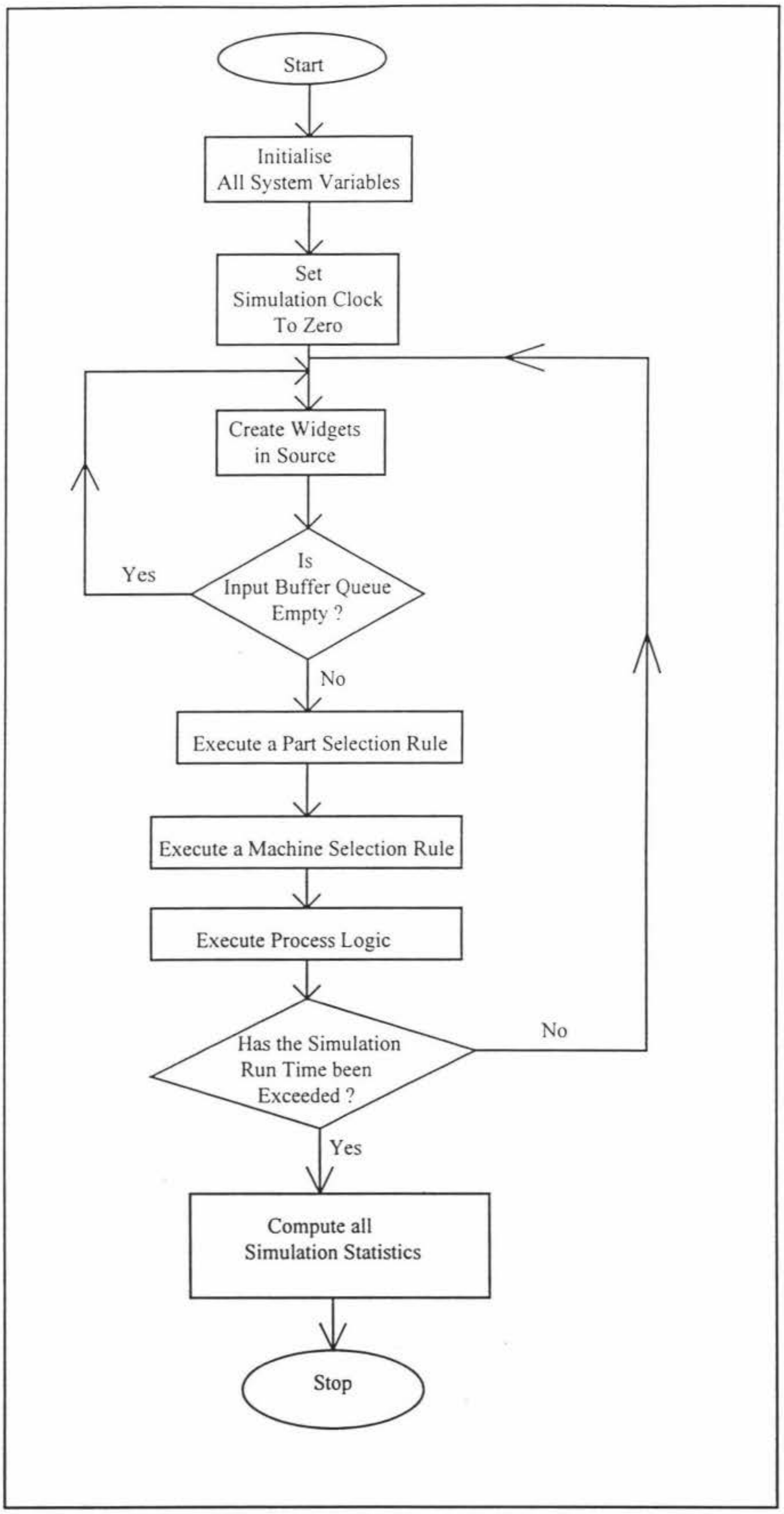


Fig. 4.10 Flow Chart for Model Control using Part Selection and Machine Selection Rules.

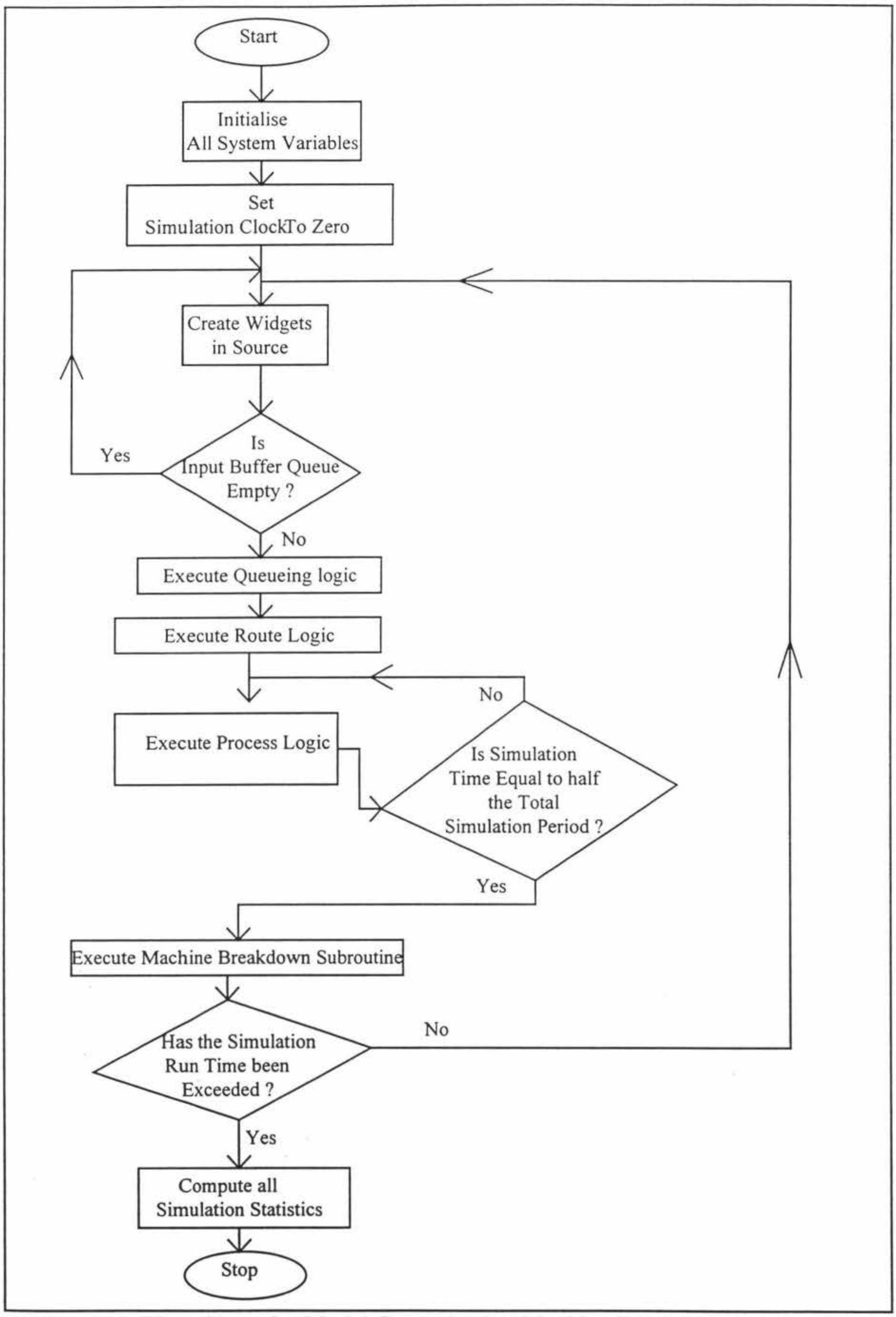


Fig. 4.11 Flow Chart for Model Control using Machine Breakdown Logic

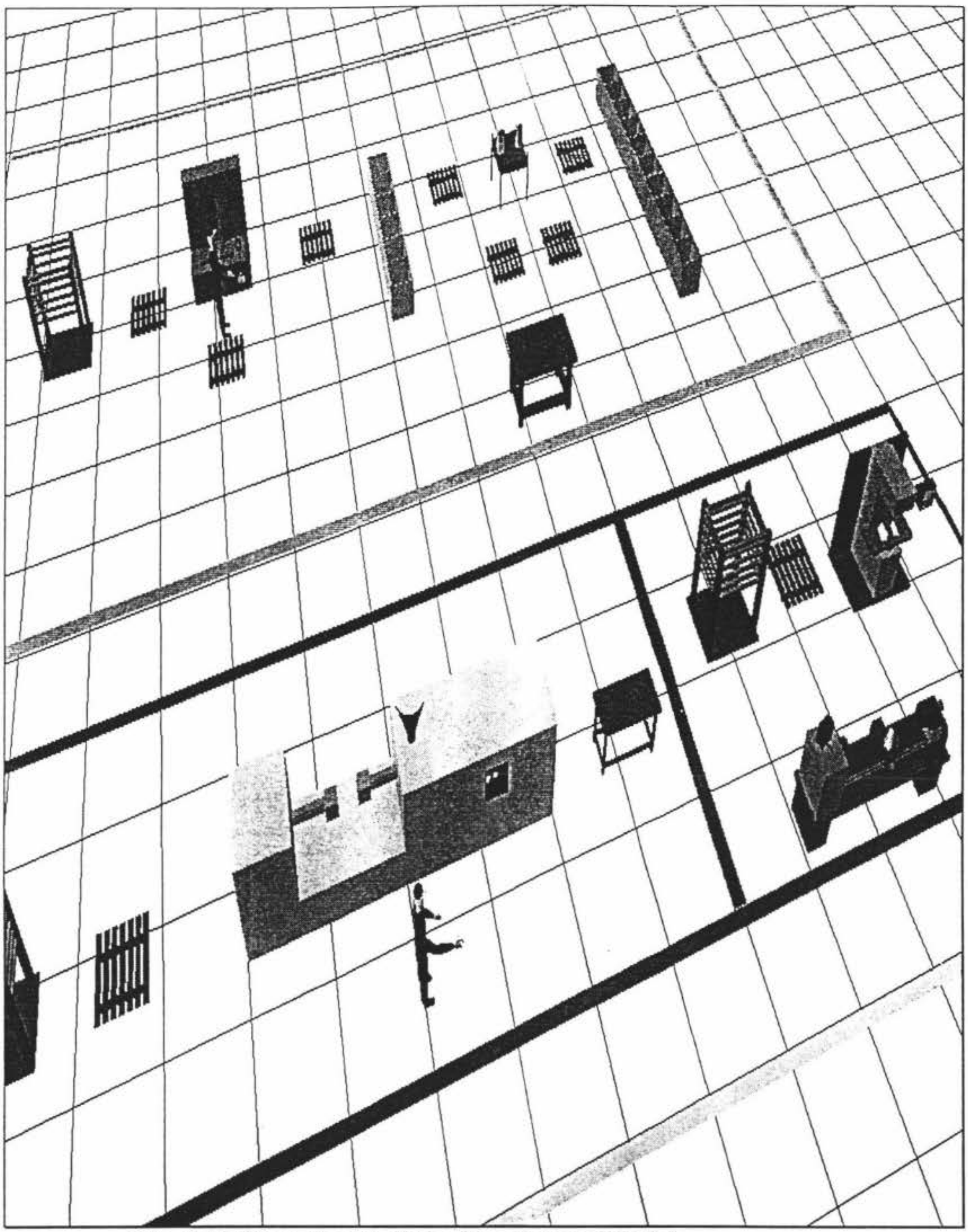
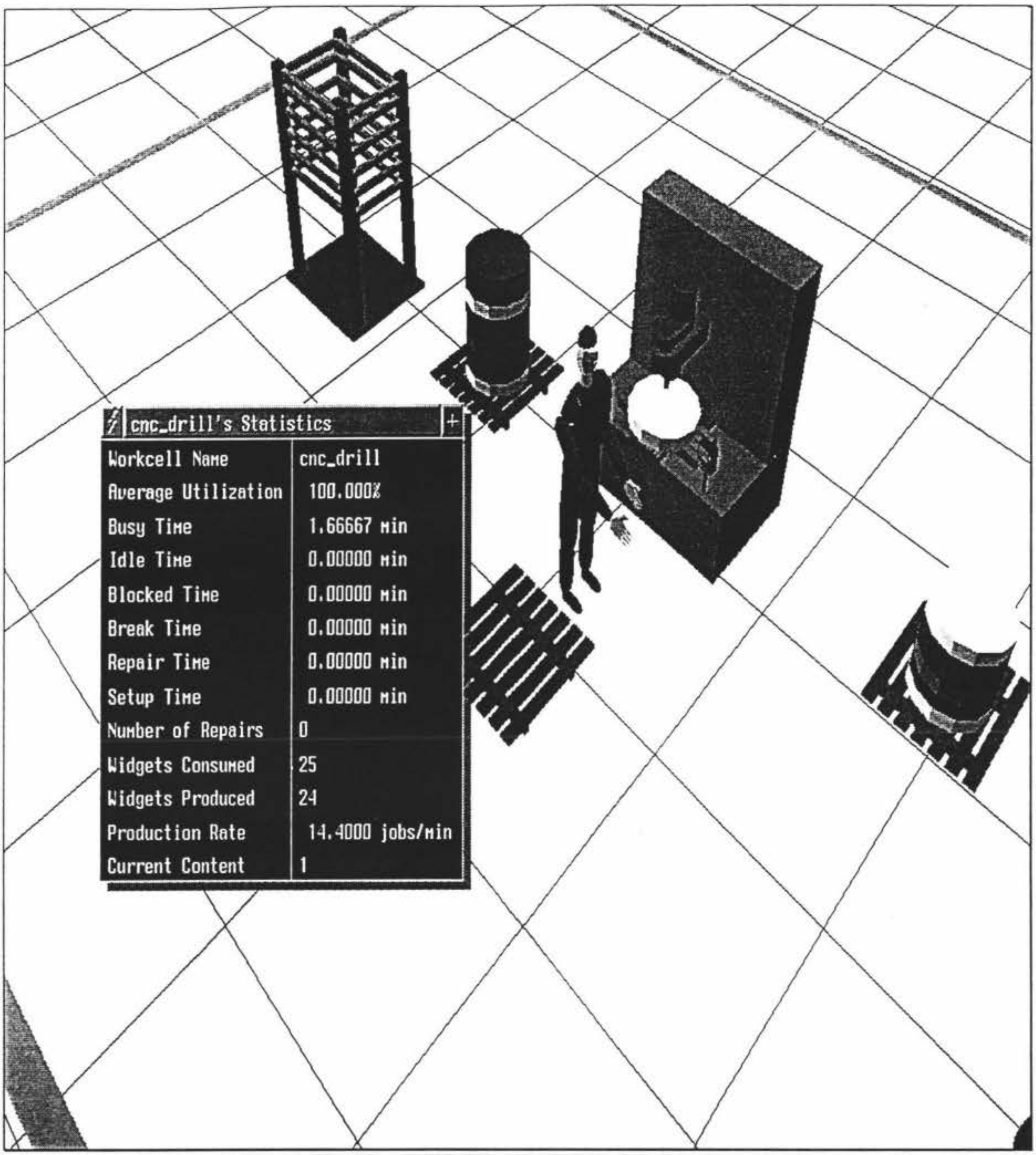


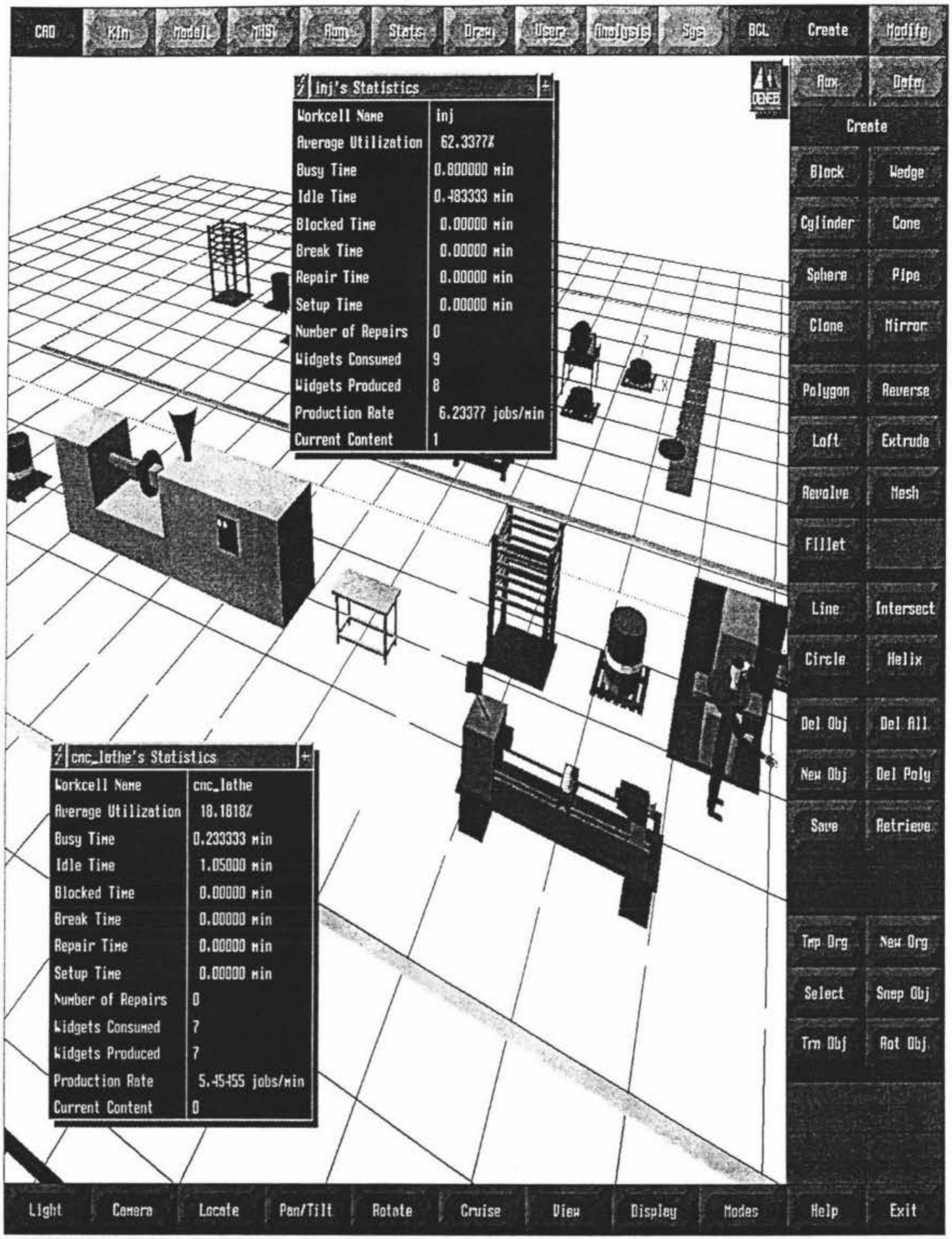
Fig. 4.12 Snapshot-1 of the Simulation Model



cnc_drill's Statistics

Workcell Name	cnc_drill
Average Utilization	100.000%
Busy Time	1.66667 min
Idle Time	0.00000 min
Blocked Time	0.00000 min
Break Time	0.00000 min
Repair Time	0.00000 min
Setup Time	0.00000 min
Number of Repairs	0
Widgets Consumed	25
Widgets Produced	24
Production Rate	14.4000 jobs/min
Current Content	1

Fig. 4.13 Snapshot-2 of the Simulation Model



inj's Statistics

Workcell Name	inj
Average Utilization	62.3377%
Busy Time	0.800000 min
Idle Time	0.483333 min
Blocked Time	0.00000 min
Break Time	0.00000 min
Repair Time	0.00000 min
Setup Time	0.00000 min
Number of Repairs	0
Widgets Consumed	9
Widgets Produced	8
Production Rate	6.23377 jobs/min
Current Content	1

cnc_lathe's Statistics

Workcell Name	cnc_lathe
Average Utilization	18.1818%
Busy Time	0.233333 min
Idle Time	1.05000 min
Blocked Time	0.00000 min
Break Time	0.00000 min
Repair Time	0.00000 min
Setup Time	0.00000 min
Number of Repairs	0
Widgets Consumed	7
Widgets Produced	7
Production Rate	5.45455 jobs/min
Current Content	0

Fig. 4.14 Snapshot-3 of the Simulation Model

Figures 4.12, 4.13, 4.14 show the snapshots of the Simulation Model developed in this research.

The following Chapter explains in detail the simulation experiments.

Chapter 5

SIMULATION EXPERIMENTS

This chapter explains about the hypothetical FMS modelled, the simulation experimental setup for the Machine Selection Rules, Part Selection Rules and the Machine Breakdown Logic. It also elaborates the Simulation Performance Criterion chosen for the experiments and the explanation of System Performance Measures, such as, Average Machine Utilisation, Mean Flow Time, Average Queue Length and System Production Rate and also the estimation of Steady-state condition.

5.1 Description of the Hypothetical FMS

An hypothetical FMS model is being developed with four cells and six jobs as shown in fig. 5.1. The model has nine machines, the details of which is explained below. The number of machines has no influence on the relative performance of the scheduling rules (Rowe, 1989).

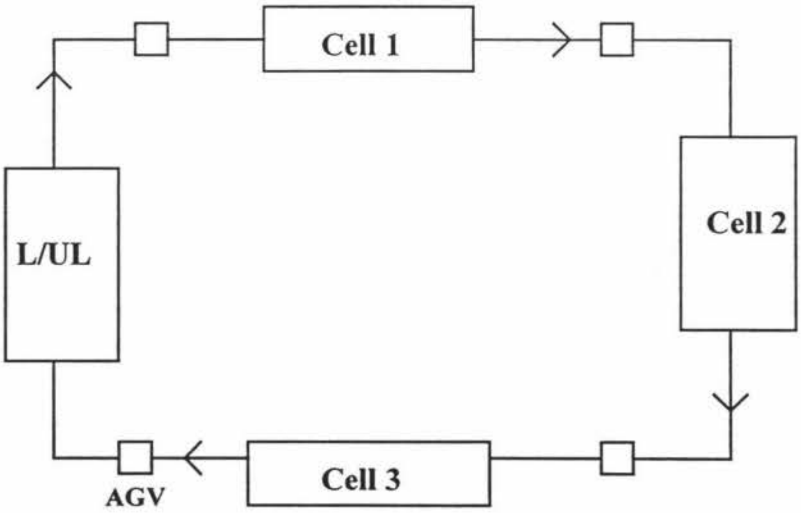


Fig. 5. 1 Configuration of the FMS model.

The FMS model designed has the following components :

- A loading/unloading cell(L/UL) which has two machines of different types. One machine loads the jobs on to the pallets before releasing them into the system and the other machine unloads the jobs off the pallets after the operations are completed on the jobs.

- A drilling cell(C1) which has two drilling machines (D1, D2) of same type with different tooling facility.

- A milling cell(C2) which has two milling machines (M1, M2) of same type with different tooling facility.

- A machining centre cell(C3) which has three different types of machining centres(MC1, MC2, MC3) with different tooling facility.

- A transportation system which has an AGV to transport the jobs through the system.

5.1.1 Assumptions made for the development of the FMS model

The following assumptions are made for the development of the FMS model:

- There is only one operation at a time on a machine.

- The setup times are included in the processing times for each machine type.

- Buffer storage of specified capacity is available at the input and output of each machine. Buffer capacity of 10 is fixed for all the machines in the system.

- The maximum input and output buffer capacities in each of the cells is five.

- The same pallets can be used for different jobs.

- The processing time of an operation is generated from exponential distribution with mean values (20, 25, 30 and 35 minutes).

- Batch size for each job is generated from a uniform distribution U[5,10].

- Loading and unloading times are generated from exponential distributions with mean values of 15 and 10 minutes, respectively.

- Transportation times are generated from exponential distributions with mean values of 10 minutes.

5.2 Simulation Experiments

Simulation experiments are designed to evaluate the various scheduling rules in the FMS. Basically the scheduling rules can be classified into two set of rules, a) part-selection rules and b) machine selection rules (Richard H. Choi et. al., 1988).

In this research, the following rules have been investigated to evaluate their effect on the system performance:

5.2.1 Part-selection rules

- FIFO(first in first out): The job types are selected in the same order as they arrive and then dispatched for further processing in the system.

- PRIORITY: Job types can be classified as high priority and low priority jobs based on the processing times or due-dates. Based on these priorities the jobs will be released into the system.

5.2.2 Machine selection rules

- RANDOM : Machines which are idle are selected in random order to perform the next operation.

- LUM : Machines which are least utilised.

- MINQ : Machines with minimum queue in terms of jobs waiting for service in the buffers in front of the machines.

- MINW : Machines which have the least work in queue in terms of process time.

5.3. Measurement of System Performance

Simulation data collected from the above mentioned experiments will be analysed to measure the system performance . The following indices is taken up for measuring the system performance:

- Average Machine Utilisation
- Mean Flow Time of jobs
- Average Queue Length(WIP)
- System Production Rate

5.4 Simulation Performance Criteria

In this research, the simulation performance criteria are set as follows : (Richard H. Choi et. al., 1988 and Rajan Suri et. al., 1984).

5.4.1 Average Machine Utilisation

$$\overline{MU_{ij}} = \frac{(TST - IT_{ij})}{TST} X 100 \qquad (5.\ 1)$$

where:

$\overline{MUij}$ = Utilisation of Machine *j* in Cell *i*

TST = Total Simulation Time

$ITij$ = Idle Time of Machine *j* in Cell *i*

Average Machine Utilisation in each Cell (U_w):

$$Uw = \frac{\sum_{k=1}^{n} \overline{MU_{ij}}}{n} \qquad (5.\ 2)$$

where '*n*' is the number of Machines in each Cell

Average Machine Utilisation in the hypothetical FMS :

$$U_{fms} = \frac{\sum_{k=1}^{w} U_w}{w} \qquad (5.\ 3)$$

where '*w*' is the number of Cells

5.4.2 Mean Flow Time

This is defined as the total time spent by a particular job type in the system.

$$\overline{JT}(i) = \frac{\sum_{j=1}^{pr(i)} (Cj - Aj)}{PR(ij)} \qquad (5.\ 4)$$

where:

$\overline{JT(i)}$ = the mean flow time under dispatch rule *i*

$C(j)$ = the completion time of job *j*

$A(j)$ = the arrival time of job *j*

$PR(ij)$ = the total production output of job *j* under dispatch rule *i* during the total simulation period

also, $C(j) = PT(j) + TT(j) + WT(j)$ *(5. 4a)*

where,

$PT(j)$ = the total processing time for job *j* which includes the operation time, setup time and the load/unload time

$TT(j)$ = the transportation time for job *j*

$WT(j)$ = the waiting time for job *j*

5.4.3 Average Queue Length [WIP] :

This is the number of jobs waiting in the buffers to be processed by the machines in the cell. This also includes the jobs under process on the machines at the time of completion of simulation.

5.4.4 System Production Rate (SPR):

This is defined as the ratio of the total production output of all the job types by the total time of production (i.e., total period of simulation).

$$SPR(i) = \frac{\sum_{j=1}^{N} PO_j}{TST(i)} \qquad (5.\ 5)$$

where :

$SPR(i)$ = the system production rate for the dispatching rule i
POj = the total production output for job type j
N = the number of job types introduced into the system
$TST(i)$ = the total simulation time for the dispatching rule i

5.5 Simulation Input Data

Table 5.1 gives the details of the routing of various jobs to the workstations in the model.

Table 5.2 shows the mean process time of each of the job types at each workstation.

JOB TYPE	ALT. ROUTE	WORKSTATION ROUTING
1001	1	L, C1(D1), C2(M1), UL
	2	L, C1(D2), C2(M2), UL
1002	1	L, C2(M1), C3(MC1), UL
	2	L, C2(M2), C3(MC2), UL
	3	L, C2(M1), C3(MC3), UL
1003	1	L, C3(MC1), UL
	2	L, C3(MC2), UL
	3	L, C3(MC3), UL
1004	1	L, C1(D1), C3(MC1), UL
	2	L, C1(D2), C3(MC2), UL
	3	L, C2(D2), C3(MC3), UL
1005	1	L, C1(D1), UL
	2	L, C1(D2), UL
1006	1	L, C2(M1), UL
	2	L, C2(M2), UL

Table 5.1 The job routing to the Workstations

JOB TYPE	ALT. ROUTE	MEAN PROCESS TIME AT EACH WORKSTATION (min)
1001	1	15, 25, 30, 10
	2	15, 30, 25,10
1002	1	15, 35, 35, 10
	2	15, 25, 30, 10
	3	15, 35, 25, 10
1003	1	15, 35,10
	2	15, 30,10
	3	15,25,10
1004	1	15, 25, 35,10
	2	15,30, 30, 10
	3	15, 30, 25,10
1005	1	15, 25, 10
	2	15, 30, 10
1006	1	15, 35, 10
	2	15, 25, 10

Table 5.2 The Mean Process Time of each job type at different Workstations.

5.6 Simulation Experimental Setup

Three sets of experiments have been done:

The first set of experiments are done to test the Machine Selection Algorithms as mentioned in section 5.2.2

The second set of experiments are done to test the Part Selection Algorithms as mentioned in section 5.2.1

The third set of experiments are done to verify the Machine Breakdown Algorithm.

5.7 Experimental Considerations

For all the experiments mentioned above the jobs were released into the system in random order and the mean time between arrivals(MTBA) follows an exponential distribution with a mean value of 30 minutes. It was observed that at this rate of release all the resources in the system were well balanced with load.

The jobs were released in small batches following a uniform distribution [5, 10] i.e., with a minimum batch size of 5 and a maximum batch size of 10.

5.7.1 Simulation Period

The simulation time for all the experiments was set at 40 hours. The warm-up period was set at 3 hours, so that the initialisation bias in collecting the statistical data can be removed. This is explained in the following section.

5.8 Estimation of Steady-state of the Simulation Model

A steady-state simulation analysis is concerned with the long run behaviour of the system. In most manufacturing lines the system is not cleared of parts at the end of the day. In these situations a steady-state analysis is appropriate.

The simulation model was started with the initial conditions that all the buffers in front of the machines were empty and the machines in idle condition. So, it would take some time before all the queues build up and all the machines in the system become busy , after which the simulation data can be collected to analyse the system performance measures. This initial conditions of the model (warm-up period) will create the bias in the results. To avoid the initialisation bias, the data collected during the warm-up period is removed from the data collected for final analysis.

The length of warm-up period is determined by plotting a graph between key output performance measure such as Average Queue Length of the bottleneck machine over time. This is shown in figure 5.2 below. It is observed from the graph that the initialisation warm-up period ends around 3 hours . This is when the system attains a steady-state. To remove the initialisation bias, the statistics should be cleared after 3 hours. Hence, the warm-up period is set to 3 hours for all the experiments.

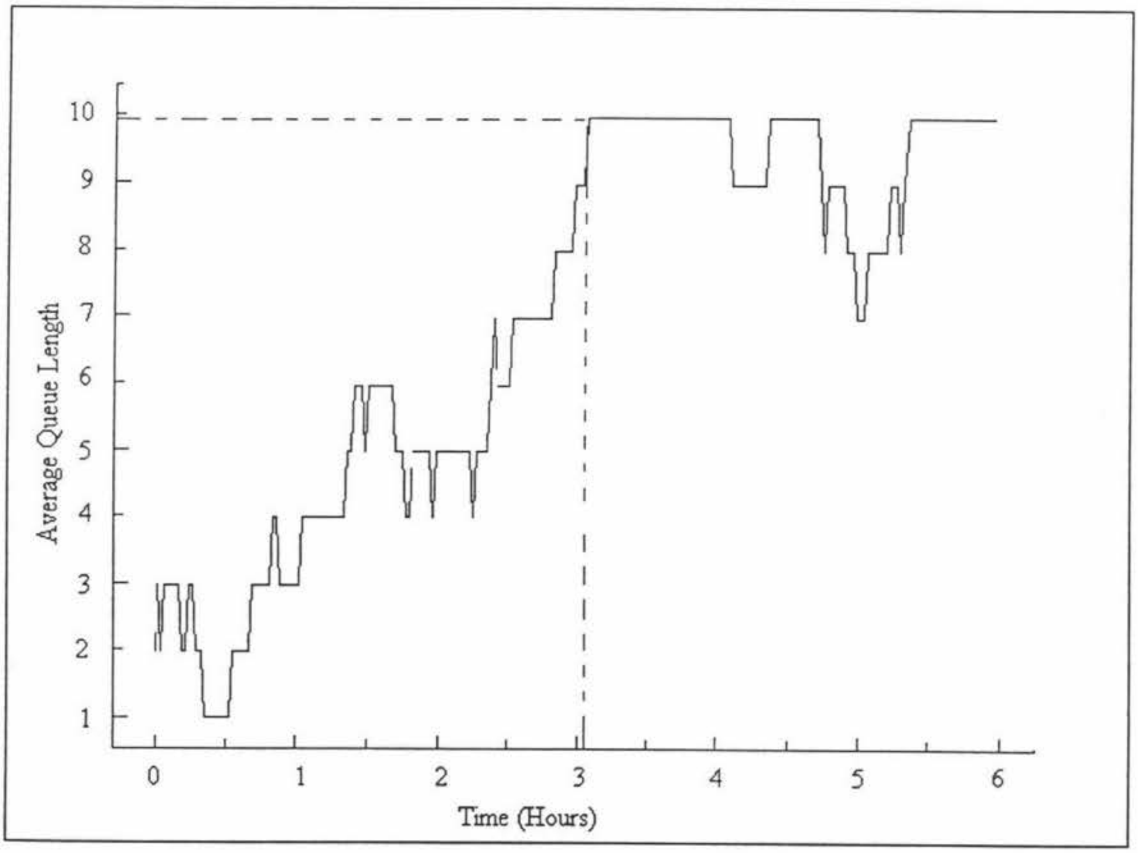


Fig. 5.2 Number of Jobs waiting for processing over time.

The simulation results and analysis are discussed in the following chapter.

Chapter 6

RESULTS AND DISCUSSIONS

This chapter covers the analysis of System Performance Measures which have been used for this research i.e., Average Machine Utilisation, Mean Flow Time of jobs, Average Queue Length (WIP) and the System Production Rate and how these performance measures are affected by the Machine Selection Rules & Part Selection Rules. It also discusses how the system can take care of most likely system disturbance like Machine Breakdown.

6.1 Simulation Results and Analysis

Simulation data which is collected from all the experiments is tabulated and put in Appendix C. For each of the experiments five simulation runs are conducted to check the consistency of the data obtained. It has been observed that the data has been consistent throughout all the experiments.

6.2 Analysis of Machine Selection Rules

The jobs are released into the system by using the FIFO algorithm i.e., the job which arrives first is released first into the system. The next machine, where the next task will be performed, is selected by one of the four Machine Selection algorithms i.e., RANDOM, LUM, MINQ, MINW. For comparing their performances, the four system performance measures as mentioned above were considered. Table 6.1 shows the relative performance of the four rules.

MACHINE SELECTION RULE	AVERAGE MACHINE UTILISATION	MEAN FLOW TIME	AVERAGE QUEUE LENGTH	SYSTEM PRODUCTION RATE
RANDOM	1	3	3	1
LUM	4	2	4	3
MINQ	3	4	2	1
MINW	2	1	1	2

Table 6. 1 Relative Ranks of Machine Selection Rules on the System Performance Measures.

The MINW algorithm shows the overall best performer when compared to the other algorithms. However, if mean flow time of jobs is considered, the MINQ algorithm performed the best. For the system performance, system production rate, both LUM and MINW have performed well. If the overall performance is taken into consideration, MINW performed the best and the LUM performance is very poor. This compares well with the results obtained by O'Keefe and Kasirajan (1992) and Choi and Malstrom (1988).

Fig. 6.1, to 6.10 show the graphs plotted with the results obtained for the four Machine Selection Rules against the system performance measures, Average Machine Utilisation, Mean Flow Time, Average Queue Length, System Production Rate.

As seen in fig. 6.1 the average cell utilisation of the model varies from 74.17% to 81.10% for the four rules. So, it can be concluded that the cells are evenly balanced with the loading of jobs. Also, the machine utilisation is best with the Random rule and worst with the LUM rule.

Fig. 6.2 to 6.7 show the variation in mean flow times with the four rules for all the six job types. Fig. 6.8 shows the mean weighted flow times with the four rules. Mean weighted flow time is the combined measure of flowtime across all the job types. It can be observed that the MINW rule is the best performer with the lowest value of

497.44 minutes and the MINQ rule is the poorest performer with the highest value of 619.21 minutes. The reason for MINW performing well is that the jobs have minimum waiting times at the machines. According to this rule, the machine whose buffer has the lowest work in terms of processing time is selected for processing the jobs.

Fig. 6.9 shows the Average Queue Lengths (Work in Process Inventory) for all the four rules. Again it can be observed that the MINW is the best performer followed by MINQ, RANDOM and LUM in that order. Since lower waiting times for jobs indicate faster processing of jobs and hence, lower work in process inventory. Therefore, MINW has given the best result.

Fig. 6.10 represents the performance of the four rules for the measure of System Production Rate. RANDOM and MINQ have given the highest production rates while LUM gave the lowest production rate.

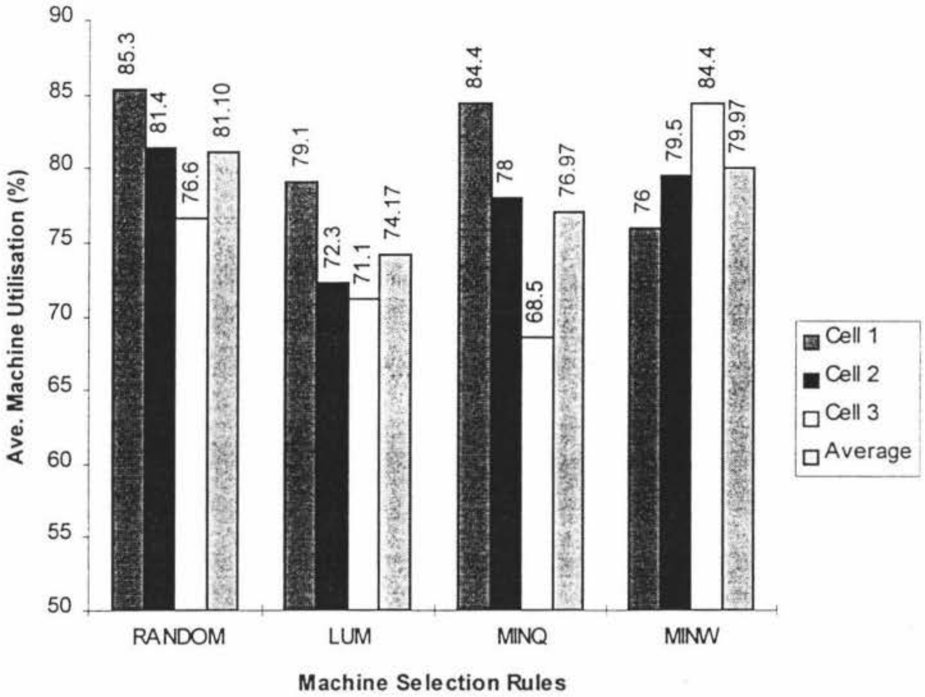


Fig. 6.1 Machine Selection Rules vs Machine Utilisation

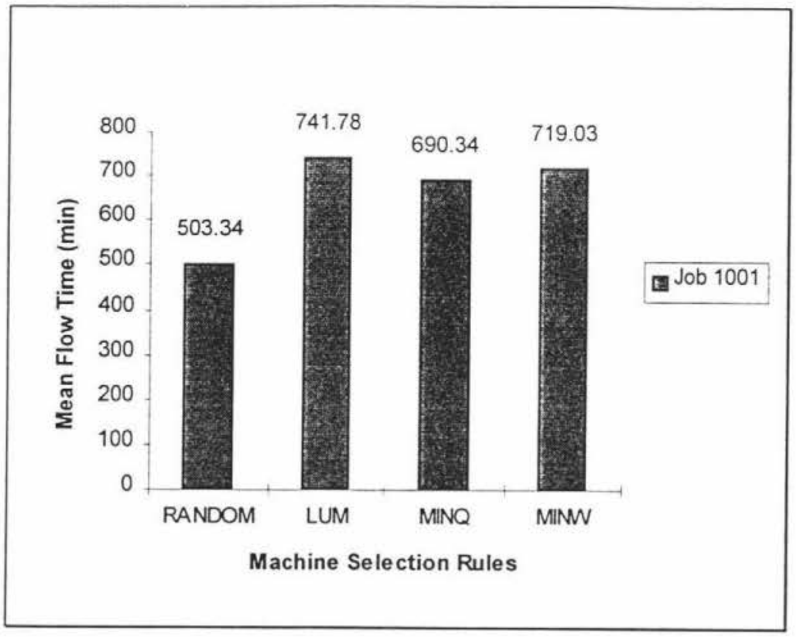


Fig. 6.2 Machine Selection Rules vs Mean Flow Time (Job 1001)

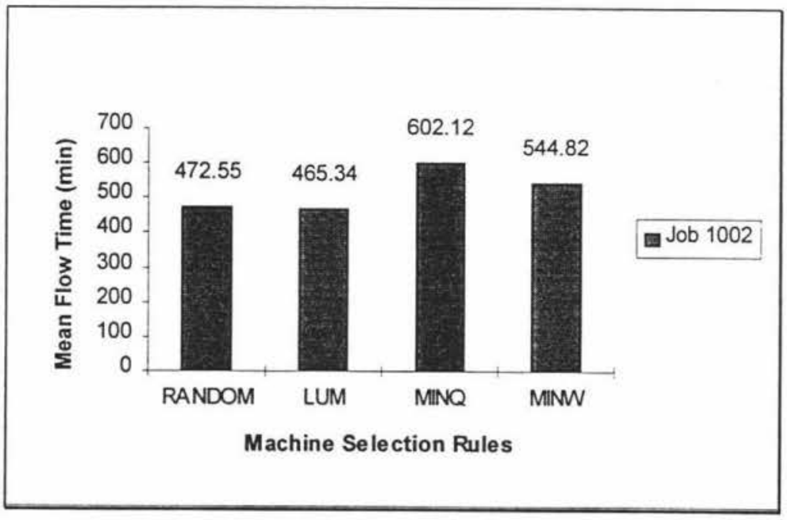


Fig. 6.3 Machine Selection Rules vs Mean Flow Time (Job 1002)

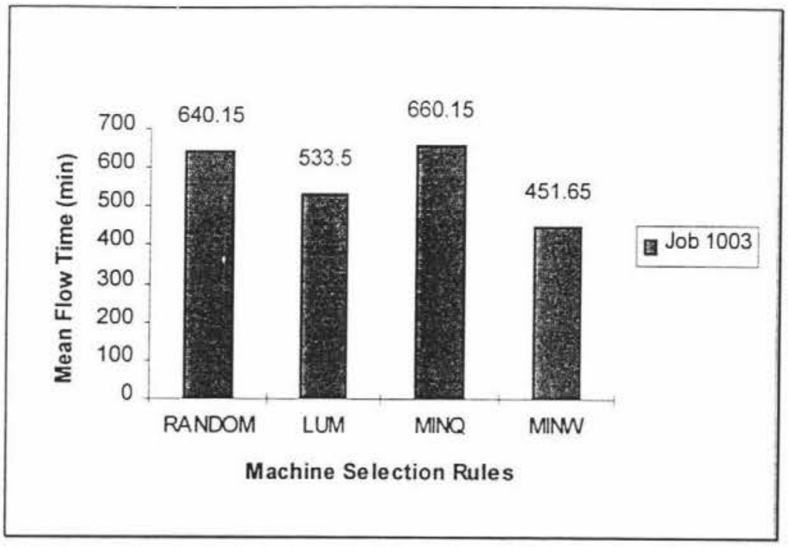


Fig. 6.4 Machine Selection Rules vs Mean Flow Time (Job 1003)

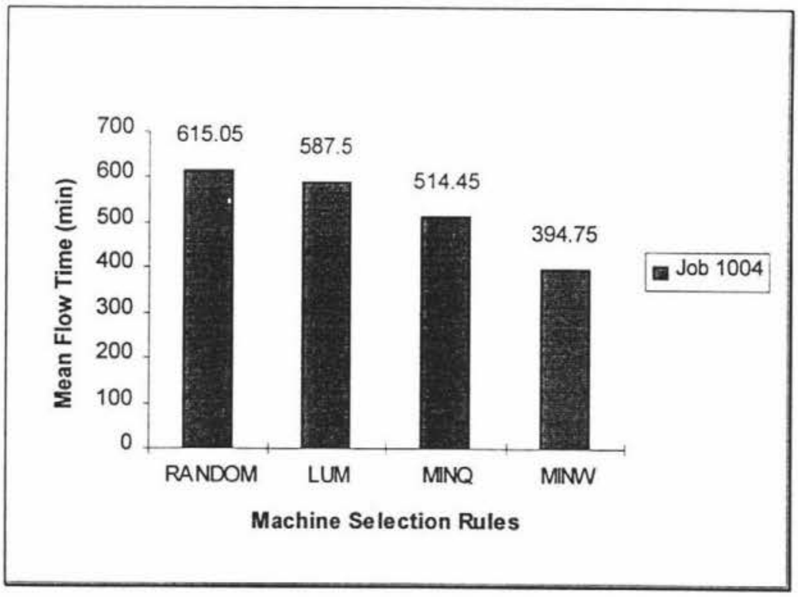


Fig. 6.5 Machine Selection Rules vs Mean Flow Time (Job 1004)

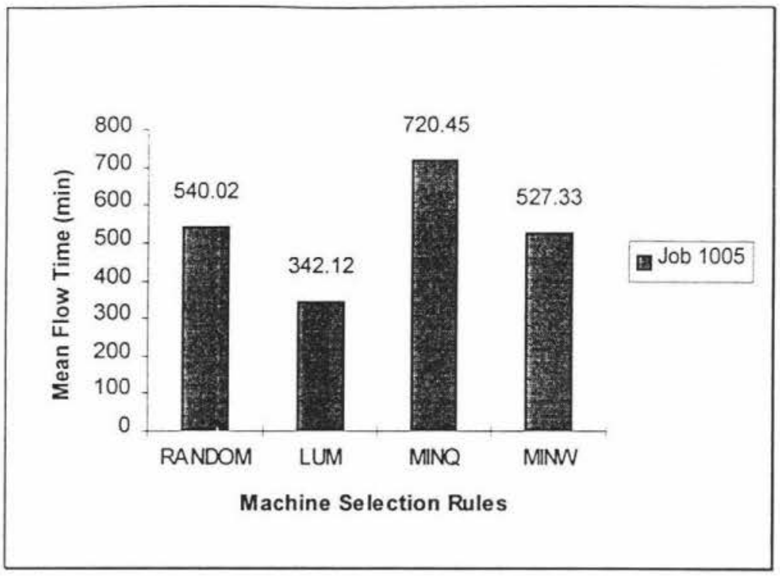


Fig. 6.6 Machine Selection Rules vs Mean Flow Time (Job 1005)

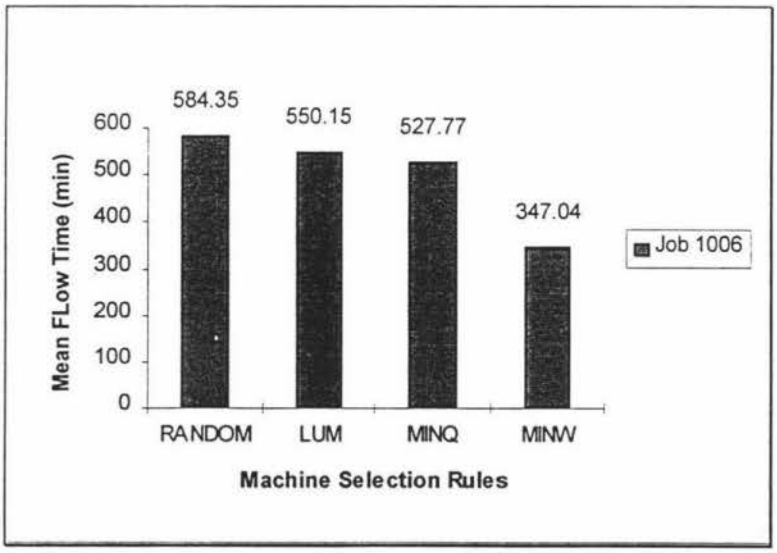


Fig. 6.7 Machine Selection Rules vs Mean Flow Time (Job 1006)

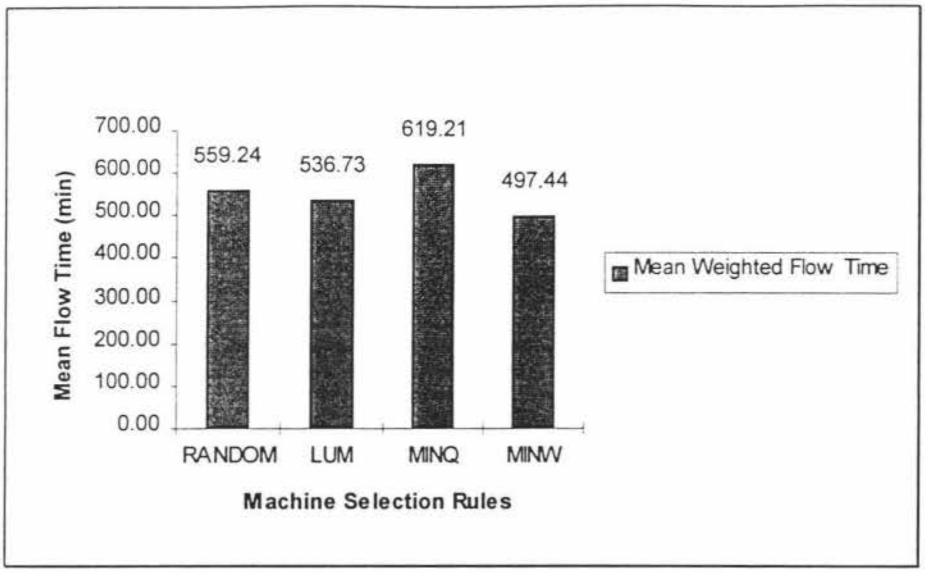


Fig. 6.8 Machine Selection Rules vs Mean Weighted Flow Time of all the Jobs

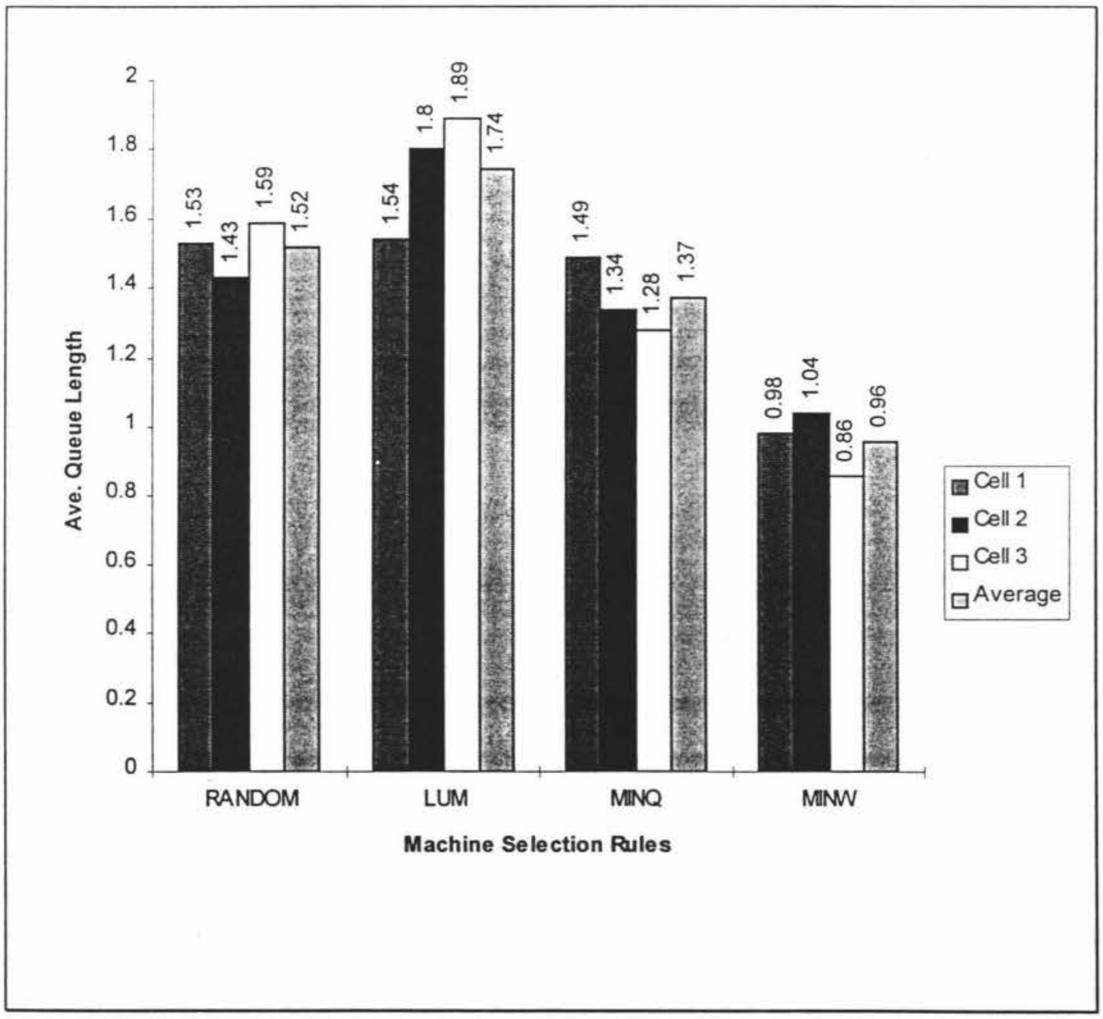


Fig. 6.9 Machine Selection Rules vs Average Queue Length

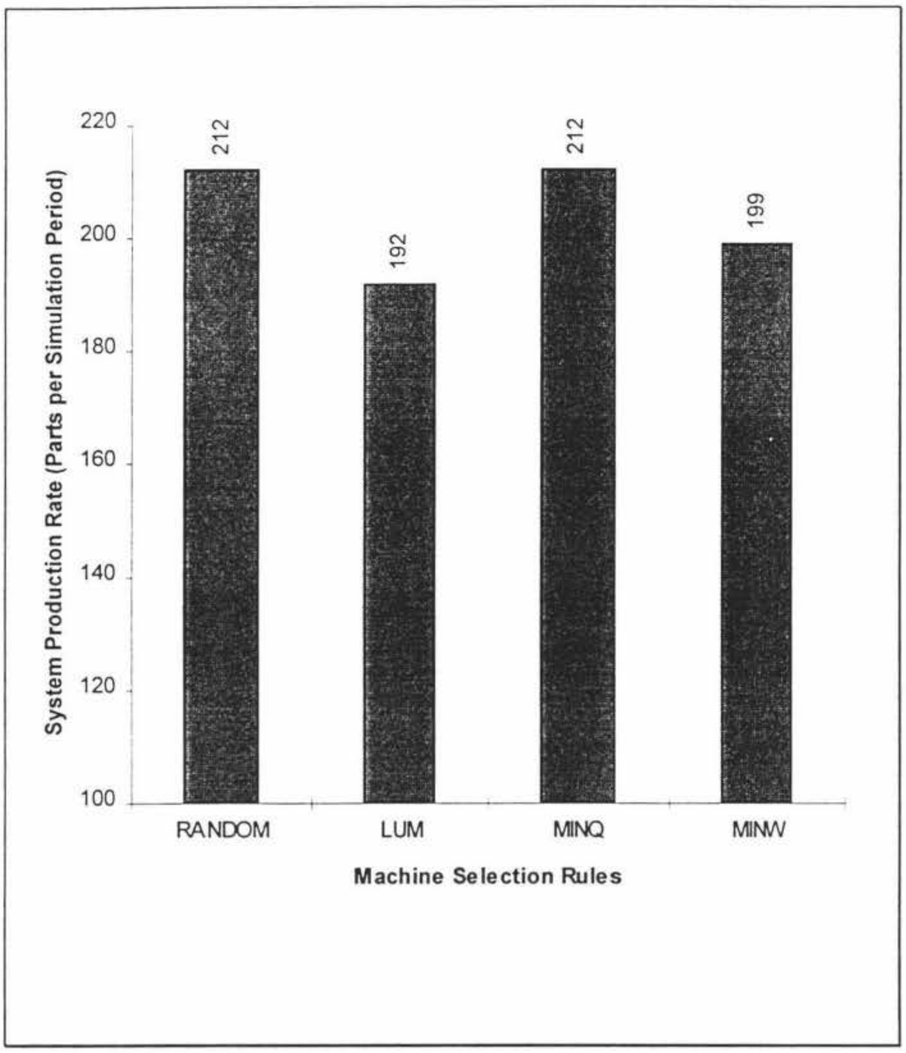


Fig. 6.10 Machine Selection Rules vs System Production Rate

6.3 Analysis of Part Selection Rules

FIFO and PRIORITY are the part selection rules used in this research. The jobs that arrive into the system and cannot find any idle machines for processing will form a queue in the buffers in front of the machines. The jobs in the queue are prioritised according to the FIFO and PRIORITY algorithms. According to the FIFO rule the jobs that arrive first are selected first to be released into the system for processing. According to the PRIORITY rule, each of the job types is given a priority number based on the jobs due-dates and processing times etc., and then the job with the highest priority number will be released first into the system for processing.

To test these rules, the job types 1001, 1002, and 1005 have been given high priority and job types 1003, 1004, and 1006 have been given low priority, before releasing them into the system. The results obtained are plotted as graphs in fig. 6.11 and 6.12. Fig 6.11 shows the comparison of the system performance measure production rates for the two rules. It can be observed that the production rates of high priority jobs has increased when compared to the FIFO rule and the production rates for low priority jobs has decreased or remained the same as in the case of job type 1006.

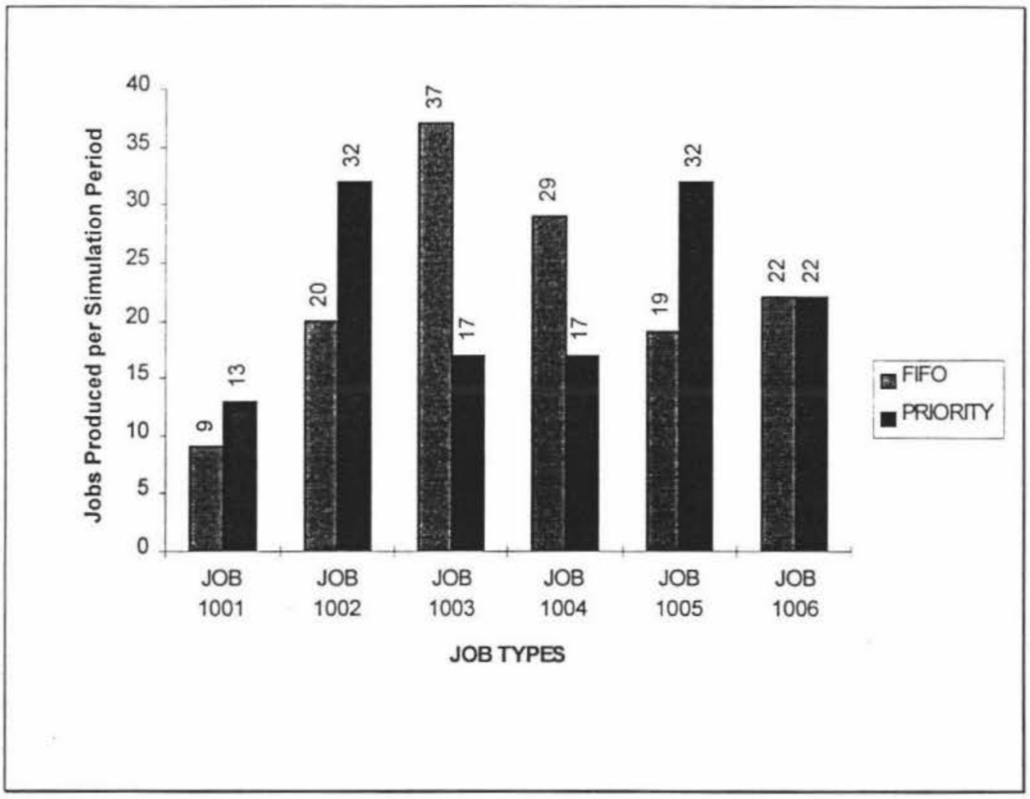


Fig. 6.11 Production Rates for Part Selection Rules

Also, it can be concluded from fig. 6.12 that the mean flow time for the high priority jobs has decreased when compared to the low priority jobs. This indicates that the PRIORITY rule is functioning as desired.

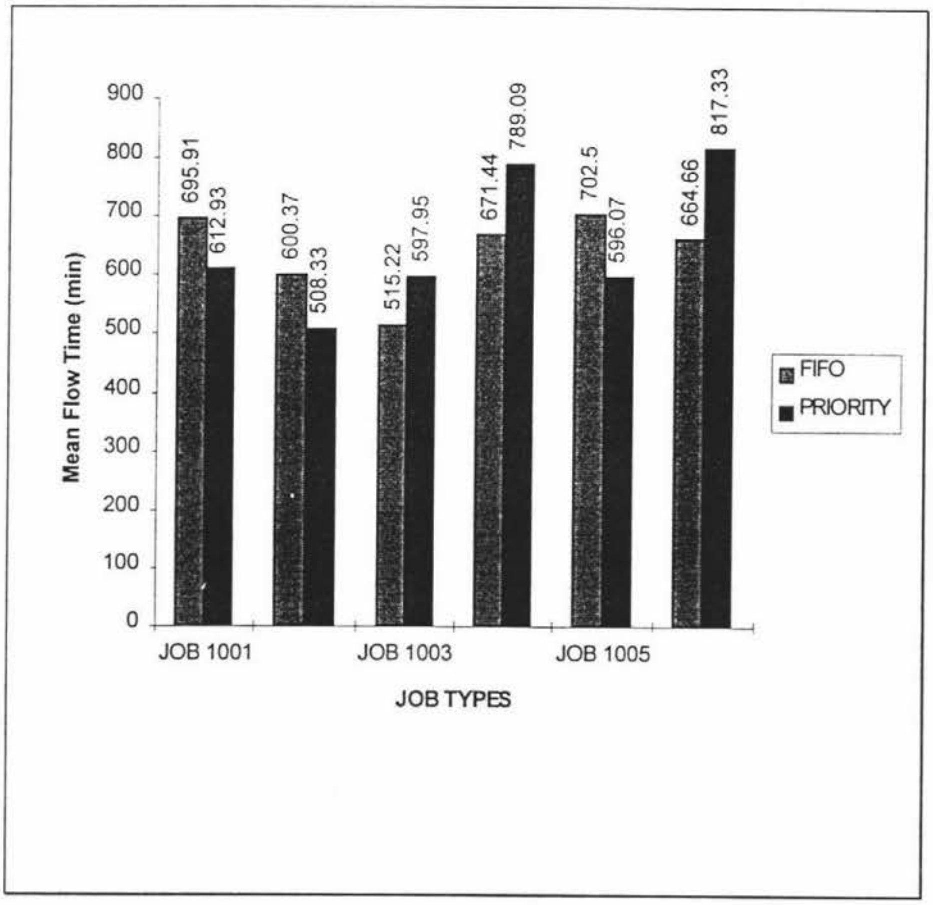


Fig. 6.12 Comparison of Mean Flow Times for the Part Selection Rules

6.4 Analysis of Machine Breakdown Algorithm

Machine Breakdown Algorithm allows the user to simulate the real machine breakdown scenario in a manufacturing facility. This can be achieved by incorporating this algorithm in the process logic of any selected machine in the model. To test this algorithm, it has been assigned to machine D2 of Cell 1. The breakdown time can be set as per the user's requirement. In this case it has been set to half the simulation time i.e., the machine D2 does not process any jobs during that period. However, the jobs are diverted to machine D1 in the same Cell with the help of machine selection decision making logics as explained earlier. This can be observed from the fig. 6.13 that the utilisation of machine D1 has increased and that of D2 has decreased. Also, the subsequent utilisation of machines M1, M2, MC1, MC2, MC3 in the model has decreased. This is because of the dependency of these machines on the output of machine D2, as is evident from the job routeing sequence explained in Chapter 5.

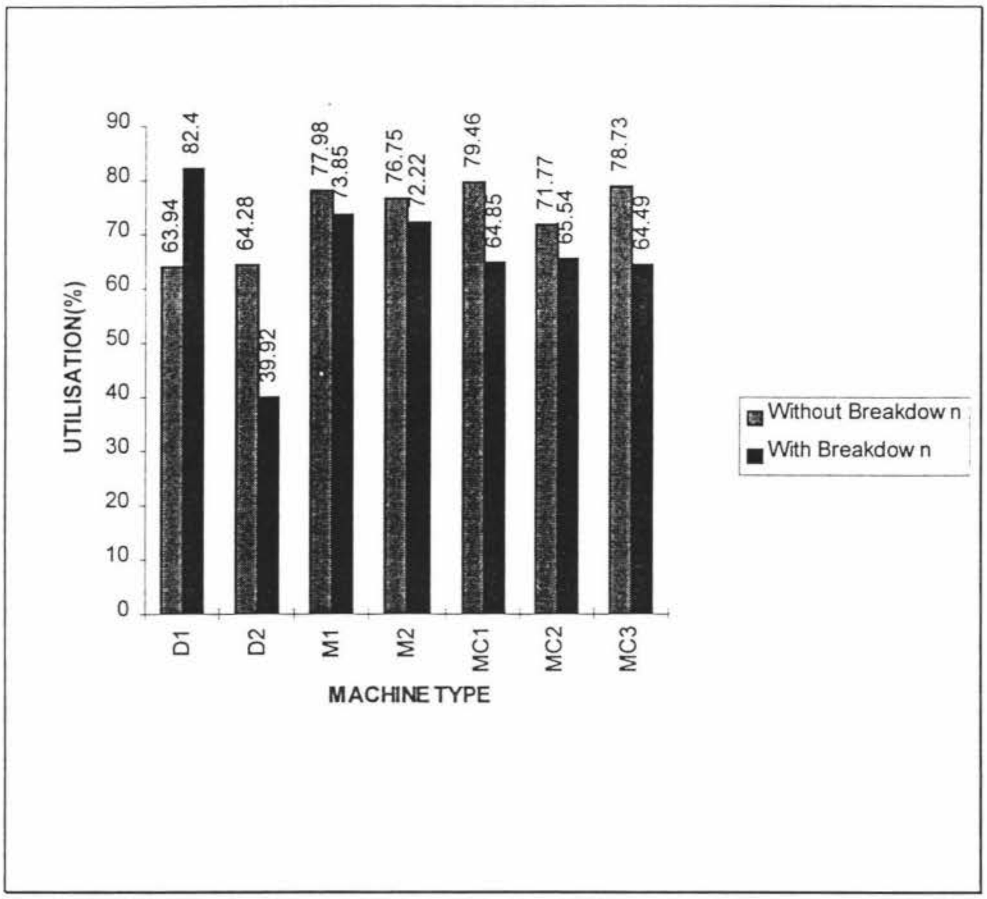


Fig. 6.13 Machine Breakdown Analysis

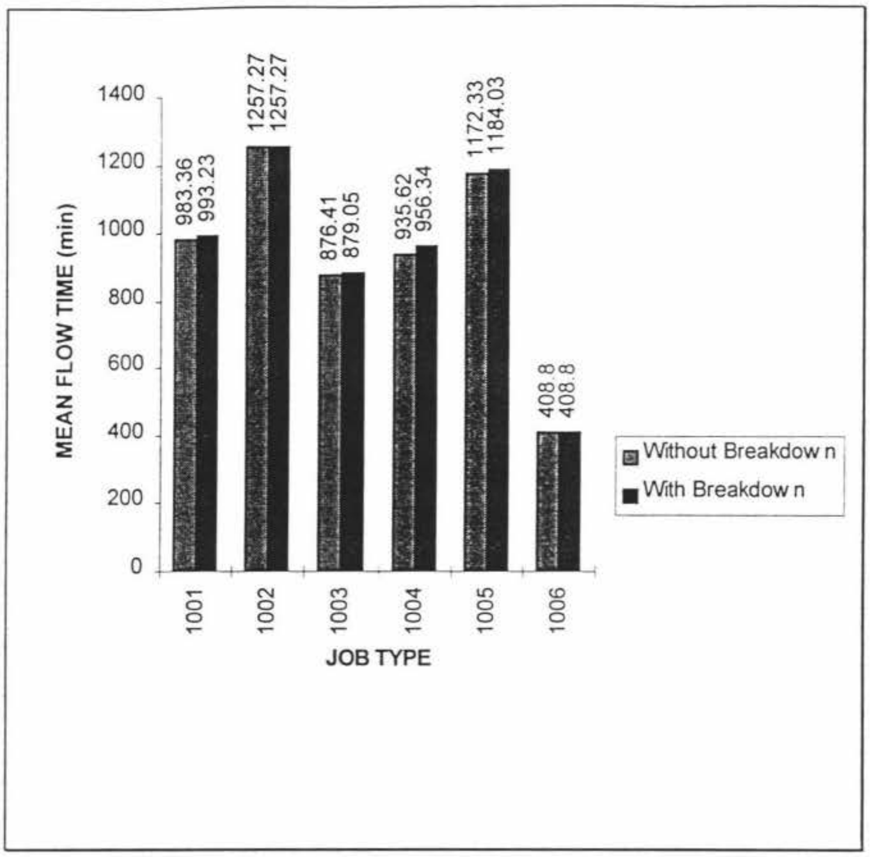


Fig. 6.14 Effect of M/c Breakdown on Mean Flow Time

The effect of machine breakdown on mean flow time of the jobs has also been studied and it can be seen from fig. 6.14 that the machine breakdown has little or no influence on the mean flow time of the jobs. This indicates that the breakdown logic combined with the machine selection logics and the flexible (multiple) routeing of jobs helps in maintaining the stability of the system even though it is affected by disturbances like machine breakdown.

The following Chapter draws the conclusions and achievements from this research.

Chapter 7

CONCLUSIONS

This Chapter presents the conclusions drawn from this research.

This thesis has given an overview of the new concepts like Virtual Manufacturing Systems and Virtual Manufacturing in general. Various areas of manufacturing have been identified for the application of Virtual Manufacturing software. These software tools helps the engineers visualise and solve the problems in various areas of manufacturing activities such as, Product Design, Process Design, Production Planning, Scheduling, Quality Control, Logistics and Cost Estimation.

A Virtual Reality Software tool QUEST® has been used to realise the concept of Virtual Manufacturing and to achieve the research objectives. Manufacturing Scheduling is an important and complex problem. A lot of research is being done in this area. Most of the research is being done in analysing a large number of traditional scheduling rules particularly part dispatching rules. However, not much attention is being given in studying the equally important machine selection rules. Hence, this research concentrated in this aspect of scheduling.

Various Process logics, Route logics, Queueing logics have been written in Simulation Control Language, to analyse the performance of various machine selection rules in a manufacturing facility. An hypothetical FMS (Flexible Manufacturing System) has been modelled to study the performance of these rules. The system performance measures Average Machine Utilisation, Mean Flow Time, Average Queue Lengths and Production Rates are measured for these rules.

The results obtained from the simulation study are positive and encouraging as discussed in Chapter 6. However, each of the machine selection rules have their merits and de-merits. The selection of machine selection algorithms depends on the objective. If the objective is to minimise the mean flow time of the jobs and reduce work in process inventory then MINW algorithm can be used. If the objective is to achieve high production rates for the jobs then RANDOM or MINQ algorithms can be used.

Also the Virtual Manufacturing System concept, that can be realised by using this Virtual Reality Software QUEST®, can be used to optimising the design of manufacturing system layout. So this research has identified that Virtual Manufacturing Software is a powerful tool by which the manufacturing system designers can identify optimal configurations and highlight the potential problems before building the expensive manufacturing systems physically.

REFERENCES

Alex Pentland & John Williams (1989), "Virtual Manufacturing", *Proceedings of NSF Engineering Design Research Conference*, June 11-14, 1989, pp 301-311.

Bennett G.R. (1995), "Virtual Reality Simulation bridges the gap between Manufacturing & Design", *Mechanical Incorporated Engineer*, April/May - 1995, pp 43-47.

Carolina Cruz-Niera (1993), "Virtual Reality Overview", ACM SIGGRAPH '93 Course Notes: Applied Virtual Reality, *Proceedings of ACM SIGGRAPH '93 Conference on Computer Graphics*, California, August 1-6 1993.

Charles Parks, David Koonce, Luis Rabelo, Robert Judd & John Sauter (1994), "Model-Based Manufacturing Integration: A Paradigm for Virtual Manufacturing Systems Engineering", *Proceedings of the 16th Annual Conference on Computers & Industrial Engineering*, vol. 27, pp 357-360.

David Kahaner (1993), "Virtual Reality in Japan", *IEEE Micro*, vol. 13, pp 66-73.

Dean McCarty, Steven Sheaby, Philip Amburn, Martin Stytz and Chip Switzer (1994), "A Virtual Cockpit for a Distributed Interactive Simulation", *IEEE Computer Graphics & Applications*, vol. 14, pp 49-54.

Deneb Robotics, Inc., (1994), "QUEST® Batch Control Language (BCL) Reference Manual", *Deneb Robotics, Inc.,* Michigan, USA.

Deneb Robotics, Inc., (1994), "QUEST® Simulation Control Language (SCL) Reference Manual", *Deneb Robotics, Inc.,* Michigan, USA.

Deneb Robotics, Inc., (1994), "QUEST® User Manual and Tutorials", *Deneb Robotics, Inc.,* Michigan, USA.

F. Kimura, H. Suzuki, M. Inui, M. Ranta (1991), "Product Feature Modelling for Integrating Product Design and Process Planning", *Manufacturing Systems*, vol. 20, no. 4, pp 335-341.

Felix T.S. Chan (1995), "Using simulation to predict system performance: a case study of an electro-phoretic deposition plant", *Integrated Manufacturing Systems*, vol. 6, no. 5, pp 27-38.

Fumihiko Kimura (1993), "Product and Process Modelling as a Kernel for Virtual Manufacturing Environment", *Annals of the CIRP*, vol. 42/1/1993, pp 147-150.

Fumihiko Kimura (1993), "Virtual Manufacturing as a Basis for Concurrent Engineering", *Proceedings of IFIP Conference on World Class Manufacturing 1993,* pp 103-117.

G.R. Bennett (1995), "Virtual Reality Simulation bridges the gap between Manufacturing and Design", *Mechanical Incorporated Engineer*, April/May 1995, pp 43-47.

Gary Bayliss, Adrian Bowyer, Richard Taylor, & Phillip Willis (1994) "Virtual Manufacturing", *Proceedings of CSG 94 - Set-theoretic Solid Modelling Techniques & Applications*, Winchester, UK, April 13-15, 1994.

Gilad Lederer (1995), "Making Virtual Manufacturing a Reality", *Industrial Robot*, vol. 22, no. 4, pp 16-17.

Gunter Wittenberg (1995), "Training with virtual reality", *Assembly Automation*, vol. 15, no. 3, pp 12-14.

Iwata K., Onosato M., Teramoto K., and Osaki S. (1995), "A Modelling and Simulation Architecture for Virtual Manufacturing Systems", *Annals of the CIRP*, vol. 44/1/1995, pp 399-402.

J. J. Mills, B. Huff, T. Criswell & J. Graham (1992), "The Virtual Manufacturing Workstation", *Proceedings of 4th Conference on Flexible Assembly Systems*, Scottsdale, Arizona, Sept. 13-16, 1992, vol. 48, pp 35-39.

Jean V. Owen (1994) "Making Virtual Manufacturing Real", *Manufacturing Engineering*, November-1994, pp 33-37.

John A. Adam (1995), "Medical Electronics", *IEEE Spectrum*, v 32, pp 80-83.

John Latta and David Oberg (1994), "A Conceptual Virtual Reality Model", *IEEE Computer Graphics & Applications*, vol. 14, pp 23-29.

Katsundo Hitomi (1994), "Analysis and Design of Manufacturing Systems", in Richard Dorf & Andrew Kusiak (Ed.,) "Handbook of Design, Manufacturing and Automation", *John Wiley & Sons, Inc. Publications*, New York.

Kevin Warwick, John Gray & David Roberts (1993), "Virtual Reality in Engineering", *IEE Publications*, UK.

M. Montazeri and Van Wassenhove (1990), "Analysis of scheduling rules for an FMS", *International Journal of Production Research*, vol 28. no. 4, pp 785-802

Mark Bolas (1994), "Human Factors in the Design of an Immersive Display", *IEEE Computer Graphics & Applications*, vol. 14, pp 55-59.

Masahiko Onosato & Kazuaki Iwata (1993), "Development of a Virtual Manufacturing system by integrating Product Models and Factory Models", *Annals of the CIRP*, vol. 42/1/1993, pp 475-478.

Michael Puttre (1992), "Virtual Prototypes Move Alongside Their Physical Counterparts", *Mechanical Engineering*, August 1992, pp 59-61.

Min Hee Kim & Yeong-Dae Kim (1995), "Simulation-Based Real-Time Scheduling in a Flexible Manufacturing System", *Journal of Manufacturing Systems*, vol. 13, no. 2 pp 85-93

Muramatsu R., Ishi K. and Takahashi K. (1995), "Flexbility in Pull and Push Type Production Ordering Systems - Some ways to Increase Flexibility in Manufacturing Systems", in Raouf & Ben-Daya (Ed.,) "Flexible Manufacturing Systems : Recent Developments", *Elsevier Science Publications*.

Neil Duffie & Vittaldas Prabhu (1995), "Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems", *Journal of Manufacturing Systems*, vol. 13, no. 2, pp 94-107

Nigel R. Greenwood (1988), "Implementing Flexible Manufacturing Systems", *Macmillan Education Ltd. Publications*.

Phillip Willis (1993), "Virtual Manufacturing", *Proceedings of International Workshop on Graphics and Robotics*, Schloss dagstuhl, Germany, April 19-22, 1993.

PJ O'Grady (1986), "Controlling Automated Manufacturing Systems", *Chapman and Hall Publications*.

Rajan Suri & Richard R. Hildebrant (1984), "Modelling FMS Using Mean-Value Analysis" , *Journal of Manufacturing Systems*, vol 3, no. 1.pp 27-38

Raouf & M. Ben-Daya (1995), "Flexible Manufacturing Systems - Recent Developments", *Elsevier Science B.V. Publications*.

Richard H. Choi & Eric M. Malstrom (1988), "Evaluation of Traditional Work Scheduling Rules in a Flexible Manufacturing System with a Physical Simulator", *Journal of Manufacturing Systems*, vol. 7, no. 1, pp 33-45

Robert N. Tomasik, Peter B. Luh, and Guandong Liu (1996), "Scheduling Flexible Manufacturing Systems For Apparel Production" , *IEEE Transactions on Robotics & Automation*, vol. 12, no. 5, pp 789-799

Robert O'Keefe and Kasirajan (1992), "Interaction between dispatching and next station rules in a dedicated flexible manufacturing system" *International Journal of Production Research*, vol. 30, no. 8, pp 1753-1772

Rowe, A.J. (1989), "Sequential Decision Rules in Production Scheduling", *Ph.D. Dissertation*, UCLA.

S. A. Irani, T. M. Cavalier & P. H. Cohen (1993), "Virtual Manufacturing Cells : exploiting layout design and intercell flows for the machine sharing problem", *International Journal of Production Research*, vol. 31, no. 4, pp 791-810.

S. K. Goyal, Kapil Mehta, Rambabu Kodali and S.G. Deshmukh, (1995); "Simulation for analysis of scheduling rules for a flexible manufacturing system", *Integrated Manufacturing Systems*, Vol. 6 No. 5, pp. 21-26

Sandy Ressler (1994), "Applying Virtual Environments to Manufacturing", *National Institute of Standards and Technology*, USA.

Stewart Robinson (1993), "The Application of Computer Simulation in Manufacturing", *Integrated Manufacturing systems*, vol. 4, no. 4, pp 18-23.

Sudershan Jetley, Dominic Selven & Tirumala Malisetty (1994), "Virtual Manufacturing", *The Journal of Industrial Technology*, vol. 10, no. 4, pp 2-4.

Szu-Yung David Wu and Richard A. Wysk (1989), "An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing", *International Journal of Production research*, vol 27, no.9, pp 1603-1623

Talla Ajay & Saeid Nahavandi (1996), "Study & Analysis of Logictics of a Production System affected by a change in product features in a Virtual Manufacturing Environment", *Proceedings of the third New Zealand Conference of Postgraduate Students in Engineering & Technology*, pp 235-238.

Van looveren A.J., Gelders L.F. and Van Wassenhove (1986), "A review of FMS Planning Models", in A. Kusiak (Ed.,) "Modelling and Design of Flexible Manufacturing Systems", *Elsevier Science Publications*, The Netherlands.

Yash P. Gupta, Mahesh C. Gupta and C. R. Bector (1989), "A review of scheduling rules in flexible manufacturing systems" , *International Journal of Computer Integrated Manufacturing*, V 2, N 6, pp 356-377

BIBLIOGRAPHY

Ahmed K. Elshannawy, Ganesh M. Krishnaswamy and Mansooreh Mollaghasami (1993), "Concurrent Engineering Deployment : A Virtual Reality Approach", *Integrated Manufacturing Systems*, vol. 4, no. 4, pp 24-28.

Alan A. Desrochers (1990), "Modelling and Control of Automated Manufacturing Systems", *IEEE Computer Society Press Publications*.

Andrew Kusiak (1986), "Modelling and Design of Flexible Manufacturing Systems", *Elsevier Science B.V. Publications.*

Christian Altmann, Ines Kittel and Fumihiko Kimura (1994), "Estimation of Production Costs Using Virtual Monitoring : A Conceptual Framework", *Journal of Design and Manufacturing*, 1994, vol. 4, pp 187-194.

Conway R., William Maxwell and Louis Miller (1967), "Theory of Scheduling", *Addison-Wesley Publications*.

Corke D.K. (1987), "Production Control in Engineering", *Edward Arnold Publications.*

David J. Williams and Paul Rogers (1991), "Manufacturing Cells", *Butterworth-Heinemann Ltd. Publications*.

Desmon Ginting (1995), *"Design, Implementation & Simulation of Hybrid Architecture for Control of Scheduling activities in Flexible Manufacturing Systems"*

E. Rondeau, T. Divoux, F. Lepage, M. Veron (1993), "Creation of Virtual Manufacturing Devices by means of Information Systems Models", *Proceedings of IFIP Conference on World Class Manufacturing , 1993*, pp 259-273.

Elsayed and Thomas Boucher (1985), "Analysis and Control of Production Systems", *Prentice-Hall Publications.*

Kieth Lockyer, Alan Muhlemann and John Oakland (1988), "Production & Operations Management", *Pitman Publications*.

Martin Hardwick, David L. Spooner, Tom Rando and K. C. Morris (1996), "Sharing Manufacturing Information in Virtual Enterprises", *Communications of the ACM*, vol. 39, no. 2, pp 46-54.

Mikell P. Groover (1980), "Automation, Production Systems and Computer Aided Manufacturing", *Prentice-Hall Publications*.

Nand K. Jha (1991), "Handbook of Flexible Manufacturing Systems", *Academic Press Inc. Publications*.

Paul G. Ranky (1983), "The Design & Operation of Flexible Manufacturing Systems", *IFS Publications*, UK.

Paul T. Kidd (1994), "Agile Manufacturing - Forging New Frontiers", *Addison-Wesley Publications*.

Prof. W. B. Heginbotham (1985), "Proceedings of the 1st International Conference on Simulation in Manufacturing", *IFS Publications*, UK.

Richard C. Dorf & Andrew Kusiak (1994), "Handbook of Design, Manufacturing & Automation", *John Wiley & Sons, Inc. Publications*, New York.

William T. Wong, Jeff Y-C Pan and James D. Plummer (1992), "A Virtual Factory-Based Environment for Semi-conductor Device Development", *IEEE/SEMI International Semiconductor Manufacturing Science Symposium*, 1992, pp 117-123.

APPENDIX - A

```
-- This program has a series of procedures for buffer route logics and process logics
--for the machines in the model.

procedure buffer_route_logics()
  begin
    cwgt->drilled=false
      transfer cwgt thru output 1
  end

procedure buffer1_route_logics()

  begin
    cwgt->drilled=true
      transfer cwgt thru output 1
  end

procedure cnc_process_logics()

  var

    proc_1 : process
    proc_2 : process


  begin

    proc_1 = get_process('drilling_1')
    proc_2 = get_process('drilling_2')

  require widget any
    if(cwgt->drilled= =0) then
    cwgt->drilled=false
      do_process(proc_1)
  else
      do_process(proc_2)
  endif
  end


procedure cnc_route_logic()

  begin
```

```
    if(cwgt->drilled= =0) then
    cwgt->drilled = true
      transfer cwgt thru output 1
  else
      transfer cwgt thru output 2
  endif
  end

procedure buffer2_route_logic()

  begin
    cwgt->photo=false
      transfer cwgt thru output 1
  end

procedure buffer3_route_logic()

  begin
    cwgt->solder=false
      transfer cwgt thru output 1
  end

procedure buffer4_route_logic()

  begin
    cwgt->legend=false
      transfer cwgt thru output 1
  end

procedure photo_process_logics()

  var

    proc_1 : process
    proc_2 : process
    proc_3 : process


  begin

    proc_1 = get_process('photographic')
    proc_2 = get_process('soldermask')
    proc_3 = get_process('legend')

   require widget any
```

```
    if (cwgt->photo= =0) then
    cwgt->photo = false
      do_process(proc_1)

  else
    if (cwgt->solder= =0) then
    cwgt->solder=false
      do_process(proc_2)

  else
    if (cwgt->legend= =0) then
    cwgt->legend=false
      do_process(proc_3)

    endif
    endif
    endif
  end


procedure photo_route_logic()


  begin


    if (cwgt->photo= =0) then
      cwgt->photo=true
      transfer cwgt thru output 1

  else
    if (cwgt->solder= =0) then
      cwgt->solder=true
      transfer cwgt thru output 2

  else
    if (cwgt->legend= =0) then
    cwgt->legend=true
    transfer cwgt thru output 3

  endif
  endif
  endif
  end
```

```
-- This program has a set of procedures which when assigned to a particular machine
 --in the simulation model performs two different processes one after another on the
--same job.


procedure buffer_route_logics()
  begin
  cwgt->drilled=false
  transfer cwgt thru output 1
  end

procedure buffer1_route_logics()
  begin
  cwgt->drilled=true
  transfer cwgt thru output 1
  end

procedure cnc_process_logics()
  var
  proc_1 : process
  proc_2 : process

    begin
      proc_1 = get_process('drilling_1')
      proc_2 = get_process('drilling_2')
     require widget any
      if(cwgt->drilled= =0) then
      cwgt->drilled=false
      do_process(proc_1)
    else
      do_process(proc_2)
    endif
    end


procedure cnc_route_logic()

  begin
    if(cwgt->drilled= =0) then
    cwgt->drilled = true
      transfer cwgt thru output 1
    else
      transfer cwgt thru output 2
    endif
  end
-- This procedure connects all the resources in the model automatically by reading
--them from a file. This connection enables the flow of jobs from one resource to
--another in the model during simulation.


procedure connect()

var
  count : integer
   a1 : string
   a2 : array[10] of string[32]
   k1, k2 : string
   i, j, temp : integer



begin

        a1 = 'Source_name'
        count = 9
        k1 = '1'
        k1 = k2

--This opens a file f1 which has a sequence of all the resources in the model and are
--read one by one by this program.

        open file 'c:/questlib/sclmacro/f1' for input as 1


        for i=0 to count do
          k1 = k2
          k2 = '1'

          if (i>=1) then
          temp = 0
            for j = i-1 to 0 by -1 do

                 if (a2[i] = = a2[j] then
                 temp = val(k2)
                 temp = temp + 1
                 k2 = str('%g', temp)
                 endif
```

```
              endfor
          endif

read ( #1, a2[i] )


BCL(“connect “‘+a1+’” output “+k1+” to “‘+a2[i]+’” input “+k2+’””)

          a1 = a2[i]

          k1 = ‘1’
    endfor
end
```

```
--This procedure gives the total time spent by a Widget or job in the system.

user_attrib

start_time1 : real
start_time2 : real

procedure finish()

var

  delta_time : real

begin

  if (cwgt->class_name = = 'Widget_1') then

  delta_time = sim_time - cwgt->start_time1


  write('finish 1       ',delta_time,cr)

  transfer cwgt thru output 1
else

  delta_time = sim_time - cwgt->start_time2


  write('finish 2      ',delta_time,cr)

  transfer cwgt thru output 1
endif
end
```

```
-- This procedure routes the jobs to the next available machine in the system

procedure route_next_free_res()

var

  outchoice    :   integer

begin

   -- All  resources call function to return the output connected to
   -- the next free resource

   wait until ready (ANY)

   while (TRUE) do

     outchoice = free_res_output(cwgt)

     if (outchoice = = -1 ) then
          if (cres->num_outputs = = 1) then
            if (cres->resource_type <> SOURCE) then
               unload sample_unload_time()
        endif

        transfer cwgt to cres->out[1]
            return
      endif

     else

          if (cres->resource_type <> SOURCE) then
            unload sample_unload_time()
      endif

      -- transfer to the selected output

      transfer cwgt thru output outchoice
         return
     endif
```

```
      delay 1

    endwhile
end
--This procedure routes  the jobs to the least utilised resources in the system


procedure route_least_utilised()

var

   outchoice    :   integer

begin
    -- call function to return the output connected to the buffer that
    -- has the lowest utilisation to this point in the run.


      outchoice = least_util_output(cwgt)

    if (outchoice = = -1) then
      route_next_free_res()
      return
    endif

    if (cres->resource_type <> SOURCE) then
      unload sample_unload_time()
    endif

     -- transfer to the selected output

    transfer cwgt to cres->out[outchoice]

end
```

```
--This procedure routes the jobs to the machines which have the least number of jobs
--in the queues  on buffers in front of them.

procedure route_min_queue()

var

  outchoice    :   integer

begin

    -- call function to return the output connected to the buffer that
    -- has the fewest number of widgets


     outchoice = min_q_output(cwgt)


    if (outchoice = = -1) then
      route_next_free_res()
      return
    endif

    if (cres->resource_type <> SOURCE) then
      unload sample_unload_time()
    endif

    -- transfer to the selected output

    transfer cwgt to cres->out[outchoice]

end
```

```
--This procedure routes the jobs to the machines where the jobs have to wait for the
--minimum time on the buffers in front of them.

procedure route_min_wait()

var

  outchoice    :   integer

begin

   -- call function to return the output connected to the buffer that
   -- has the minimum waiting time to this point in the run.

        outchoice = min_wait_output(cwgt)


   if (outchoice = = -1) then
     route_next_free_res()
     return
   endif

   if( cres->resource_type <> SOURCE) then
     unload sample_unload_time()
   endif

   -- transfer to the selected output

   transfer cwgt to cres->out[outchoice]

end
```

```
-- This procedure gives the flowtime or the total time spent by each of the jobs in the
--system during simulation. The times are appended into separate files.

procedure flowtime()

var

  sim_time : real

begin

  require widget any

    if(cwgt->class_name = = 'j1') then

    open file '/questlib/reports/tab/job1' for append as 1
    write (#1, 'flowtime1    ' ,sim_time,cr)

  else

    if (cwgt->class_name = = 'j2') then

    open file '/questlib/reports/tab/job2' for append as 2
    write (#2, 'flowtime2     ' ,sim_time,cr)

  else

    if(cwgt->class_name = = 'j3') then

    open file '/questlib/reports/tab/job3' for append as 3
    write (#3,  'flowtime3     ',sim_time,cr)


  else

    if (cwgt->class_name = = 'j4') then

    open file '/questlib/reports/tab/job4' for append as 4
    write (#4, 'flowtime4     ',sim_time,cr)

  else
```

```
    if(cwgt->class_name = = 'j5') then

    open file '/questlib/reports/tab/job5' for append as 5
    write (#5, 'flowtime5      ',sim_time.cr)

  else

    open file '/questlib/reports/tab/job6' for append as 6
    write (#6, 'flowtime6      ',sim_time.cr)


endif
endif
endif
endif
endif

  pass()


close #1
close #2
close #3
close #4
close #5
close #6

end
```

```
--This procedure checks for the job type and selects the process to be done.

procedure proc()

var

  proc_1 : process
  proc_2 : process

begin


  proc_1 = get_process('process_1')
  proc_2 = get_process('process_2')

    require widget any

      if(cwgt->class_name = = 'Widget_1') then

    do_process(proc_1)

else

    do_process(proc_2)

endif
end
```

```
--This procedure  gives  the time spent by each of the jobs or widgets for separate
--processes

procedure machinetime()

var

proc_1 : process
proc_2 : process

begin

proc_1 = get_process('process_1')
proc_2 = get_process('process_2')

  require widget any

  if (cwgt->class_name == 'Widget_1') then
  do_process(proc_1)

  write ('w1     ',sim_time,cr)

else

  do_process(proc_2)
  write ('w2     ',sim_time,cr)

endif

end
```

```
-- This procedure arranges the different classes of jobs into groups or queue_orders
--and then sorts them in the ascending order. Then the jobs are processed as per the
--order number.


user_attrib

  queue_order : real


procedure initiate()

begin

 require widget any

  if(cwgt->class_name = = 'Widget_1') then
  cwgt->queue_order = 1

else

  if(cwgt->class_name = = 'Widget_2') then
  cwgt->queue_order = 2

else

  if(cwgt->class_name = = 'Widget_3') then
  cwgt->queue_order = 3

endif
endif
endif

--calls the function to sorts the orders in ascending order.

  q_ascending()

end
```

```
--This procedure simulates the setup times between processes and enables the user to
--define setup time as shown in the statement 'setup 200' where 200 is the setup time
 --in seconds.


user_attrib
tab : real

var

  curproc : process
  lastproc : process


procedure s()

begin


  curproc = get_process('process_1')
  lastproc = get_process('process_2')

  require widget ANY

  if(cwgt->tab = = 0) then

  cwgt->tab = false


    do_process(curproc)

   else

    do_process(lastproc)

   endif


  if(lastproc <> NULL and lastproc <> curproc) then
  setup 200
  endif
  do_process(curproc)
  lastproc = curproc
end
-- This procedure routes the jobs in order of the priorities assigned to them, i.e., jobs
--with the highest priority are routed first

procedure route_priority()

var

  outchoice    :   integer

begin

   -- call function to return the output connected to the resource
   -- has the highest priority


     outchoice = priority_output(cwgt)


   if (outchoice = = -1) then


     route_next_free_res()
     return
   endif

   if ( cres->resource_type <> SOURCE) then
     unload sample_unload_time()
   endif

   -- transfer to the selected output

   transfer cwgt to cres->out[outchoice]

end
```

```
-- This is the procedure for simulating a machine breakdown scenario in the model.

USER_ATTRIB
fail_count : integer


VAR
proc_id : Process

procedure abproc()

begin
        require widget any
          proc_id = get_process('process_1')
          work sample_cycle_time(proc_id)

-- The statement 'fail 0,60' affects the machine failure and the failure time is 60
--seconds starting at 0 seconds from the beginning  of the simulation. This time can be --
altered as per the users requirements.

                fail_count = fail_count + 1
                if(cres -> fail_count == 10) then
                fail 0,60
                cres -> fail_count=0
                endif

            pass()

end
```

APPENDIX - B

The following are the results of the simulation experiments :

	Cell 1	Cell 2	Cell 3	Average
RANDOM	85.3	81.4	76.6	81.1
LUM	79.1	72.3	71.1	74.167
MINQ	84.4	78	68.5	76.967
MINW	76	79.5	84.4	79.967

Average Machine Utilisation of all the machines in each of the Cells.

	RANDOM	LUM	MINQ	MINW
Job 1001	503.34	741.78	690.34	719.03
Job 1002	472.55	465.34	602.12	544.82
Job 1003	640.15	533.5	660.15	451.65
Job 1004	615.05	587.5	514.45	394.75
Job 1005	540.02	342.12	720.45	527.33
Job 1006	584.35	550.15	527.77	347.04
Mean Weighted Flow Time	559.24	536.73	619.21	497.44

Mean Flow Time of all Job Types for the Machine Selection Rules.

	Cell 1	Cell 2	Cell 3	Average
RANDOM	1.53	1.43	1.59	1.52
LUM	1.54	1.8	1.89	1.74
MINQ	1.49	1.34	1.28	1.37
MINW	0.98	1.04	0.86	0.96

Average Queue Lengths

RANDOM	212
LUM	192
MINQ	212
MINW	199

System Production Rate

	JOB 1001	JOB 1002	JOB 1003	JOB 1004	JOB 1005	JOB 1006
FIFO	9	20	37	29	19	22
PRIORITY	13	32	17	17	32	22

Job Type Production Rates for Part Selection Rules

	JOB 1001	JOB 1002	JOB 1003	JOB 1004	JOB 1005	JOB 1006
FIFO	695.91	600.37	515.22	671.44	702.5	664.66
PRIORITY	612.93	508.33	597.95	789.09	596.07	817.33

Mean Flow Times for Part Selection Rules

	Without Breakdown	With Breakdown
D1	63.94	82.4
D2	64.28	39.92
M1	77.98	73.85
M2	76.75	72.22
MC1	79.46	64.85
MC2	71.77	65.54
MC3	78.73	64.49

Machine Utilisation Analysis

	Without Breakdown	With Breakdown
1001	983.36	993.23
1002	1257.27	1257.27
1003	876.41	879.05
1004	935.62	956.34
1005	1172.33	1184.03
1006	408.8	408.8

Mean Flow Time Analysis