A KNOWLEDGE BASED MODELLING SYSTEM FOR THE DESIGN AND EVALUATION OF FLEXIBLE MANUFACTURING FACILITIES

Volume I

by

WEI WANG

A Doctoral Thesis
submitted in partial fulfilment of the requirements
for the award of

Doctor of Philosophy

of Loughborough University of Technology

Department of Manufacturing Engineering

March 1989

DECLARATION

No part of the work described in this thesis has been submitted in support of an application for any other degree or qualification of this or any other University, or the C.N.A.A. or other institute of learning.

To My Parents for Their Love and Support

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my supervisor Professor R. Bell for his direction, support and interest in this research; Dr. K. Popplewell for his support of the Xerox Workstation and help throughout this work. My gratitude to Mr. S. Newman and Dr. R. de Souza for their helpful discussions and provision of the results for the comparative case study. I am grateful to my colleagues, Mr P. Zhang, Miss L. McQueen and Miss J. Adams of the FMS Research Group, the Alvey Program and the ISS Research Group for their valuable discussions; Mrs. J. Booth and Mrs. D. Donnellon for their administrative support.

I am indebted to Cummins Engine Company Ltd., Daventry, for providing the data for the case study; Mr. D. Gough for his work on the collection and collation of the data.

I thank Mrs. H. Jiao for her assistance and helpful discussions; and her research into the knowledge based factory modelling on the Xerox Workstation.

My thanks are also made to all my friends for the friendship, inspiration and help throughout the period of my work; in particular, Dr. R. Luo, Mr. G. Varvakis, Mr. Y. Huang, Mr. A. Rui, Mr. J. Zhao and Miss F. Zheng for their tremendous help in the preparation of this thesis.

Finally, I specially thank the government of the People's Republic of China and the British Council for providing the financial support during this project.

ABSTRACT

The objective of this thesis is to explore the application of artificial intelligence (AI) techniques for modelling in flexible manufacturing. The work consists of three main parts. In the first part, the structure and performance of various types of flexibly automated batch manufacturing systems are discussed, the modelling challenge for the design of these types of manufacturing systems is identified, and the currently available modelling techniques are examined and comparatively assessed.

In the second part, the research into the structure and design of a knowledge based modelling system is reported. Potential advantages of AI techniques for manufacturing systems modelling are identified. The modelling system is then developed using the LOOPS knowledge engineering language on the Xerox 1186 AI Workstation. Major features of the modelling system include its knowledge driven requirement to enable evaluation of alternative systems with different criteria, the capability of modelling over multiple levels of detail, the transparency of its solution procedure, and the modularity of the system structure to allow convenient modification and extension.

The third part is concerned with the evaluation of the AI based modelling method. Parallel experiments are conducted on an extended case study cell by using the knowledge based modelling system, the emulator and the tool flow modelling system. Merits of the AI based method are then critically assessed, drawn on the comparison of the results obtained from the three studies. Conclusions drawn from this research and directions for future work are finally indicated.

Table of Contents

Chapter 1
INTRODUCTION

Automated manufacturing is becoming an increasingly significant feature for most of the industries. Flexible manufacturing concepts have been introduced to increase the productivity of the batch production sector of industry.

Although there are tremendous benifits, such as higher machine utilisation, lower unit costs, shorter lead times, higher quality and quick response to market changes, to be gained from the introduction of flexible manufacturing concepts, these potential advantages have to be set against the high capital cost and acquisition risk. In addition, the complexity of the system and the novelty of flexible manufacturing technology involve some measure of adventure into the unknown.

As a result, if the potential benifits of a system are to be adequately realised, a systematic and structured analysis and evaluation of the system design is required. Hence there is a great demand for models that can be used to predict, evaluate and assess the performance of the system, but it is impossible to develope models and to attain valuable results without the aid of methods. Recently there is evidance that these methods have been supported by software tools.

Drawn on an assessment of a range of currently available modelling methods, it is identified that there is scope for improvement of these techniques. Therefore this thesis explores the application of artificial intelligence (A.I.) techniques for modelling in flexible manufacturing, with a knowlege-based modelling system being successfully developed. Major features of this knowledge-based modelling system include its knowledge driven requirement to enable evaluation of alternative systems with different criteria, the capability of modelling over multiple levels of detail, the transparency of its solution procedure, and the modularity of the system structure to allow convenient modification and extension.

This work was made possible by the departmental aquisition of the Xerox 1186 Workstation shortly after the start of the research. The knowledge based modelling system described in this thesis is the author's work unless explicit references to the Xerox Workstation Manual [187] and the LOOPS Manual [27].

Following this introduction, a literature search is conducted to provide a representative coverage of relevant current knowledge in Chapter 2. Chapter 3 describes the structure and performance of system variants for batch manufacturing. This is followed by a discussion of the modelling challenge in the field in Chapter 4. In Chapter 5, major currently available modelling methods are discussed and comparatively assessed.

In Chapter 6, the scope for this research is identified and highlighted. The research into the design and structure of the knowledge-based modelling system is reported in Chapter 7. Chapter 8 describes the modelling knowledge embedded within the system. The user interface and the system outputs are presented in Chapter 9 and Chapter 10 respectively. In Chapter 11, the workstation constraints associated with the work are indicated.

Chapter 12 introduces the three machine cell which is used for the comparison of the three levels of modelling, the data being provided by a British company. Then the behavioural rules corresponding to the basic operation of the three machine cell are described in Chapter 13. After this, the decision rule and data input of the three machine cell model is presented in Chapter 14. In Chapter 15, results of the knowledge based modelling of the three machine cell are discussed and compared against the three levels.

The extended cell is then introduced in Chapter 16 to allow a more critical assessment of the knowledge based modelling system based on the three machine cell, with the input to the modelling system being illustrated. Chapter 17 summarizes and discusses the results obtained from the knowledge based modelling of the cell. In Chapter 18, merits of the AI based method are then critically assessed drawn on the comparison of the results with those obtained from the Emulator based study and the tool flow modelling of the extended cell.

In Chapter 19, the major issues arising from the research work, the experience obtained from the case studies and the the potential value of the modelling system are highlighted. Finally, specific conclusions drawn from this research and recommendations for further work are given in Chapters 20 and 21 respectively.

Chapter 2
LITERATURE SURVEY

2.1 Introduction

This literature review is conducted with the emphasis placed on locating valuable contributions in each relevant topic to provide a representative coverage of its state of the art. Major problems are identified and the approaches adopted are examined.

An overview is offered of the overwelming flexible manufacturing technology. Design methodologies and the process required for a complete study of flexibly automated batch manufacturing systems are discussed. Some light is shed on the control systems that are utilised for the efficient operation of the system. Finally major effort is spent on a cross section of modelling techniques inclusive of AI and expert systems, which can be applied during all stages of system development.

2.2 Flexible Manufacturing

The primary purpose of reviewing the concept and functional aspects of flexible manufacturing in the context of this research, is to identify the significant system characteristics and objectives that should be taken into account.

So far as low-volume and high-volume production are concerned, the automation of the production process has attained a high level. Between these two fields of production, flexible manufacturing has been offering the potential for achieving a high level of automation as well as productivity in the field of batch manufacturing [285] (Figure 2.1). The effect of major economic and technological change is becoming dramatically evident in the field. In addition to the improvement in automation and productivity, flexible manufacturing technology is capable of producing substantial benifits, such as reduced work in progress, high equipment utilisation, short lead time, high quality and improved flexibility [24]. However, these advantages can only be realised through careful system design and evaluation [98].

A comparison [179] [169] between flexible manufacturing and other production methods can be made in a number of dimensions (Figure 2.2, Figure 2.3). Other relevant influential factors for comparing the feasibility of flexible manufacturing technology with that of conventional manufacturing systems are listed in Figure 2.4 [216].

The first ever attempt in flexible batch manufacturing was introduced by Williamson [297] some twenty years ago. The System 24 provided a tremendous challenge to conventional thinking in system configuration, machine tool design, workpiece and tool flow and in computer control. In the intervening years development in computer technology has made available both a hierarchy of computers suitable for use in real time control and a whole range of mechanical elements coupled with relatively flexible software controls [40]. CNC machine tools are now supported by readily available automated conveyors, pallet transfer devices and industrial robots [125]. This range of hardware plus the computer technology needed for communication between individual equipments constitute the core of advanced flexible manufacturing technology.

Development of modern flexible manufacturing has made it possible to design manufacturing systems using available building blocks (Figure 2.5). Specification of the system is chosen to match manufacturing need which is expressed in terms of part, quantity and variety [198] [15] [190]. Bell [40] has built a typical graphical representation of the interplay between volume, variety and manufacturing system configuration (Figure 2.6). Three major categories of systems are possible for batch manufacturing: the unmanned station(UMS), the flexible manufacturing cell(FMC), and the flexible manufacturing system(FMS). In addition, multi-cell flexible manufacturing (Multi-Cell FM) and large-scale DNC systems have come into being [156] [270] as a result of recent development in flexible manufacturing technology.

The UMS is a stand-alone CNC machine tool equipped with a robot or muti-pallet changer, and machining monitoring system to allow it to operate without direct manning for the majority of its productive time [23] [275] [197] [14] [103] [125].

An FMC is in general terms a group of processing modules supported with the automation of workpiece flow to bring a family of parts to desired level of completion without leaving the cell [33] [17] [78] [125]. It is considered as a system concept for production with limited routing ranges of batches and with a small number of operations [158] [135] [159] [20] [120] [12].

The FMS is a highly automated variant with greater capability to respond to short term changes in manufacturing requirement. In addition to the automated work flow, it possesses advanced auxiliary functions for machining operations, monitoring function and a comprehensive computer control function; and requires NC data control, and production scheduling and control [25] [126]. An FMS installation offers varying degrees of flexibility in workpiece variety, batch size and batch distribution [149] [155a] [298] [66] [16] [127] [157] [96] [214] [213] [210] [147] [57].

A Multi-Cell system is a large automated processing network, in which different types of cells are linked by material handling devices, all under computer control. Typically, it consists of cells for fabrication, machining and assembly [103] [206]. Therefore the system is capable of automatically processing workpieces of varying shapes, sizes and materials [156] [290] [129] [17] [82] [95].

The large-scale DNC system is the latest development in the field of batch manufacturing. It is a highly integrated manufacturing configuration comprising various work stations all under central computer control. The work stations can be any forms of manufacturing units, e.g. individual machine tools, DNC cells, automated manufacturing cells, or FMSs. This type of system provides a solution for the automation of a total manufacturing sector [13] [163] [21] [22] [28] [30] [19] [11] [29].

To date, hundreds of these systems have been implemented in the world (Figure 2.7) [47]. Most of the flexible manufacturing aspects are covered by the annual International Conference on Flexible Manufacturing Systems, the FMS Magazine and papers presented at almost every conference in manufacturing, automation and similar fields. Ranky [217] published the first book on flexible manufacturing, which provides an insight into how to set up and run a flexible manufacturing system successfully. The broad sweep of current flexible manufacturing technology has been comprehensively reviewed by Hartly [125]. The striking feature of this contribution is the extensive illustration of applications which are currently in operation.

2.3 Design of Flexible Manufacturing Facilities

One of the early contributions to the design of flexible manufacturing systems is made by the Charles Stark Draper Laboratory Inc. [71]. The handbook aims to answer the following questions:

- Why an FMS?
- Will an FMS best serve your application?
- What problems might be encountered?
- How do you design an appropriate system? and
- What is required to operate a system?

Seven steps for implementing an FMS are recoganised: selection of parts and machines, design of alternative system configurations, evaluation of the alternatives, writing of a Request-for-Proposal, evaluation of the vendor proposals, installation and debugging of the system and the eventual operation of the system (Figure 2.8).

Barash, et al. [35] reports on the planning of computerised manufacturing systems. The overall process is divided into six major steps. First of all, parts belonging to the same family are selected on the basis of production needs. Next the machining content of each part and the typical batch size are determined. After that system functional elements, including the numbers and types of machine tools, are determined. Then the various feasible system configurations inclusive of material handling devices are composed. After a simulation of the operation of the variants is conducted to test their general performance, the best system can then be identified and the operating rules for the system be justified.

Warnecke and Scharf [285] in this early paper, discuss the significant criteria that should be considered in the development of integrated manufacturing systems. Industrial studies emphasising the design aspects of FMS can be found in the theses of Bilalis [49], Parris[204] and Newman [192a].

Warnecke and Vettin [286] have formulated a classification scheme to categorise FMC configuration with respect to the modes of material handling devices deployed and the operational inclusion of the flexible buffers to cope with variations of work in progress. Afentakis [5] and Afentakis, et al. [6] have described a modelling framework for the determination of the optimal physical layout for FMS.

Chan and Rathmill [69] present an integrated systematic approach to the design of FMS. The whole process involves three stages: planning, design and implementation (Figure 2.9). Within the planning phase, the components which are to be manufactured and their process plans and batch sizes have to be determined (Figure 2.10). The design phase involves detailed design of the FMS to meet the requirements identified in the planning phase. This is accomplished by firstly selecting the type of FMS most likely to be suitable, and secondly determining the detailed design issues relating to the number of machine tools, the type of material handling systems, the location and size of WIP buffers, the type of computer control and the scheduling system (Figure 2.11).

In the third phase, the system software and the actual production system are designed and implemented (Figure 2.12). The system sofeware includes the DNC software, the scheduling software, and the software to provide WIP control, process monitoring and in process inspection. The implementation of the actual production system consists of the specification and installation of machine tools, robots and automatic pallet changers, the design and construction of special purpose fixtures and pallets, design and implementation of the material handling system, the inspection system, the tooling strategy and the maintenance strategy.

Sarin and Chen [233] have formulated a systematic procedure for designing and selecting the best mix of manufacturing systems subject to the production demands at minimum overall manufacturing costs under various operating conditions. Four possible system configurations are considered: job shop, NC job shop, FMS and transfer line. The selection decision is made on the basis of an integer programming model. Results have shown that a mixed system configuration can be better than one system alone in meeting the desired production requirements.

Kumar and Vannelli [160] discuss the issue of redesigning the traditional production system into a disaggregated cellular production system using group technology techniques. The disaggregation process is accomplished by evaluating critical strategical decisions regarding subcontracting of parts as well as balancing of capacity between the various cells.

Azadivar and Lee [34] propose a procedure for determining the optimum number of buffer spaces for each work station so that for a desired level of machine utilisation the overall WIP is minimised.

Stecke [255] defines the FMS design problems which involve the making of decisions in two stages: the initial specification and the subsequent implementation. Figure 2.13 shows the decisions that are made in these two stages respectively.

Another integrated design system is illustrated in Figure 2.14, which consists of an analysis stage and a design stage [90].

2.4 Control of Flexible Manufacturing Facilities

The operational control aspects of flexible manufacturing have been addressed by a number of researchers. Warnecke and Scharf [285] and Eversheim and Westkamper [104] were among the earliest to recognise the need for a hierarchical structure for controlling flexible manufacturing facilities, based on their research in West Germany. Succeedingly, Buzacott and Shanthikumar [59] in their paper on the analysis of FMS, which has been cited by many others, indicate that the complexity of the overall control problem necessitates the system analysis at the following three levels (Figure 2.15):

- The pre-release planning is concerned with the selection of suitable parts for production over a medium term time interval compatible with overall system resources.

- The release or input control is to determine the sequence and timing of the release of jobs to the system.

- The operational control deals with the movement of parts between machines, route management and resource disruptions.

This hierarchical approach to the overall control problem is also supported by other researchers, notably Bell and Bilalis [41], Canuto, et al. [62], Kimenia and Gershwin [152], Solberg [250], Stecke [254], Suri and Whitney [269].

The level one pre-release planning problem is significantly addressed by Menon and O'Grady [183] [182]. A linear programming model is put forward which takes tools required, machine capacity, tool availability and due dates into account and selects jobs which should be loaded. The model's objective function is an summation of the weighted deviations from the desired level of these parameters. By varying the weighting factors different solutions can be obtained. Figure 2.16 shows the operational considerations contained within the model.

The sequence of the release of orders to the system in the release or input control level is best performed by the use of priority rules [143] [196] [49] [41]. These rules can be categorised into either static or dynamic ones. The former makes the decision by examining the state of the system at the decision time without considering the impact of the decision on the system, while the latter considers not only the current system state but also the state resulted from the decision being made.

Rules can also be developed on the basis of either the orders to be released into the system or the system status [262]. Rules based on orders are mainly concerned with the properties of the orders, such as due dates or processing times [51], whereas rules based on the system status examine mostly the workload of each station. The workload includes the total machining time of the parts which are currently being machined or are to be machined at the work station, and parts which are waiting in the temporary stores to visit this work station [49].

The timing of the release of new orders to the system at the release or input control level depends in great measure upon the mode of operation (manned or unmanned) and the corresponding preparation strategies. Three strategies are proposed in [49]:

- Release a new batch when the previous batch has been introduced into the system and a new palletisation function starts.

- Orders are released at the beginning of each shift until the capacity of each work station has been fulfilled.

- Release orders each time the workload of a machine tool drops below a certain limit.

Consideration of a range of priority rules for loading and scheduling at the operational control level of flexible manufacturing has been undertaken by a number of researchers. Prominent among these was the work by Stecke and Solberg [257]. Based on an industrial situation, an experimental investigation of operating strategies for an FMS was established, and policies for loading and real time flow control were determined and tested. The results showed the high level of dependence of system performance on the policies chosen for the loading and control. Altogether five loading rules and sixteen scheduling rules were tested, and from the results loading and control methods were identified that dramatically improved the production rate of the system.

In the study by Bell and Bilalis [41] concentrating on the choice of control strategies for rotational part FMS, it was concluded that different measures of performance will require the development of different rules and completely different results must be anticipated.

Hutchinson [138] has considered the issues and problems associated with the control and efficient operation of FMS. The control mechanisms are viewed in the contex of the general problem of allocating scarce resources so that multiple conflicting objectives can be achieved simultaneously. The overall system control includes both automatic and adaptive control mechanisms and is organised in a five-level hierarchy ranging from the second by second decisions up to the whole structural level (Figure 2.17).

Whitney [292] reports on the control concepts and principles in flexible manufacturing. Extractions are made from the FMS application of whatever requirements and proposed solutions may be of general interest.

Sackett [232] proposed control policies for realising high performance in a mixed flexible and conventional manufacturing system. Conclusions were drawn on the implications of the control strategies for the effective total operation in such an FMS inclusive environment (Figure 2.18).

Edghill and Cresswell [97] consider the current research dealing with the control of FMS, in particular the production scheduling and tool management. It is concluded that generalised control strategies for FMS are not currently feasible with regard to both the diversity of system design and the sensitivity of system performance to the optimising criteria used.

2.5 Modelling

So far as the analysis of conventional manufacturing systems is concerned, the methodologies for a single CNC machine tool and automatic transfer lines have been considered to be fairly clear and their applications have been well-established. However, the problems posed in the design of flexible manufacturing facilities are far more difficult to be solved and traditional design techniques for conventional manufacturing systems have been proven largely ineffective when they are applied to flexible manufacturing [166]. Therefore there is great demand for models which can assist the design process of flexible manufacturing.

There are many different kinds of decisions to be made in the design and control of flexible manufacturing. Hence there exist many different ways to model a manufacturing system, depending upon the emphasis given to the different aspects. Classification of the models, as a result, can be conducted along several dimensions [265a]. Solberg [250] has classified models according to the form, the system objective, the time nature and the variability. Wilhelm and Sarin [295] and Looveren et al. [171] have provided classifications on the basis of various decisions to be made in the models.

Another classification system for models is presented by Doumeingts et al. [91] based on the level of abstraction, the nature of the model and the various steps of the life cycle. For the purpose of this thesis, modelling is broadly classified into approximate modelling, simulation based modelling and knowledge based modelling in terms of the complexity of logical details and the level of intelligence contained within the model.

2.5.1 Approximate Modelling

Approximate modelling provides a quick estimate of how a manufacturing system behaves and how its components interact, or provides decisions arising in the design and operation of a manufacturing system. The basic requirement of approximate modelling is that firstly the system performance output should be realistic and be effective, secondly, the modelling process should be efficient. The use of such a model helps to determine appropriate procedures to set up a system or strategies to help run a system efficiently.

Techniques which can be considered as approximate methods include static capacity analysis, queueing networks, mathematical programming, hybrid queueing networks/mathematical programming, heuristic algorithms, semi-Markov process and Petri nets.

2.5.1.1 Static Capacity Analysis

To set up targets for the capacity and performance of manufacturing systems being designed, a static capacity analysis can be conducted [217] [31].

Lenz [165] reports on a computer program called SPAR which is part of an advanced manufacturing system design tool. The program is used for aggregate capacity planning of manufacturing systems. Typical inputs to the program include component identification, production requirements, cycle times, transporte times and pallet load/unload times. By running the model, the production capacity of the system is given with the number of stations, transporters and pallets needed for feasible production levels.

2.5.1.2 Queueing Networks

The preliminary theory of queueing networks was established by Jackson [144] where he identified the criteria for the construction of a network of queues. It is assumed that a network of queues consists of a certain number of stations, each of which has one or more identical servers, and a customer may leave one station and proceeds to another. This work was subsequently extended in Jackson [145] to a broader class of networks. Further extention of this theory was presented by Gorden and Well [118] to pertain to the determining the steady state distribution of customers in a general class of closed queueing networks in a product form.

The most general model was developed by Baskett et al. [39] where solutions can be provided for closed, open and mixed networks of queues with different classes of customers. Graham [119] showed that open models are usually easier to solve than closed ones, but closed queueing networks are ofen better representation of real systems. Based on the two-dimentional iterative techniques, Buzen [60] presented efficient computational algorithms for solving these queueing models.

Throughput, one of the most important performance measures of queueing networks, was studied by Schweitzer [235]. It was concluded that the maximum possible throughput in a finite- capacity system is equal to the arrival rate which just saturates the slowest station in a corresponding infinite-capacity system, and a system with a infinite capacity can have a strictly greater throughput than that with a finite capacity.

As a result of the complexity of the real systems, queueing models can, in some cases, be inordinately expensive to attain exact solutions. Thus approximate methods are required which can retain the qualitative behaviour of the actural system while permitting adequately good estimates of the quantities of interest, such as the average queue length. The principal techniques are decomposition and diffusion [70] which are of a heuristic nature and computationally feasible.

Because of the equivelance between queueing networks and flexible manufacturing systems where machine tools and transporters can be considered as stations of a queueing network, and parts as customers flowing in the network, a cross section of models have been developed based on this theory.

Solberg [248] [249] [251] developed the first model for FMS design drawn on the theory of closed queueing networks. The CAN-Q (Computer Analysis of Queues of Networks) model allows the user to predict, with great ease and efficiency, a number of system performance figures, such as production rate, machine utilisation, queue length distribution, flow time and output sensitivity.

Although features like finite storage space, workpiece blocking and time or state dependent routing and scheduling could not be modelled by this early version of queueing model, recent research by Vinod and Solberg [282] [283] has greatly enhanced the capability of the model. Issues, such as the optimum system configuration subject to the operation cost and the maximum productivity have been addressed. Using the proposed partial implicit enumeration algorithm the optimum number of machines in each group and the minimum WIP can be determined. Other extentions have been presented by Diehl and Suri [87] [88] which take into account both tool sharing and workpiece blocking due to the finite local storage.

Models based on the open queueing networks have been developed by Buzacott and Shanthikumar [59]. They are used to study in particular the part selection and release problem and the effect of various buffer storages. It is concluded that if the release of jobs to the system can be controlled, the more the diversity of job routing, the higher the production rate that can be achieved.

Buzacott [58] studied the control strategies using open queueing network models. It was shown that the production capacity of a system depends in great measure on the release rules which are determined based on the feedback from the system at each control level.

Based on Little's formula [170], an alternative method called Mean Value Analysis for analysing queueing networks was introduced by Reiser et al. [225] and Bard [36]. Without computing the product terms and normalization constants, which is required by the CAN-Q model, performance measures like mean throughput, mean utilisation and mean queue length can efficiently be obtained using this approach.

Suri and Hildebrant [268] developed the MVAQ model for FMS design based on the methods of Hildebrant [130] and Cavaille and Dubois [67]. The MVAQ model is proven to be an efficient tool for determining the optimum number of machines in each machine group, the minimum number of pallets/fixtures, the best routings for multi part types and many other issues. Moreover the algorithm used in the model has a physically meaningful interpretation that can be considered as a basis for further heuristic extensions [292]. Recently Shalev-Oren [239] has extended MVAQ to PMVA which models various non-preemptive priority service disciplines as well as multiple product types and parallel machine stations.

Notably, Solberg's CAN-Q model, Buzacott's opening queueing network model and the MVAQ model of Suri and Hildebrant are all based on unrealistic assumptions, such as exponential processing times and probabilistic part routing [91]. To alleviate this problem, the Operational Analysis approach of Denning and Buzen [86] can be adopted. Based solely on operationally testable assumptions, this approach does not begin with stochastic hypotheses which are impossible to validate. This makes it possible to model features, such as bottlenecks and load dependent behaviours, which have to be ignored in models like CAN-Q and MVAQ. Use of such an approach for FMS modelling is reported in [172a].

2.5.1.3 Mathematical Programming

As a well-established quantitative method, mathematical programming is of particular significance for the optimum decision-making involved in the design and operation of FMS. The primary techniques available include linear programming, non-linear programming and dynamic programming [171].

Afentakis [5] has employed linear programming method to determine the optimum physical layout of flexible manufacutirng systems. A similar approach is used by Azadivar and Lee [34] for the determination of optimum number of buffers for each machine station of the system.

Stecke [254] formulated the machine grouping and loading problems as non-linear mixed integer programs subject to the constraints of part operation assignment and tool magazine capacity. The objective of machine grouping was expressed as maximising the pooling of all machines of the same type into one group, while several objectives were recoganised for the loading problem. To solve the computational problem, heuristic algorithms, rather than a direct attempt to achieve the objectives, were utilised for each problem.

To minimise the machine utilisation losses due to the batching of parts, Luca [172] solved the batching problem as a linear program using the simplex method.

2.5.1.4 Hybrid Queueing Networks/Mathematical Progarmming

There is evidence that queueing networks and mathematical programming can be used together as an enhanced method [138] [139], in which aggregate work flow, rather than the movement of individual parts, is modelled.

Based on The CAN-Q model, Kimemia and Gershwin [151], [153] employed a non-linear programming approach for non-deterministic systems (i.e.,the arrival and processing times are stochastically distributed) and a linear programming approach for deterministic systems to optimise the production rate and WIP inventory of the system. Characterised by the investigation of the aggregate work flow rather than the movement of individual parts, this model solved the problem of choosing an optimum mix of operating strategies for an FMS. A similar approach to Kimemia and Gershwin was adopted by Suardo [264] to optimise the resource assignment problem.

The approach of Micheletti [184] also relied on the closed queueing network theory. The model was formulated as a non-linear program to maximise the production rate of the system by distributing the operations of parts among available machines in a way consistent with the capacity of each machine.

2.5.1.5 Heuristic Algorithms

The complexity of the FMS modelling problem encorages the consideration of heuristic methodologies which facilitate an efficient determination of solutions which are feasible and acceptable [296] [256].

Stecke studied the loading problem for FMS using heuristic algorithms [146]. Five situations were considered and efficient algorithms were proposed. A similar approach was employed by Shanker and Tzen [241] to investigate the loading problem in the context of FMS scheduling.

Iwata et al. [142] considered the application of heuristic algorithms for the scheduling of FMS which consists of machine tools, buffer storages, and material and cutting tool transportation systems. The algorithm consists of three main steps: selection of machine tools, selection of cutting tools and selection of transport devices. It was concluded that the proposed heuristic procedure, using decision rules, may be used as a powerful tool to control the operation of an FMS. Nakamura and Shingu [192] also consider the scheduling of FMS using a two- stage algorithm based on a heuristic approach.

Kusiak and Cyrus [162] solved the routing and scheduling problems for automatic guided vehicles. Conclusions were drawn that the algorithms developed could help in determining the number of vehicles required for the system.

De Souza [84] [42] and Zhang [304] [305] [306] both use a heuristic approach to investigate the modelling of tool flows in flexible manufacturing facilities for prismatic and cylindrical parts respectively. As shown in Figure 2.19, the total tool flow in a factory has been represented as a hierarchy of levels of tool flow automation, with each level having its own focal point of tool supply. For the defined machine, cell and factory levels, the primary tool store, the secondary tool store and the central tool store are the corresponding focal points.

The input structure for the prismatic parts model is closely related to the specification of a tool flow network for a specified level of automation, and is based on the use of interactive data insertion. The network considered is a tool transport network, interlinking a hierarchy of tool stores, coupled with tool exchanges, automated or otherwise, at the machining stations, so as to allow the movement of tools around the flexible machining installation, including the central tool store and refurbishment facility. The cylindrical parts model covers the same area but the distinction is drawn with the modelling of live tools and the more complex automation at the machine level (Figure 2.20). These two projects have been carried out in collaborative interaction with this research work and are the subjects of complementary theses.

2.5.1.6 Semi-Markov Process

A semi-Markov process is a random process, where the successive state occupancies are governed by the transition probabilities of a Markov process but the stay in any state is described by a random variable which depends on both the state presently occupied and the next state to which transition will be made [71].

The first application of this approach in modelling FMS was presented by Seidmann and Schweitzer [237] to study the part selection policies.

Seidmann and Nof [236] developed a capacity model that incorporated the influence of stochastic feedback flow on the productivity of a single-part FMC. This work was soon extended to describing the productivity capacity of special multi-part manufacturing cells with stochastic activity times as well as random feedback flow. With this model, performance measures, such as total batch processing times, number of parts recycling, and cell productivity, can readily be obtained.

Alam et al. [7] presented a semi-Markov model for the performance evaluation of FMS with both exact and approximate solution procedures considered.

2.5.1.7 Petri Nets

Petri nets are useful to model systems whose behaviour can be described as interferences between asynchronous and concurrent processes [207]. While in the past Petri nets were mainly employed to answer qualitative questions, recent advances in timed Petri nets have enabled quantitative evaluation of system performance [76].

The preliminary investigation of Petri nets (including timed Petri nets) to describe, model and analyze production processes is reported by Dubois and Stecke [93]. General modelling conventions, based on the Petri nets modelling capabilities, are developed to enable the modelling of various realistic aspects of manufacturing systems.

Martinez et al. [175] presented Petri nets and coloured Petri nets for the modelling and specification of FMS. It was concluded that these tools could be used both in simulation and as control models.

Alla and Ladet [8] investigated timed coloured Petri nets for the specification, validation and simulation of FMS. Conclusions were drawn that the same graphic tool could be used in all the phases of the system life, and the model could be changed from one operation to another without redefining.

2.5.2 Simulation Based Modelling

The term 'simulation' in the context of modelling for manufacturing systems refers specifically to computer-based discrete event simulation. This type of modelling mimics the detailed operation of a system through a computer program and/or other tools in order to provide adequately both operational and qualitative insights into how a manufacturing system could be designed and operated, or what procedures to run the system are better than others. Compared with approximate modelling, models of this class are capable of making detailed decisions and the information obtained from the models is greatly realistic and of significant practical use. However, considerable effort is required to acturally develop these models, and the data and computational requirements can be substantial.

To develop simulation models, four major approaches are available [209]:

- The event based approach. With this approach a program segment is written to define every event in the model. This involves defining the states the considered entity may enter following the event. Time does not advance within an event and the system behaviour is simulated by state changes that occur as events happen.

- The activity based approach. Here every activity that the entities in the model may do is defined by using two related events. This includes tests to determine whether the activity can be initiated at any point in time, and the state into which each entity passes after the activity is completed.

- The process based approach. It involves the construction of a process for each temporary entity within the model. A process is defined as the sum total of activities and events that the entity passes through whilst it is in the system. This approach can be considered as a combination of the event and activity based approaches to simulation.

- The three phase approach [276] [79]. This method consists of three basic phases at each time advance. First the clock is updated, and then the activities that can finish are finished. Finally, in a defined order, all activities that could start are tested, and if appropriate, they are started.

Shires [244] reviews these approaches and assesses their relative merits when applied to simulation of manufacturing systems.

Tools that have been developed using these approaches and can be applied for flexible manufacturing modelling can be broadly categorised as general-pupose simulation languages, generalised manufacturing system simulators and specific simulation models.

2.5.2.1 General-Purpose Simulation Languages

The general-purpose simulation languages can be defined as a class of packages, which can be used for a wide variety of purposes not merely manufacturing system, and usually consist of the basic functions together with an executive program within an event-based structure or is a specially developed high-level language with its own vocabulary and grammar dedicated to simulation [189] [188]. They all incorporate interactive graphics capabilities, and most are supported by code generators [176]. Accordingly, systems can be modelled to whatever degree of detail is necessary, though this power and flexibility requires experience and skill. Typical examples of this class of tools are SIMAN [205], GASP [211], SEE-WHY [10], FORSSIGHT [136], ECSL [74], SIMULA [131] and SIMSCRIPT [173], with the first four being collection of subroutines and procedures, and the last three being statement description languages.

2.5.2.2 Generalised Manufacturing System Simulators

Generalised simulators are a new form of simulation tools for manufacturing engineers. These may be defined as a class of simulators which can be rapidly configured, and usually consist of a validated model which the user configures to his own input data [161]. This is in contrast to the general-purpose simulation languages where the user must do some programming. In this approach the user provides only numerical data that is usually available in a data base or from feasibility data, therefore these simulators are usually referred to as data driven. More advanced users can also incorporate patches of code into the model to allow special features to be handled. This option can only be done with extreme care, and a thorough understanding of the model and its underlying assumptions is required.

Major commercially available generalised simulators include GCMS [271], SIM-FACTORY [61], WITNESS [141], MAST [164] [165], SAME [31] and those reported by among others ElMaraghy [100], Warnecke [287], Spur [253] and Iwata [143]. Figure 2.21 shows the relationship between some of these generalised simulators [189]. More detailed descriptions and comparisons of these simulation packages are given by Bevans [48] and Carrie [64] [65]. Figure 2.21 shows the relationships between some generalised simulators with regard to the generality of the tools [189].

The GCMS simulator allows a wide range of manufactuirng systems to be modelled. It can model various types of material handling systems and user defined assignment and scheduling rules.

Evolved from GCMS, the MAST simulator has been enhanced to incorporate SPAR and BEAM modules to provide an integrated simulation environment. It is relatively easy to use in the sense that within its capability no programming skill is needed. To design a model, the user, however, must be capable of editing a data structure by manipulating numbers corresponding to the type of resources and the problems the user has in mind. SPAR is a static analysis package which pre-processes data ready for MAST itself, and with BEAM the user can draw a physical layout of the model and see the dynamic movement of carts, parts, machines working in a colour graphic animation. A special feature which currently makes MAST quite unique in this class of modelling tools, is its capability of modelling control algorithms that the user selects from the library.

Similarly, WITNESS has defined a variety of input and output rules which the user enters through the detail menus. It is outstanding, however, that the push/pull manufacturing strategies can also be modelled by WITNESS.

When sufficiently detailed aspects of the real system are modelled by the simulators, they can be termed as emulators [60]. A major example of generalised manufacturing emulators developed to date is LUTE reported by Bell [43]. It is actually an integrated design system consisting of an evaluation phase and an emulation phase (Figure 2.22). In the evaluation phase, a rapid appraisal of system performance using average measures is facilitated based on a closed queueing theory model [251]. The emulation phase generates detailed dynamic information which allows fine tuning decisions to be made concerning system configuration and operation. The entities defined include part storage buffers, loading/unloading buffers, machines, tooling and automatic guided vehicles. Extremely detailed system characteristics like the direction of the rotation of a rotary buffer are modelled by the system. It has been shown that in the emulation phase, the total model can possibly be decomposed into a series of modules which can then be processed in parallel [244].

Recent extension of LUTE has incorporated modelling of various manual operations [161], modelling of integrated part and tool flow [181], and modelling of highly detailed multi-cell systems [1] in the emulation phase. Waterlow [288a] reviews the five significant ACME funded simulation projects and compares LUTE with the others.

2.5.2.3 Specific Simulation Models

These are the models developed using general-purpose programming or simulation languages to study the performance of particular flexible manufacturing installations. Among the earliest to use simulation models in the analysis of flexible manufacturing configurations were Weck & Schuring [291], Warnecke & Gericke [284], Hutchinson [138], Chan & Rathmill [68], Nof et al. [196], and Stecke & Solberg [257].

It was recoganised that simulation is an effective tool for evaluating various alternative FMS configurations. Simulation models which have been developed for this purpose are reported by among others Rathmill et al. [220], Martin & Pritsker [174], Carrie et al. [66], Mills [187], Browne & Rathmill [55], Rathmill & Chan [219] and Musselman [191].

Hutchinson & Holland [139] built a simulation model for evaluating systems with different degrees of flexibility. Wilhelm & Shin [296] concluded, by using a few simulation experiments that specifying an aggregate routing mix at the pre-release planning level may improve system performance. Stecke & Solberg [257] used simulation to validate their queueing network analysis [126] and indicate that pooling of machines may improve system performance and balancing machine workloads is not always the best loading policy.

Simulation also appears to be an appropriate tool for analusing the effect of various control policies. Release policies have been studied by Nof et al. [196], Iwata et al. [143], ElMaraghy [100] and Bilalis [49]. Loading and dispatching rules for the efficient operational control were tested by Nof et al. [196], Steche & Solberg [257] and Bell & Bilalis [41]. The interaction of vehicle dispatching rules and machine to vehicle allocation rules was studied by Egbelu & Tanchoco [99] based on an experimental simulation.

2.5.3 Perturbation Analysis

It has been recognised that both analytical method, such as queueing networks, and simulation have their inherent weekness as well as strength for analysing FMS [130]. As a result, perturbation analysis, which was initially developed by Ho et al. [132] [133] [134], has been introduced to the analysis of FMS.

Perturbation analysis is a hybrid method based on both simulation and mathematical analysis [266]. It retains the precision of a detailed simulation run while incorporating the efficiency of analytical techniques.

Suri and Cao [266] extended the early version of perturbation analysis approach to discrete event systems to model flexible manufacturing. The model can be used to derive gradients with respect to processing times and buffer sizes. To optimise the number of pallets/fixtures allocated to the production of a particular part type, the marked customer method and the phantom customer method were introduced [132]. It has been shown that the use of such a model enables efficient and accurate optimisation of FMS performance, particularly with respect to those factors that are not adequately covered by queueing networks models.

2.5.4 Artificial Intelligence in System Modelling

Artificial intelligence (AI) is the study of how to design and program computers to accomplish tasks that are accomplished by people using their intelligence [234a] [72] [226]. In many areas, this methodology is increasingly coming to be seen as an alternative to conventional approaches.

AI encompasses many different ideas and disciplines [234a] [226] [195] [37] [38] [77]. According to the scope of the domain knowledge, AI systems can also take the forms of knowledge based systems [3] and expert systems [106]. The relationship among these system variants is shown in Figure 2.23 [289]. Figure 2.24 [234a] shows a generic AI system architecture and a functional structure of AI is depicted in Figure 2.25 [186].

The main characteristics of AI systems are that they deal mainly with symbolic representations. They also use heuristics, cope with incomplete data, and often show learning abilities [234a]. It can thus be seen that AI attempts to model complex systems by making use of subjective and heuristic methods similar to those used by humans.

Elzas [101] [102] has discussed the relationships between artificial intelligence tools and mdelling and simulation techniques and concluded that a large degree of similarity exists between knowledge-strcturing paradigms for modelling and simulation and their conterparts in AI. In addition, there are mutual benefits for these two fields when the techniques of one field are applied in another field. These conclusions were supported by a more general comparison between AI techniques and operational reserach methods [208].

To highlight the similarities, Doukidis [89] has shown that the three-phase simulation model can be considered as a rule-based model. In fact, both simualtion models and knowledge based systems can be viewed within a common framework for modelling [27]. Both kinds of systems have a state characterization, state transformation operators, and input/output interfaces.

However, there are also differences between simulation and AI approaches [279]. One of these differences is that each field maintains a slightly different emphasis: dynamic behaviours for simulation and logical inference for AI. In traditional simulation, state representation has largely been numeric, while in AI is has been symbolic.

Recent development in both fields has shown that the two can be intergrated [121] [303]. Simulation has developed statistical and graphical output presentation while AI has focussed on explanatory output and natural language input. Besides, traditional simulation model processing has employed the forward chaining mode, whereas the inference engines of AI systems can run under both forward and backward chaining control. Time ordering and dynamic processes have been at the centre of simulation modelling, but AI has opened up the possibility of integrating traditional dynamic modelling with other symbolic forms of state transition representation such as causal inferencing [148] [201]. For a discussion on the potential use of AI in modelling and simulation, refer to [203].

2.5.4.1 AI Approaches to Modelling

The need to develop models within AI has led to the application of both AI methods and AI software tools to this development. These approaches to modelling include knowledge based simulation, planning, qualitative modelling, hierarchical abstruction, temporal reasoning, intelligent front-ends and expert decision-makers, and learning [200] [46] [245].

2.5.4.1.1 Knowledge Based Simulation

The application of AI programming paradigms in simulation has led to the development of knowledge based simulation systems. The simulation is constructed by using a knowledge based framework, with the system being simulated represented within a typical knowledge structure [45]. The inference mechanism commonly employed within the knowledge structure is extended by the addition of a time flow mechanism [245].

Oren [201] explored the use of AI to enhance simulation methodology and technology to make it a powerful tool for designing different types of complex systems. The application of programming with rules, logic and objects within simulation has been discussed by Bernemann et al. [46]. It is concluded that an integrated knowledge programming environment can make the process of modelling, simulation and analysis easier and more flexible.

Figure 2.26 summerizes the characteristics of major software tools that have arisen from AI efforts in simulation. ROSS [178] was developed by the Rand Corporation and is probably not only the first but also one of the most fully developed AI based simulation tools. It is a LISP implemented, highly interactive system. Object oriented programming serves as a basis for ROSS, where real world systems are modelled as objects, messages are passed between objects describing actions that are to be taken, and IF-THEN rules are used to describe behaviours each object may assume [154]. This system aids the user during model execution by displaying a trace of all messages passed during the simulation. Through selective filtering of trace information, the user can determine if the model is behaving appropriately. The user can at any time halt the simulation, investigate and modify the model, and continue the somulation.

Developed at Carnegie-Mellon University, KBS is also a LISP based discrete simulation system [223] [113] [221]. Outwardly similar to ROSS, it incorporates an object-oriented paradigm to describe the real world system to be modelled. Rules are used to describe the behaviours of each object. Unlike ROSS, KBS employees the use of a sophisticated knowledge representaion scheme. All entities in KBS are represented as SRL (a frame-based knowledge representation language) schemata which incorporate inheritance relations. Goals describing the performance criteria of model components may be attached to objects and KBS informs the user whether goals are met. KBS has also been designed to be used interactively, enabling the user to examine the designed model and its behaviour. This includes model creation and alteration, run monitoring and control, and graphics display. It also allows the user to model a system at different levels of abstraction [180], and to check the completeness and consistency of the model. In addition to the above capabilities, KBS has incorporated an automatic analysis mechanism to fine-tune the input parameters of a model in order to bring the values of output variables within a desired range [224]. An application of KBS for the modelling of a corporate distribution system in a large manufacturing organization is reported in [222].

IntelliCorp has developed SIM-KIT [123] which is written on the top of KEE (the Knowledge Engineering Environment) [161]. Frames are defined to represent simulation objects and then are used to build a simulation. As SIM-KIT runs on a dedicated LISP machine, icons have been developed for visual displays by using the high-resolution bit-mapped display. Generic simulations can be constructed [105], where specific instances can be created by manipulating icons representing classes of simulation objects. The behaviours of objects are represented by rules and methods.

T-PROLOG has been designed to provide a logic programming basis for simulation [114]. It is based on an underlying theory of simulation that is quite different from the previously described systems. The time handling primitives of simulation have been combined with the symbolic processing of AI into a PROLOG superset. The resulting system allows the user to construct a simulation model by writing first order predicate statements using available features of T-PROLOG, and to execute the model with the non-deterministic problem solving methods of PROLOG. Since PROLOG programs can backtrack, T-PROLOG is capable of backtracking in time so as to attempt different paths through the simulation. This allows for some simple goal-directed simulation, where the user can specify multiple model parameters and goals the model is to achieve. The system can automatically modify the model until the simulation exhibits some desired behaviour.

Developed by Artificial Intelligence Ltd., STEM is constructed on top of LOOPS [53], a knowledge engeering language which runs on the Xerox Workstation [302]. In STEM [32], libraries of classes have been defined for different types of nodes and processes for discrete event system. The user can develop a model by identifying the appropriate nodes and processes, placing the corresponding icons on the screen and connecting the icons interactively. Monitors can also be attached to these icons so as to collect particular performance statistics of the objects. Similar to SIM-KIT, STEM allows for animated flow of tokens through the connected network. An outstanding feature of this system is that a single node on the screen can be expanded to a more detailed network, or a complex network can be considered as a single node and be linked to the other parts of a simulation. This makes it possible for the user to develop large-scale simulation models using both top-down and bottom-up approaches, and especially meta models can be developed which may be applied for different purposes. The behaviours of nodes are represented using LISP procedures which are less comprehensible than rules.

An example of using an integrated object and logic oriented programming environment for modelling of complex systems is SIMYON [228] [229]. By employing the advanced knowledge representation methods, this system provides the ease-of-use characteristic of network simulation languages, and at the same time incorporate user-specific decision processes in a complex and flexible format by defining a library of logic objects. These objects, which are analogous to the nodes of network simulation languages, are the building blocks for modelling.

Other systems exist, such as HIRES [110], BLOBS [185] and V-GOSS [242], but these are either experimental or not widely available. For an overview of object-oriented simulation environments, refer to [4].

2.5.4.1.2 Planning

Planning is concerned with deciding on a course of action before acting [77] [226]. An example which is frequently used in the AI context to demonstrate planning techniques is the modelling of movement of blocks in the blocks world. This is functionally equivelant to the problem of modelling work and/or tool flow in a manufacturing system.

There are mainly four distinct approaches to planning [77]: hierarchical planning, non-hierarchical planning, script-based planning, and opportunistic planning. For a full review of planning techniques, see [274].

Hierarchical planning generates a hierarchy of representations in which the highest levels provide a simplification or abstraction of the plan and the lowest levels provide a plan sufficiently detailed to solve the problem. The advantage of the hierarchical method is that the search involved is greatly reduced by focusing exclusively on critical subgoals before attending to detailes. Examples of hierarchical planners include GPS [193], ABSTRIPS [230], NOAH [231], MOLGEN [259] [260] and NONLIN [273].

A non-hierarchical planner develops a sequence of problem- solving actions to achieve each of its goals. It may reduce its goals to simple subgoals, or it may use means-ends analysis to reduce the difference between the current state of the world and the desired goal state. Examples of non-hierarchical planning systems are STRIPS [109] [108] and INTERPLAN [272].

Script-based planning makes use of skeleton or stereotype plans that are prestored. The prestored plans contain outlines for solving many different kinds of problems, ranging in detail from extremely specific plans for common problems to very general plans for broad classes of problems. The planning process proceeds by first finding a skeleton plan for the given problem and then filling in the abstract steps in the plan with problem- solving operators from the particular problem context [234].

The opportunistic planning approach was devised by Hayes- Roth [128], based on the blackboard control structure. It uses the blackboard as a clearing house for suggestions about plan steps that are made by planning specialists. The ordering of the operators is developed piecewise, and parts of a plan can be developed independently.

An example of using AI planning techniques in FMS modelling is suggested by Smith [247].

2.5.4.1.3 Qualitative Modelling

Qualitative reasoning is concerned with understanding and automating the techniques by which a human being reasons about the physical world. Qualitative modelling is the process of describing all possible behaviours of a system when given any valid situation for that system [215]. It models a system by representing the relationships between parameters, identifying the effects of any actions in the system, and distinguishing between different states of the system.

In some sense qualitative modelling can be perceived as an approach to system dynamics when relationships are qualitative and can be incomplete. However, whether qualitative modelling will have an impact on the modelling of manufacturing systems is still uncertain [200].

2.5.4.1.4 Hierarchical Abstraction

A similar idea that has been carried over from AI into modelling is abstraction [226]. Human thinking can reason about a system over a number of different levels of abstraction. Thus it is possible to model a system over these various levels.

Ben-Arieh [45] applied this idea in his knowledge based simulation. Five main events were considered in the simulation: next-arrival, process-finish, end-assembly, machine-failure and machine-repair. Each event can trigger hidden lower level events, such as add-to-queue (Q), remove-from-queue (Q), choose-next- machine and so on. These lower level events can further be decomposed until the lowest level events are reached and directlly change the data structures. Each level also has a set of primitive events that the modeller can apply, and therefore the user is able to determine the various levels of modelling detail according to his needs.

2.5.4.1.5 Temporal Reasoning

Considerable interest in modelling human abilities to reason about time has led to the development of AI temporal-reasoning techniques. Charniak and McDermott [72] discuss two main approaches for reasoning involving time: Temporal System Analyzer, and Time Map Manager which can be either point-based or interval-based.

Vere [280] used ideas found in traditional PERT methods to allow a time window to be specified for any goal or action. External events are described as having some effect at a constant time. The planning system propagates the temporal links between these time windows and narrows them progressively as they become constrained by other actions. Vilain [281] and Allen [9] employed a time-interval logic to reason about temporal events. The approach can be considered as a general theoretical framework for modelling actions and their consequences.

Representing temporal knowledge and reasoning involving time are necessary in systems containing process modelling, and the use of the temporal-reasoning techniques in system modelling can be fruitful.

2.5.4.1.6 Intelligent Front-Ends and Expert Decision-Makers

The earliest use of AI approach to modelling is to produce intelligent front-ends for existing modelling packages or to combine conventional modelling methods with an expert decision- maker.

An intelligent front-end usually sits between the modelling package and the user. It is capable of generating necessary instructions or code to use the package, and interpreting and explaining results from the package [199]. However this approach can, in no way, result in any progress on adding intelligent behaviour to the model, since any flaws in the target package must be accepted. Examples of such a system can be found in [293], [111], [150] and [212].

Khoshnevis and Chen [150] reported on the construction of an intelligent interface for building SLAM [212] simulation models. Icon facilities were developed which aid the user with little or no knowledge of simualtion model building to construct a model.

Wichmann [293] reports on the integration of a simulation software with a knowledge based system. The intention of the knowledge based system is to reduce the complexity of the design process, to alleviate the risks in the FMS design, and to allow the user to easily and quickly model a proposed or an existing FMS by providing expert advice and consistency wherever necessary. Three knowledge bases have been built up corresponding to system capacity analysis, setting up of simulation goals and analysis of simulation results. The overall system structure is shown in Figure 2.27.

An expert decision-maker is usually embedded in an existing model to play the role of decision-making when required. This approach is very often used to model the decision mechanisms of a system or to develop control rules for the operation of the system. Examples of this type of systems include MPECS [301] and VISUALPLAN [246].

Wysk et al. [301] describe a cell supervisor control system which is a discrete module in a hierarchical control system. The objective of the system is to create good control strategies and to make control decisions during system run-time. The system consists of an integrated scheduling module, a simulator and a cell control module. The intelligent scheduling module generates potential scheduling alternatives based on real-time shop information and the scheduling knowledge. The simulator then evaluates alternative scheduling rules and selects the "best" one on the basis of the system performance. The cell control module finally receives a series of execution commands, which are generated according to the selected scheduling rule, to actually move the part in the system.

For a overall taxonomy for integrating simulation and expert systems, refer to O'Keefe [199].

2.5.4.1.7 Learning

Learning is a process through which explicit knowledge can automatically be acquired and therefore the performance of the system is improved [77]. This can be achieved, albeit in a rather limited sense, by adding some reasoning to an existing model. The result is that the model becomes really efficient and intelligent, e.g., the model itself can help identify important aspects that may not be obvious to the user, or automatically determine, in some sense, a "best" system design.

There are basically two approaches available to learning: man teaching approach and examples learning approach (Figure 2.29). In the first situation, the exact rules, procedures and data structures necessary for problem solving are presented to the knowledge base of the system and can be retrieved directly for subsequent use [77]. The second situation involves induction, in which the system is at first provided with various examples, and then the specific pieces of knowledge are generalised into higher-level rules, patterns or concepts [299].

Oren [202] developed a taxonomy of AI learning techniques and explored their implications in modelling and simulation.

An example of the man teaching approach is reported in [111] where learning is achieved by adjusting parameters based on the user's knowledge. Wysk et al. [301] employed the examples learning approach to induct generalised scheduling rules. First, systrem characteristics are attained from the status information. Then training instances are generated which relate the rules, the performance measures and the system characteristics together. After a series of inductive learning processes, a generalised rule is produced and sent to the knowledge base.

2.5.4.2 Applications of AI in Manufacturing System Modelling

Although not fully developed, there are applications of AI in implementing manufacturing system modelling tools at an experimental stage. On pridicting the trends in the development of these tools for mnaufacturing systems, Wichmann [294] has categorised them into the following:

- Expert system as a separate advisory system,
- Expert system integrated and interfaced with an existing manufacturing system simulator,
- Knowledge based manufacturing oriented simulators.

An expert advisory system is a decision support system which can give advice to the user about the use of a particular simulation language or contain knowledge about a problem domain which a modelling system is used to analyze.

An example of such a system is reported in [115], where a concept is described for the development of an FMS knowledge advisory system combining expert system, data base and simulation techniques.

When integrated with a manufacturing simulator, an expert system should have structural knowledge about the simulator, its data structures, formats and models in order to allow the simualtor to read or write data and check model consistency. In addition, it should contain strategic and heuristic knowledge about the manufacturing system, and this would allow the user to perform a goal driven simulation, where the objectives of the user will dictate the appropriate design of the model, the experiment to be run with it and the analysis to be performed [293].

Developed by Ford and Schroer [112], EMSS (Expert Manufacturing Simulation System) has incorporated a Natural Language Interface, a Simulation Writer and a Simulation Analyzer with the SIMAN simulation language. The Natural Language Interface accepts input in normal English sentence form. The Simulation Writer is used to convert the output of the Natural Language Interface into the SIMAN simulation code by using the modelling knowledge about the manufacturing environment, debugging techniques, and the simulation language itself. After processing the SIMAN code, results are returned to the Simulation Analyzer which chekcs for model efficiency, needed improvements and general what-if situations, and revises the necessary parameters and code and then executes the new simulation program.

An expert system for FMS design has been developed on a SYMBOLICS 3670 LISP Workstation [277] using KEE [161] as a software development tool. The system [177] analyzes the output (such as utilization, queue length and cost estimates) from an FMS simulation model, determines whether operational and financial objectives are met, identifies design deficiencies, and proposes designs which will overcome identified deficiencies.

Knowledge based manufacturing simulators are one step further away from conventional manufacturing simulators [164] [43]. They usually have a structure similar in concept to the knowledge base of an expert system, where for example, the control logic of a flexible manufacturing system is represented as rules. They also have a separate data base with the description of entities in the model represented as objects, and a separate control structure which works similar to the inference engine of an expert system.

There has been a strong trend toward developing knowledge based simulation models for the design and operation of flexible manufacturing. Ben-Arieh [44] [45] reports on an investigation into the routing of jobs in a multi-cell FMS using a knowledge based system approach.

The methodology employed by Shivnan and Browne [245] is similar to Ben-Arieh's, though they use OPS5, a rule based language, to implement their simulation. The usefulness of this AI based simulation is demonstrated in the real-time control of advanced manufacturing systems. Conlusions are drawn that, unlike conventional simulation, the emphasis of knowledge based simulation is on information flow rather than entity flow.

Strandhagan [263] reports on the use of SIMULA object-oriented programming language in simulation of manufacturing systems. The system has incorporated five components around the simulation kernel which executes the simulation process. These components are the layout modeller, the entity flow modeller, the analysis and computational tools, the knowledge based layout analyzer and the knowledge based entity flow analyzer (Figure 2.28). The layout modeller allows the user to define the physical structure of a system by picking resource and requisite objects from an icon library. The icons have a set of parameters associated with them, and the user is asked to enter the values for these parameters. With the entity flow modeller, entity objects are identified and parameter values specified. The interrelations between the entity and resource or requisite objects are defined.

The analysis and computational tools are used to perform the statistical analysis of the results. The knowledge based layout and entity flow analyzers (Figure 2.28) are expert advisory systems for model modifications, where rules are applied to the objects in turn in order to serch for the ones which need to be modified and the system automatically implements the modifications in the model.

2.5.4.3 Software Tools

To construct an AI system, tools are required that can aid the building process. These tools consist of the languages, processes, and constructs that allow the acquisition, representation, storing, transformation, and other manipulation of concepts and relationships by information processing machines [234a]. They range from very high-level programming languages to low-level support facilities, and can be divided into four major categories [289] as shown in Figure 2.30.

The programming languages used for AI applications can generally be placed in the categories of either problem-oriented languages or symbol-manipulation languages. Examples of the former are FORTRAN, PASCAL and other conventional numeric programming languages, whereas LISP [300] [73] and PROLOG [75] are the major instances of the latter. Symbol-manipulation languages are especially designed for AI applications [234a] [72].

A knowledge engineering language is a sophisticated tool for developing AI systems, which consists of an AI building language integrated into an extensive support environment. Knowledge engineering languages fall into two classes: skeletal systems (or expert system shells) and general-purpose systems. A skeletal knowledge engineering language is simply an AI system with its domain-specific knowledge removed, leaving only the inference engine and support facilities [278] [94]. A general-purpose knowledge engineering language is an AI building tool which incorporates features that make it applicable to different problem areas and types. Typical examples of this type of system include LOOPS [53] [also see Appendix IV] and KEE [161].

The system-building aids are programs that help acquire and represent the domain-specific knowledge, or programs that help design an AI system under construction [289]. These programs address very differnt tasks, but major existing aids can be classified as either design aids, such as AGE [194], or knowledge acquisition aids, such as TEIRESIAS [83]. The former helps the knowledge engineer design and build an AI system by providing him with a set of building blocks which support various AI frameworks, while the latter helps transfer knowledge from a domain expert to a knowledge base of the system. Compared with programming and knowledge engineering languages, quite a few system-building aids are available.

The last category of tools, the support facilities, are tools associated with a knowledge engineering language for helping with programming or for enhancing the capabilities of the finished system. These may include sophisticated debugging aids, friendly knowledge base editors, and advanced built-in input/output and explanation devices [289].

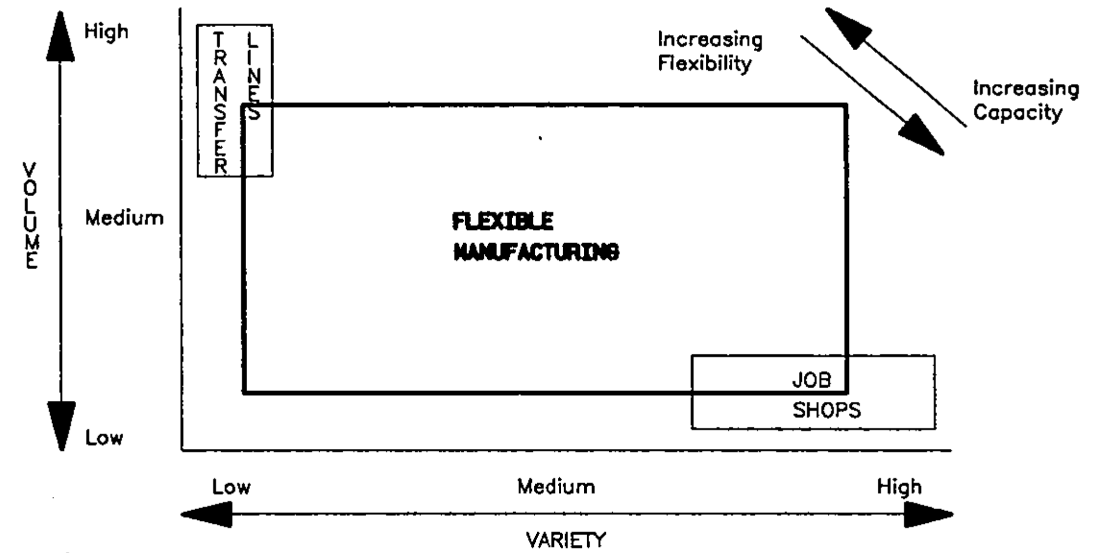


Figure 2.1	*Flexible Manufacturing for Batch Production*	**LUT - FMS Research Group**

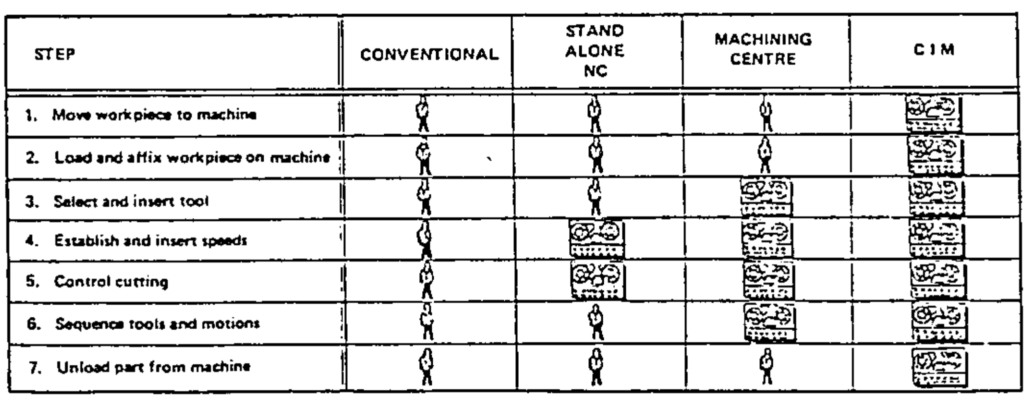


Figure 2.2	*A Comparison of Production Methods*	**LUT - FMS Research Group**

Characteristic	**FMS**	**Conventional Production**
Lead time	Short – hours	Long – weeks
Setup time	Short – minutes	Long – hours
Batch size	Small – customer oriented	Large – stock oriented
W.I.P.	Low	High
Throughput time	Short – can accommodate rush jobs	Long – rush jobs cause delays
Scrap	Low – zero defect policy	Too high

[169]

Figure 2.3	*Comparison of FMS with Conventional Systems*	LUT - FMS Research Group

Manufacturing Costs – Costs attributable to the manufacturing operation and the activities and facilities supporting the manufacturing operation. Manufacturing costs include fixted costs (such as costs of producti production equipment, factory building and facilities), direct costs (such as material costs and direct labour) and overhead costs (which include supervision, maintenance, in–process inventory, material handling, utilities, etc.).

Delivery Performance – The ability of the system to meet production schedules, both during normal opearating conditions and during transitions in which the system is adjusting to new conditions, demands and circumstances.

Flexibility of the Production System Relative to Machine Breakdowns – The ability of the system to adjust to breakdowns with minimum production losses.

Top Management Involvement – The positive or negative degree of interest, enthusiasm, support or participation of management towards the production system or towards the staff which supports the system

Group Morale – The degree of involvement, commitment and decision–making which the individual shares with the co–worker towards the functionaing of the production system.

Figure 2.4	*Factors for the Comparison of Manufacturing Methods*	LUT - FMS Research Group

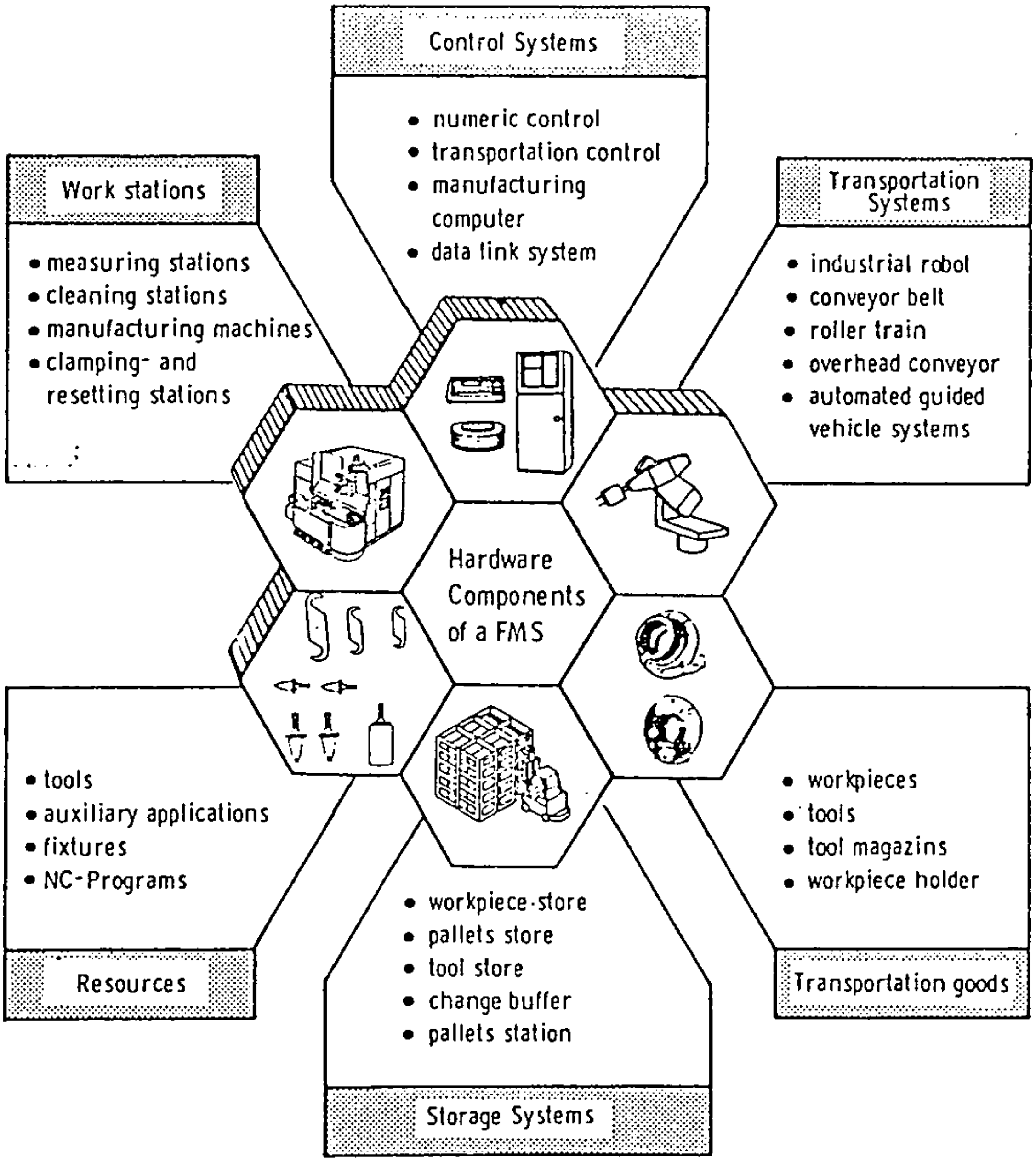


Figure 2.5	*Hardware Components of Flexible Manufacturing*	LUT-FMS **Research Group**

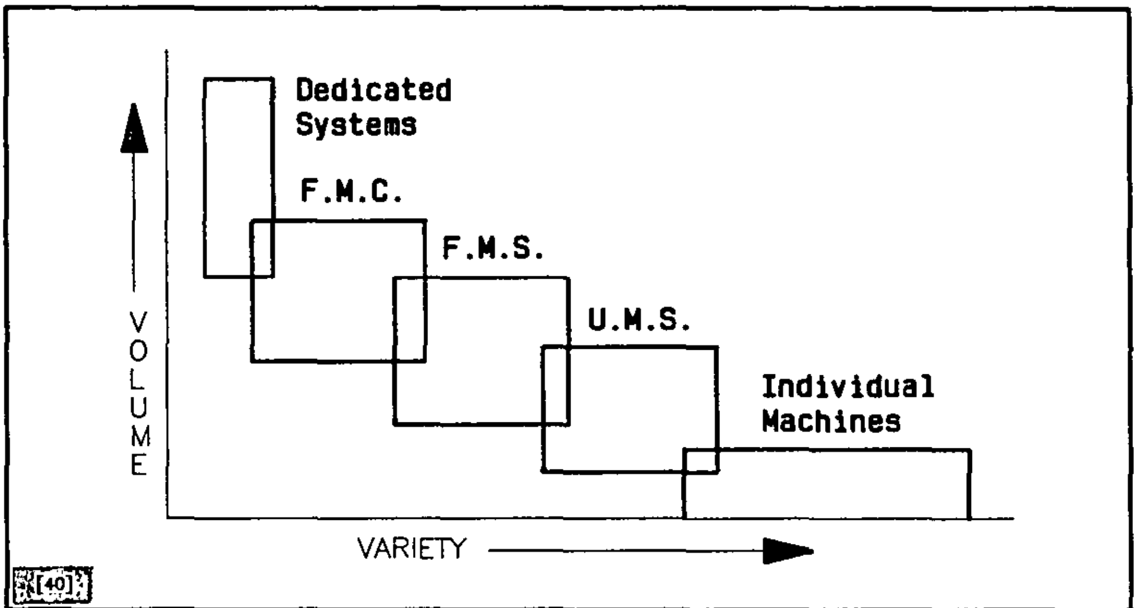


Figure 2.6	*Volume, Variety and Manufacturing Systems*	LUT - FMS Research Group

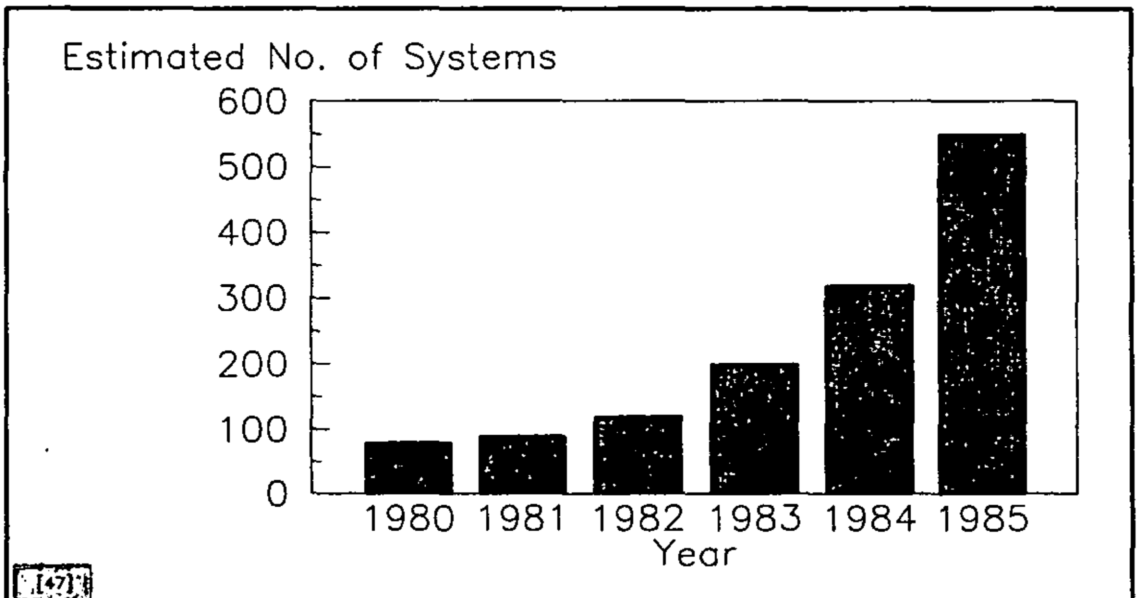


Figure 2.7	*Worldwide Growth in Use of Flexible Manufacturing*	LUT - FMS Research Group

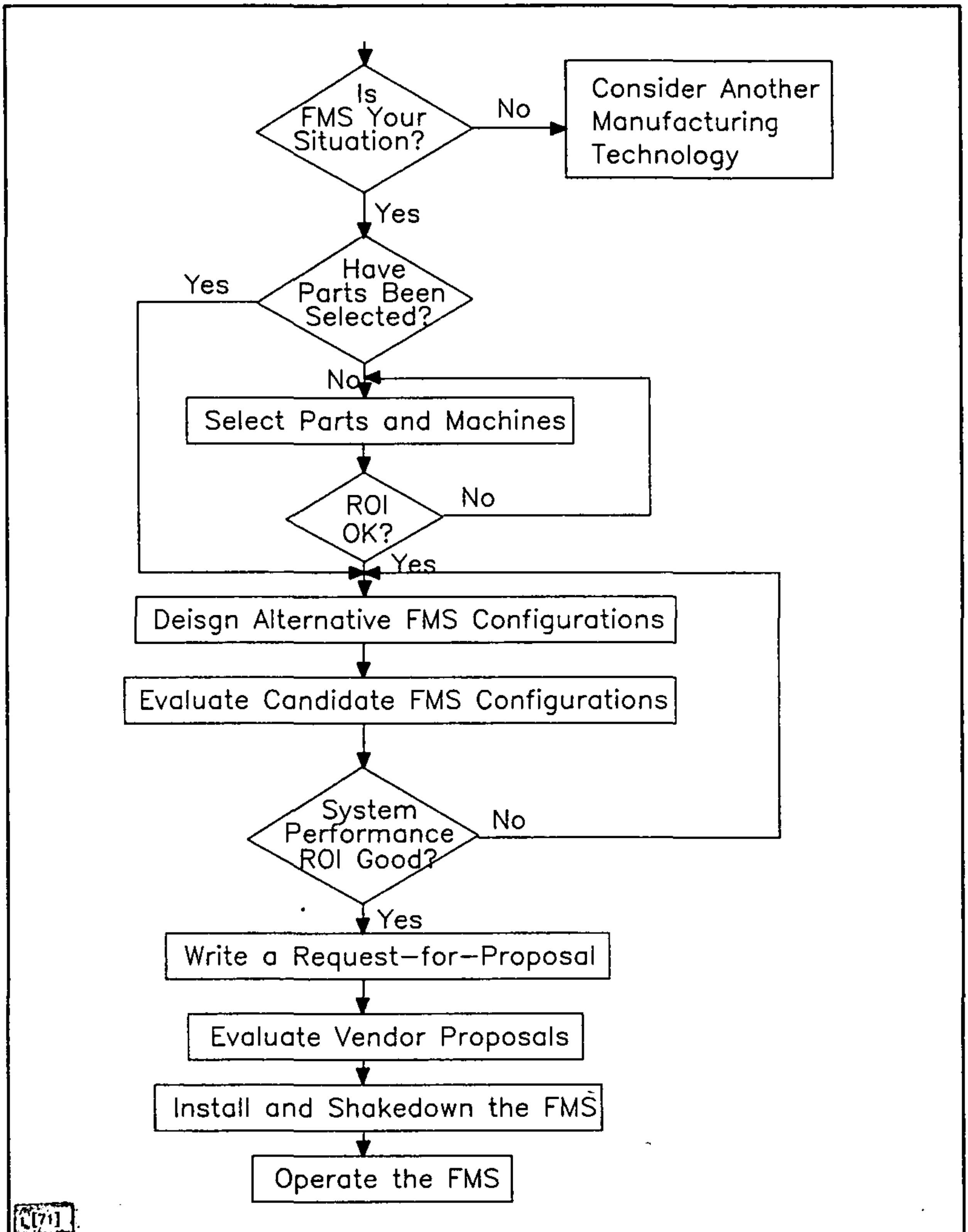


Figure 2.8

Decision Flowchart for the Acquisition of a Flexible Manufacturing System

LUT-FMS
Research Group

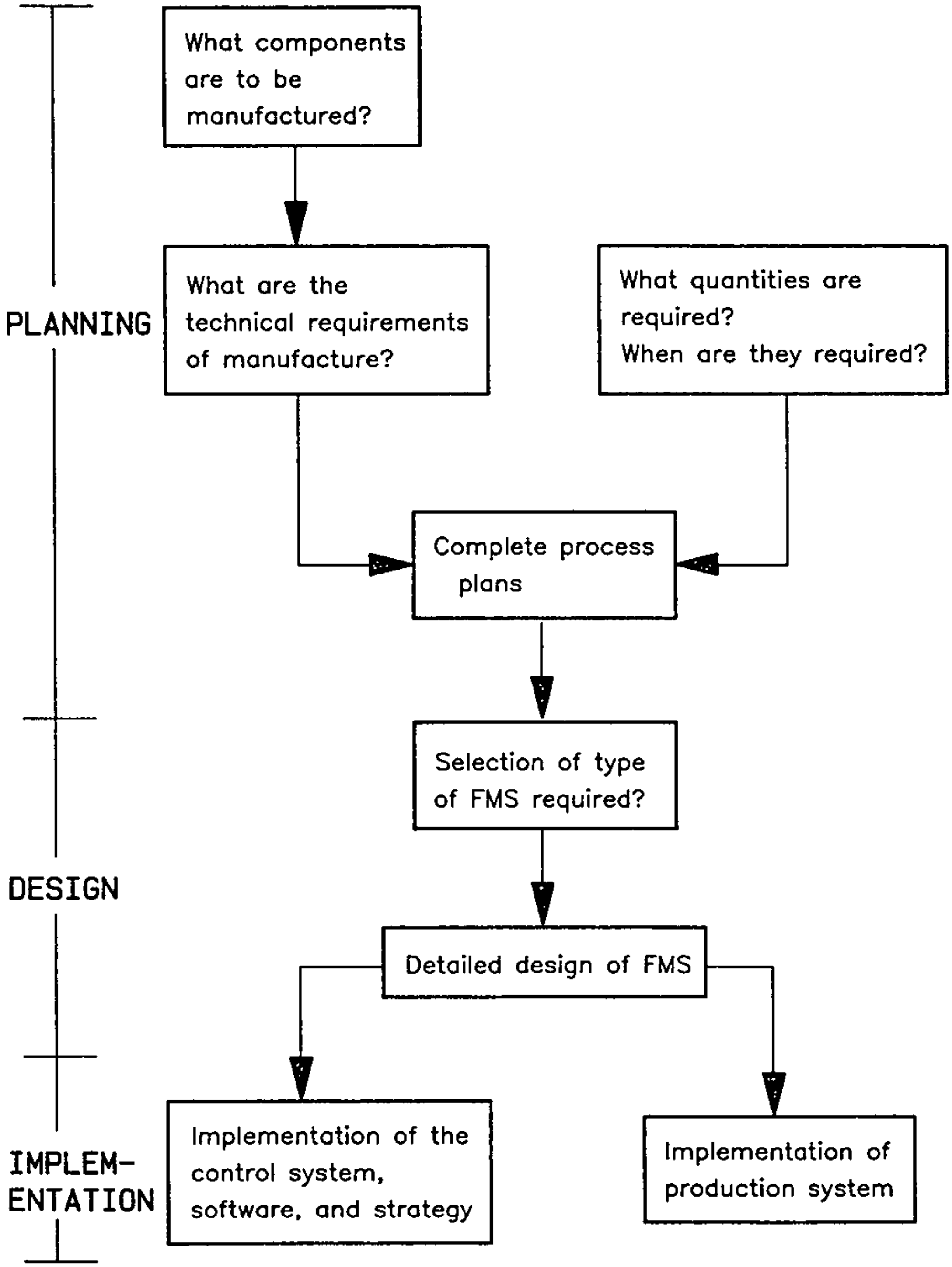


Figure 2.9

Overview of the FMS Design Procedure

LUT-FMS Research Group

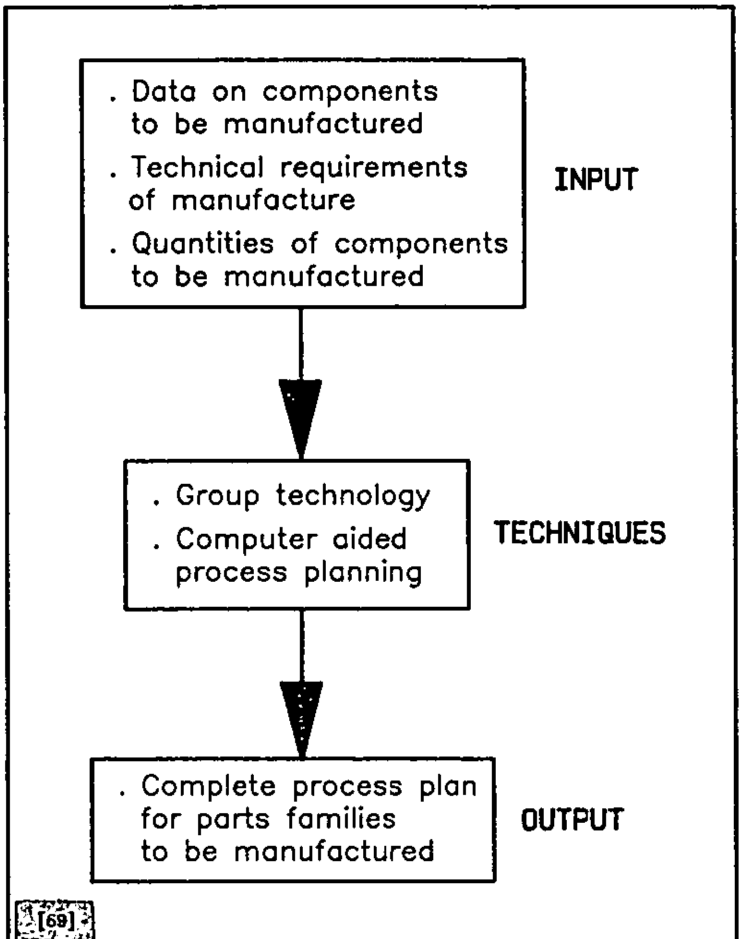


Figure 2.10	*The Planning Stage*	LUT-FMS Research Group

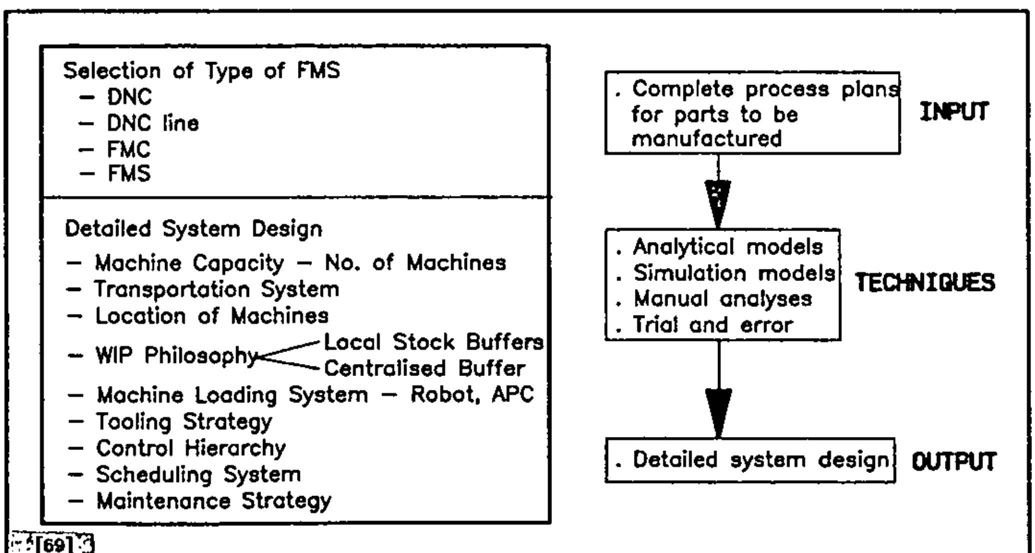


Figure 2.11	*The Design Stage*	LUT - FMS Research Group

Implement Control Strategy	**Implement Production System**
– DNC System (Machines, Materials Handling Materials – Production Scheduling System – WIP Control – In Process Inspection – Process Monitoring – Management Repairs Generation – Tooling Monitoring	– Design of Fixtures/Pallets – Robots/APCs Specification – Materials Handling Spec. – Tooling System – Inspection System – Machine Specification

[69]

Figure 2.12	*The Implementation Stage*	LUT - FMS Research Group

The Initial Specification Stage involves the specification of the following parameters:

(1) The range of part types
(2) The process plan, and the numbers and types of machine tools and robots
(3) The types and amounts of flexibilities
(4) The type of manufacturing systems
(5) The type and capacity of material handling systems
(6) The type and size of buffers
(7) Computer control hierarchy

The Subsequent Implementation Stage includes the making of the following decisions:

(1) The layout of the system
(2) The number of pallets
(3) The type, number and design of fixtures
(4) The general control strategies
(5) The system control software

Figure 2.13	*The FMS Design Problems*	LUT-FMS Research Group

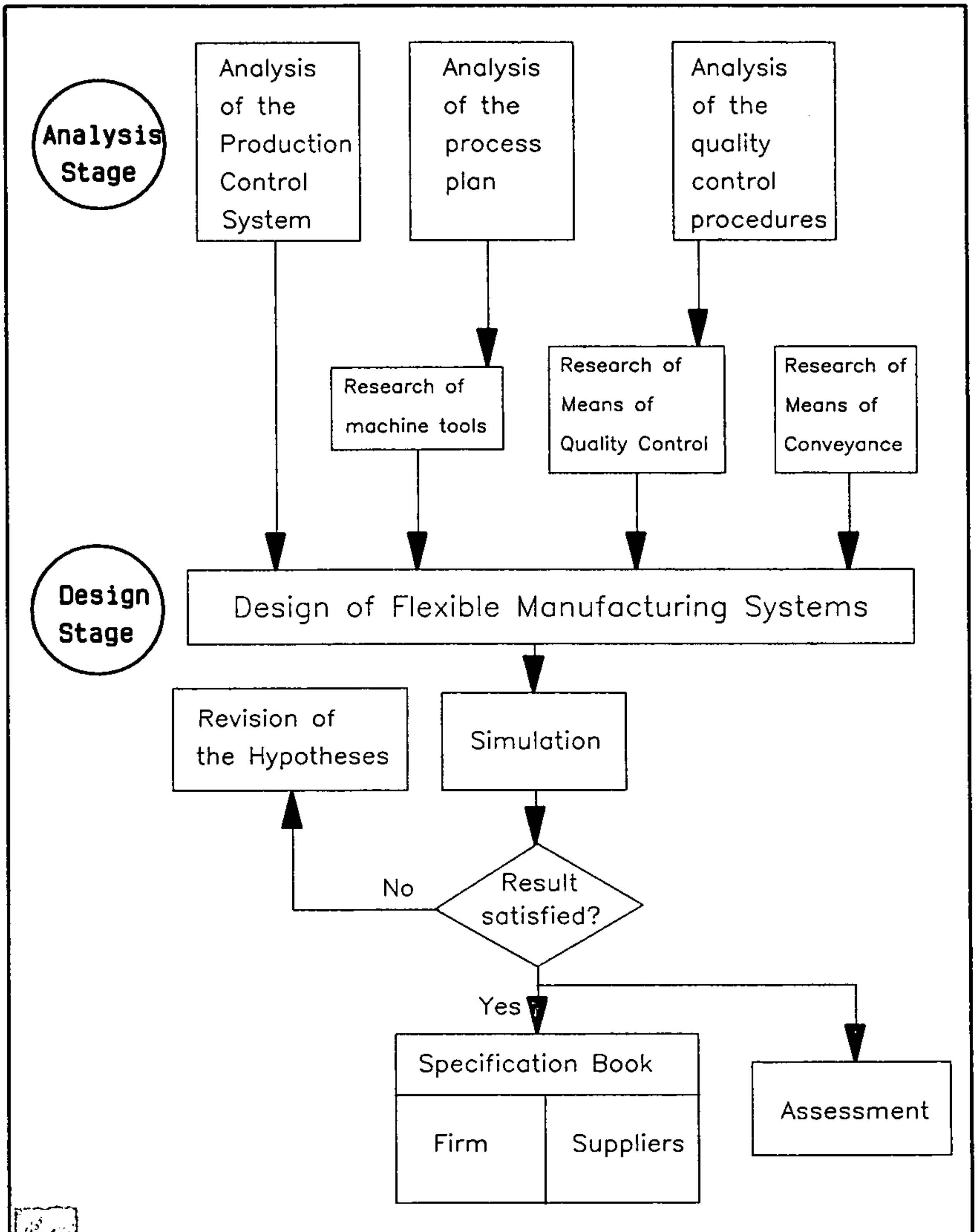


Figure 2.14	*Structure of an Integrated Design System for FMS*	LUT-FMS Research Group

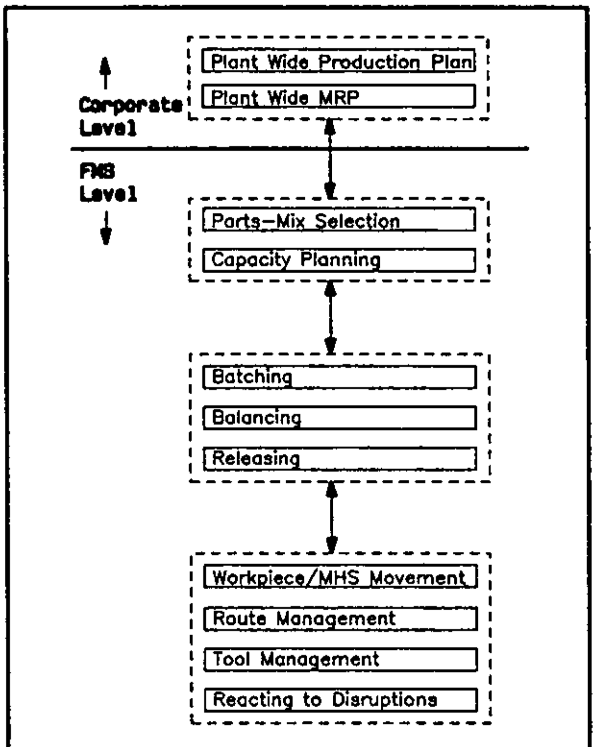


Figure 2.15 | *The Three-Level Control Hierarchy for Flexible Manufacturing Systems* | **LUT-FMS Research Group**

. Interdependent consideration of candidate orders and requisite tooling at machines, whereby mutual viability is recognized.

. Linked groups of orders such that release is conditional on the entire group attaining feasibility.

. Recognition of tool magazine capacity as a soft or firm constraint.

. The availability of duplicate tooling for assignment to different machines.

. Allocation of machine hours required by each order at various machines subject to availability and prevailing policies on desired utilisation and scope for overtime operations.

. The option to use alternative process routings using equivalent machines in systems with such flexibility.

. Consideration of the different due dates for completion of orders so that the relative urgency for order release can be taken into account.

. The volume of orders released may require regulation with respect to some threshold, corresponding to some physical capacity, or it could be a target for desired throughput.

. The need to expedite certain specific orders may arise from time to time and hence provision to satisfy such forced selection in the planning framework is desired.

Figure 2.16 | *Operational Considerations in a Pre-Release Planning Model* | **LUT-FMS Research Group**

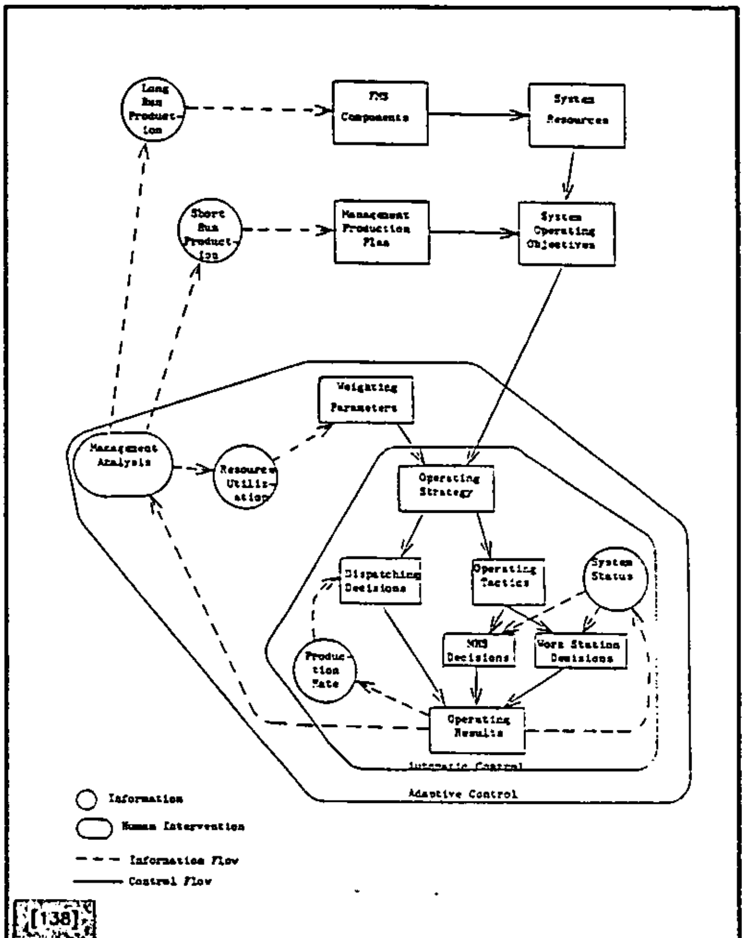


Figure 2.17	*An FMS Control Structure Including Both Automatic & Adaptive Control Mechanisms*	**LUT-FMS Research Group**

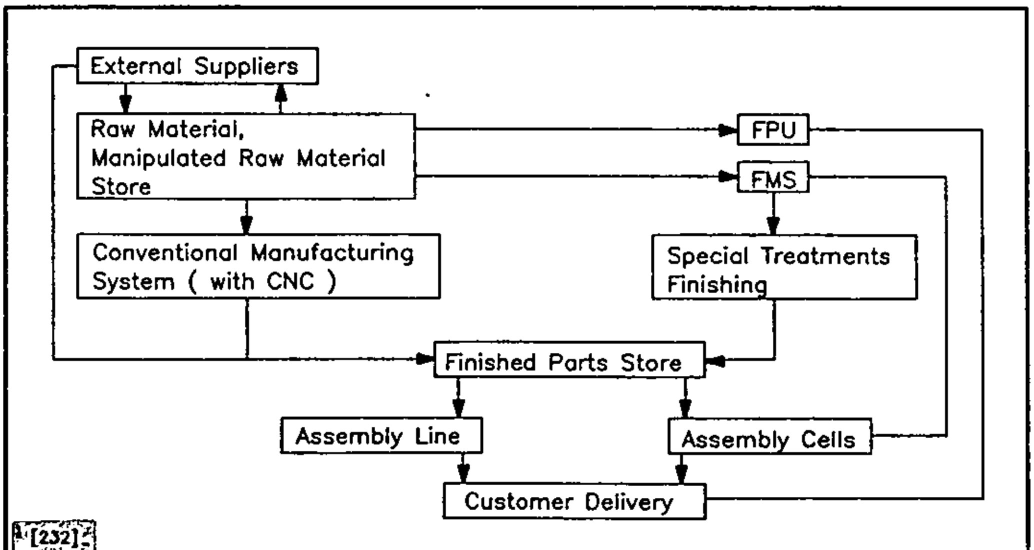


Figure 2.18	*Material Flow in a Mixed Flexible and Conventional Manufacturing System*	**LUT - FMS Research Group**

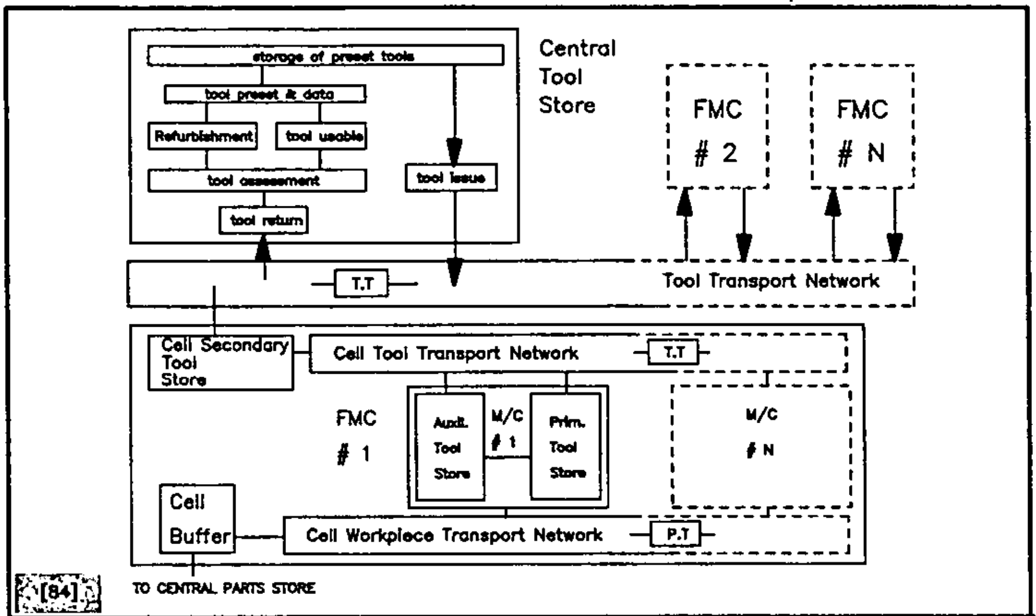


Figure 2.19	*Structured Representation of Total Factory Tool Flow*	**LUT - FMS Research Group**

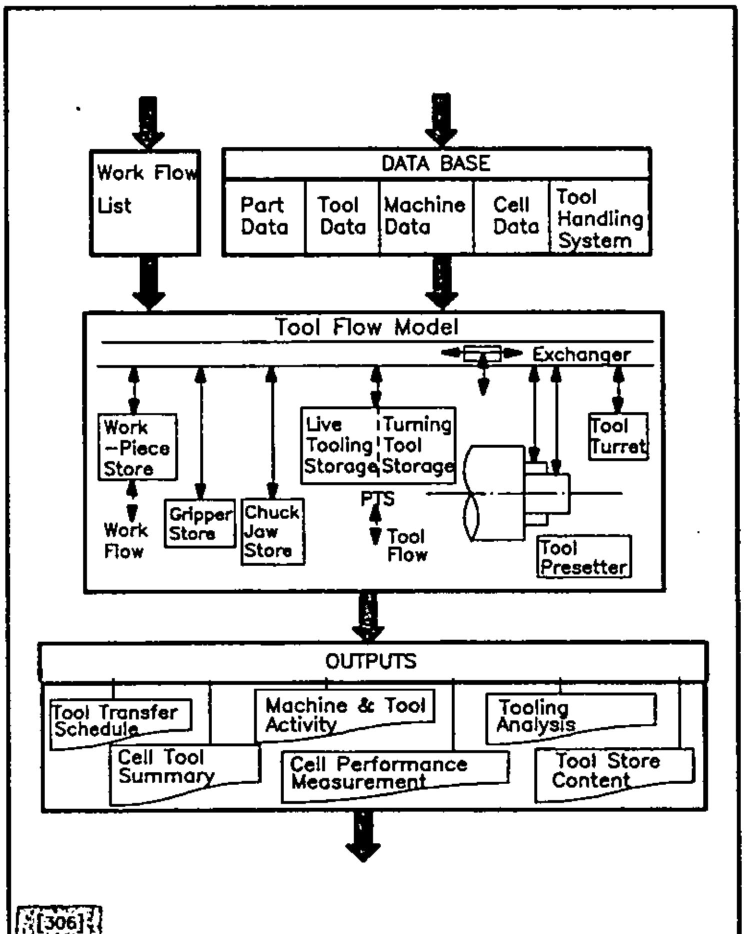


Figure 2.20	*Cylindrical Parts Model*	**LUT-FMS Research Group**

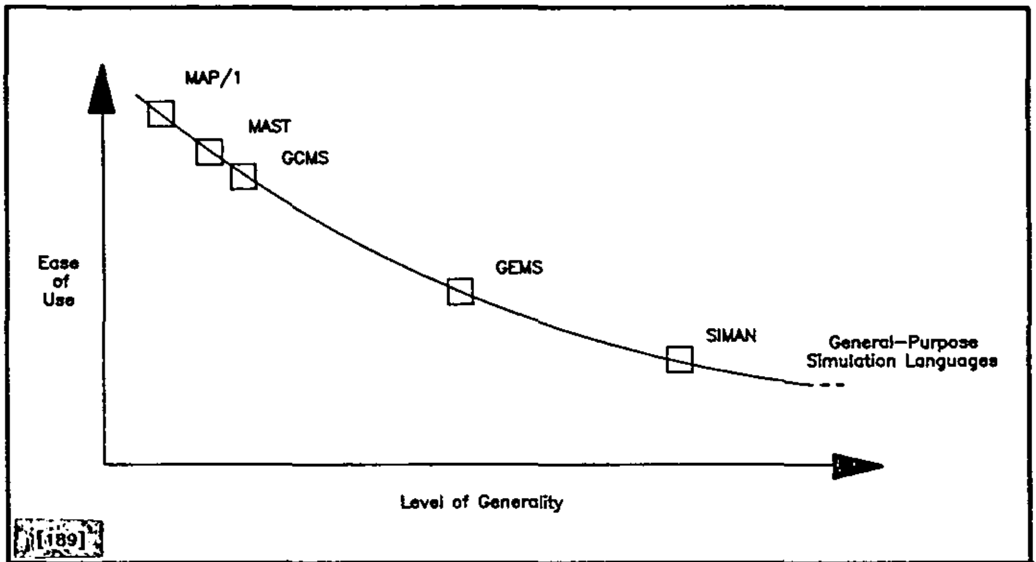


Figure 2.21	*Relationship between Some Generalised Simulators*	**LUT - FMS Research Group**

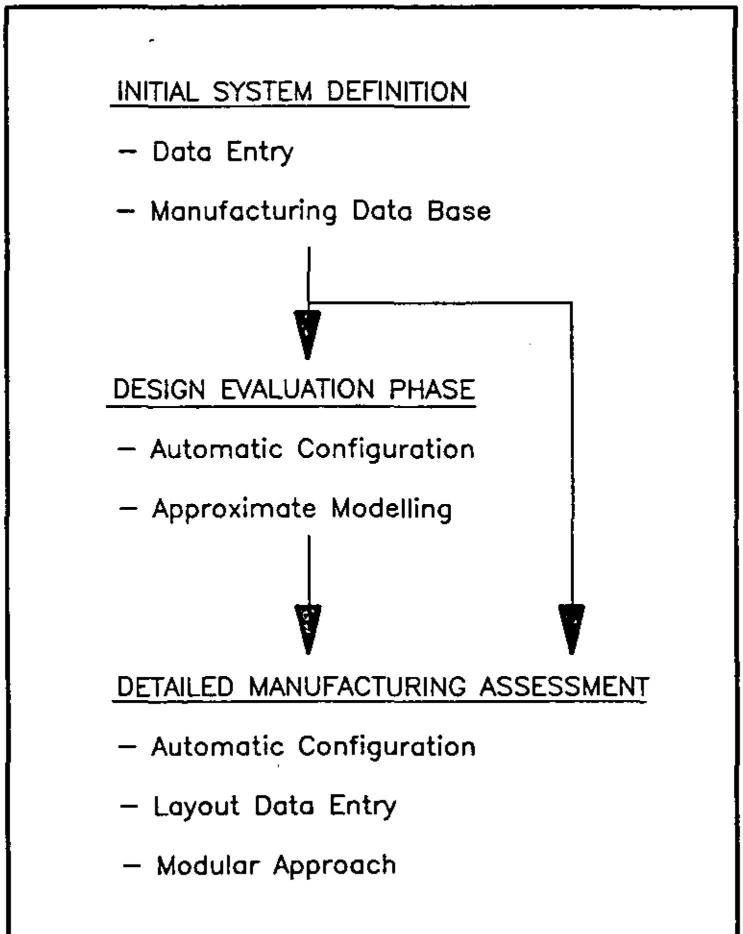


Figure 2.22	*Operational Structure of LUTE Prototype*	**LUT-FMS Research Group**

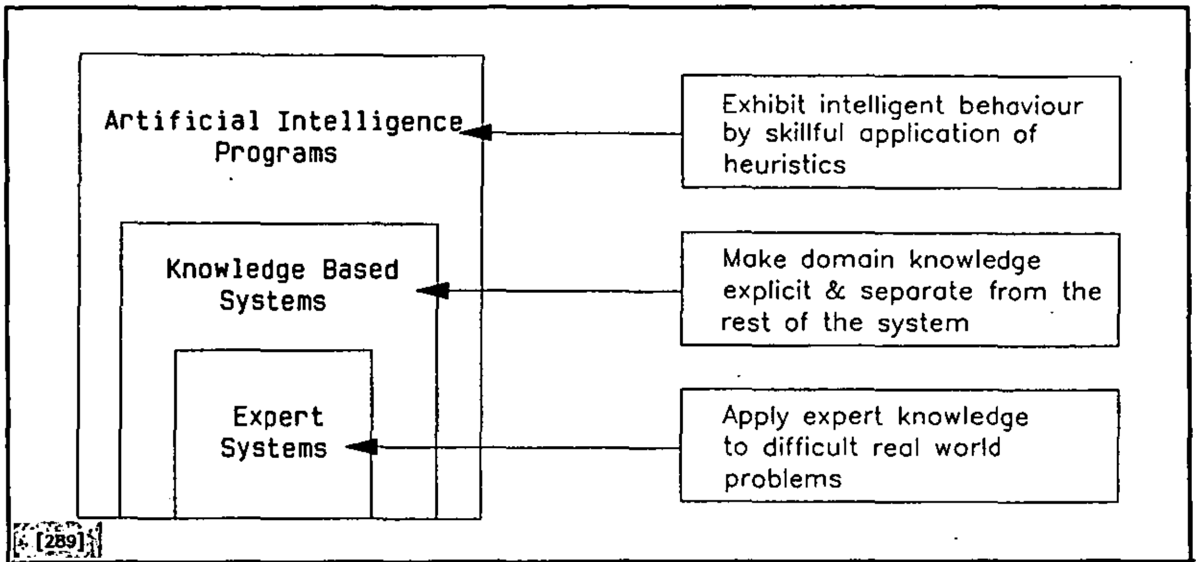


Figure 2.23 — *AI, Knowledge Based Systems and Expert Systems* — LUT - FMS Research Group

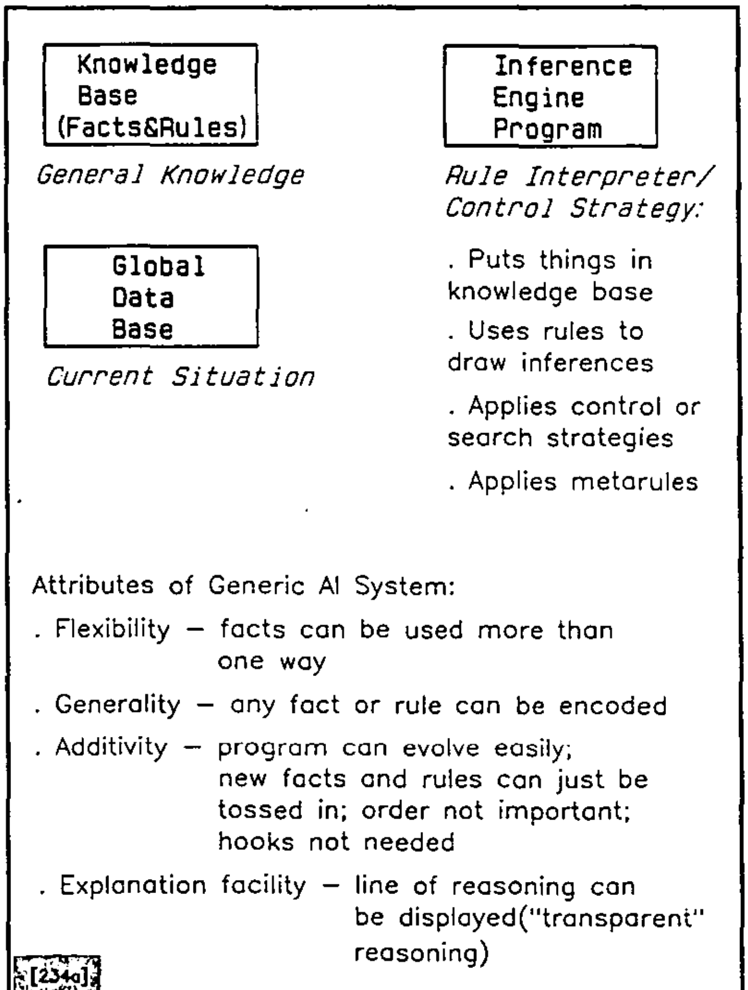


Figure 2.24 — *Generic AI System* — LUT-FMS Research Group

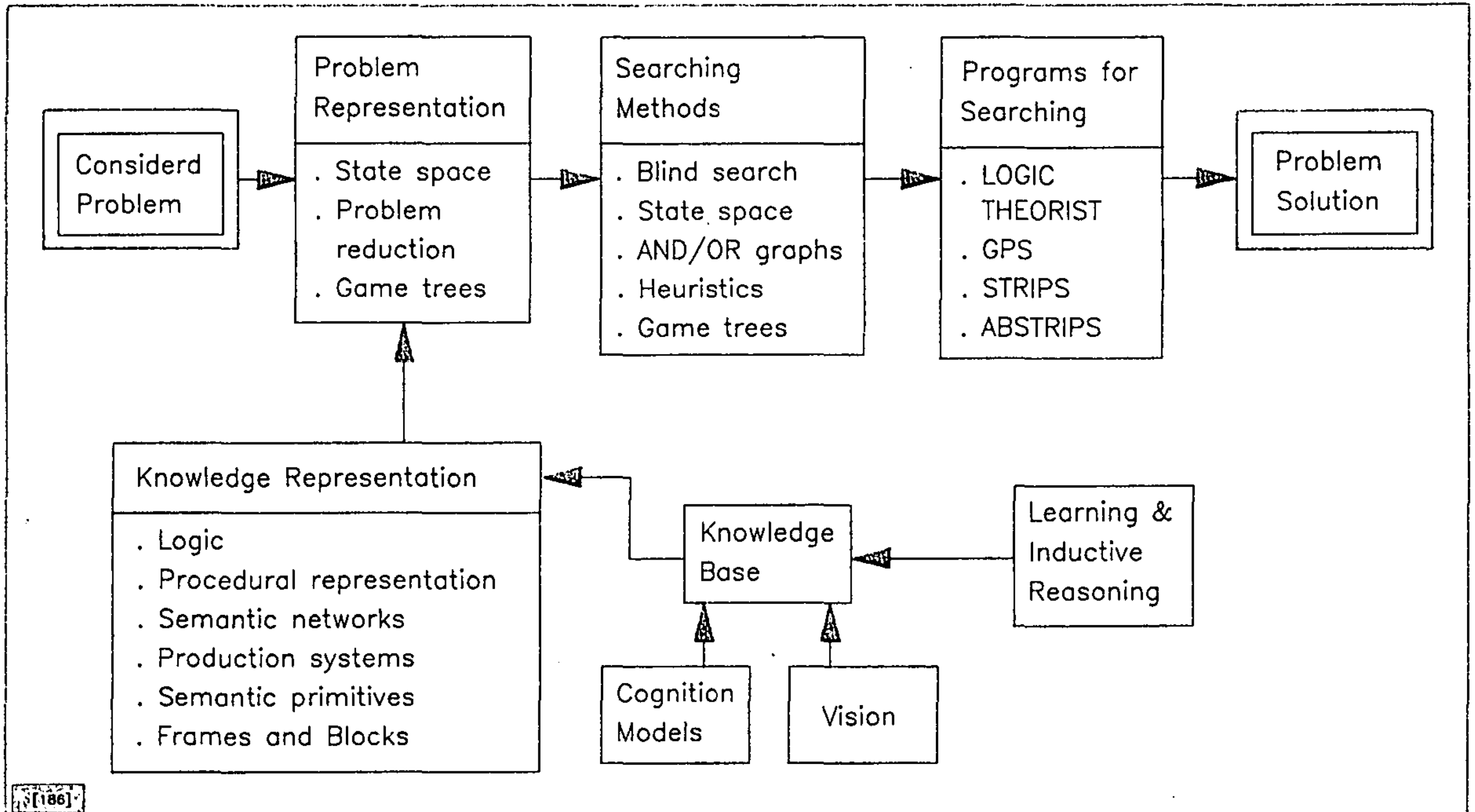


Figure 2.25

Functional Structure of Artificial Intelligence

LUT - FMS

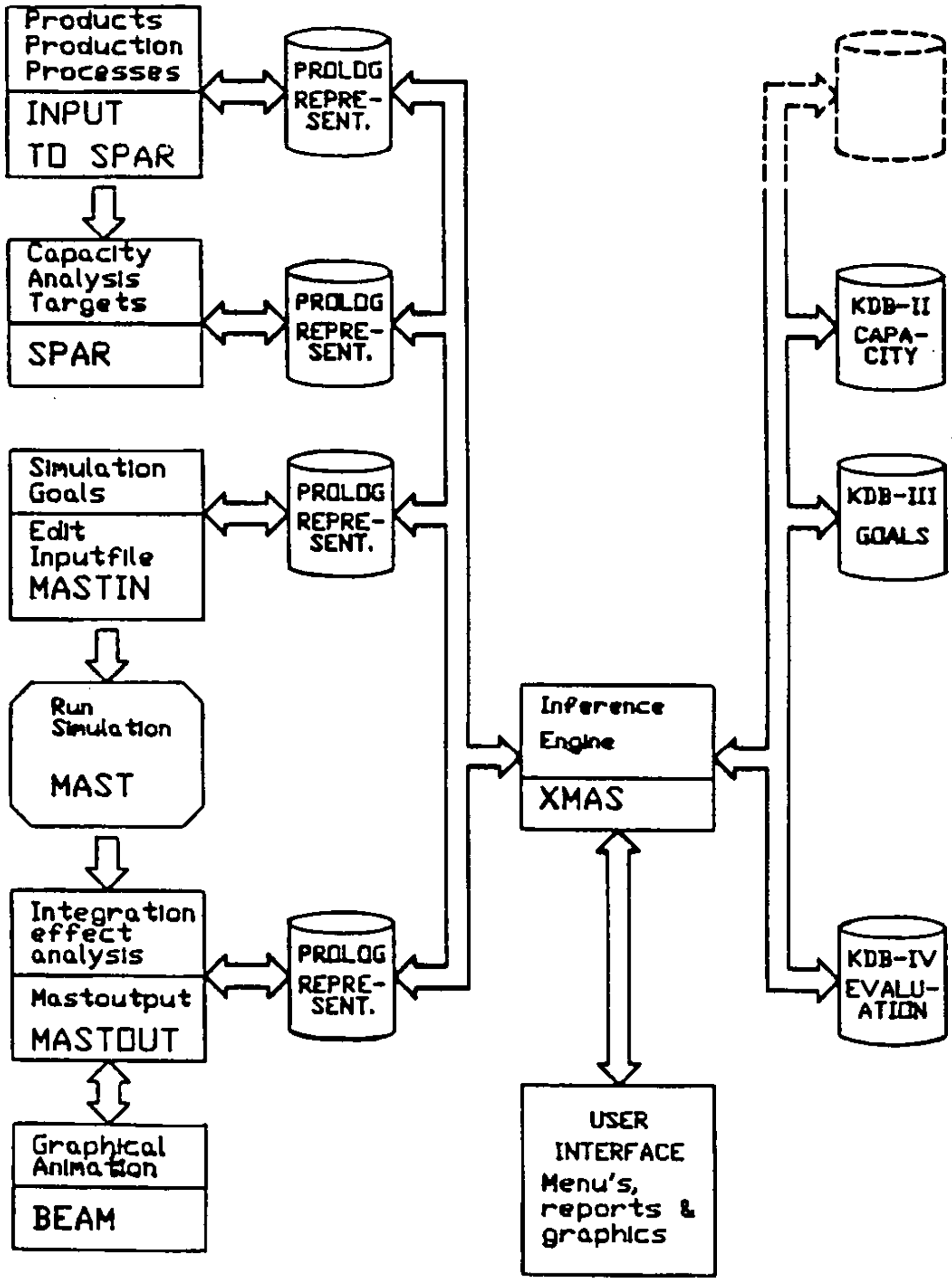
Research Group

Features \ Tools	ROSS	KBS	Sim–Kit	T–Prolog	STEM
Developer	Rand Corporation AI Group	Carnegie–Mellon University	IntelliCorp	Hungarian Institute for Coordination of Computer Techniques	Artificial Intelligence Ltd.
Software Implementation	Lisp	SRL	KEE	M–Prolog	LOOPS
Main Knowledge Represent. Methods	Objects Rules	Frames Rules	Frames Rules	Prolog Clauses	Objects Procedures
Hierarchical Modelling Capability	No	Yes	No	No	Yes
Goal–Oriented Modelling Capability	No	Yes	No	Yes	No
Icon Facilities for Model Building	No	No	Yes	No	Yes
Graphic Output Facilities	Animation	Windows	Animation	No	Animation
Automatic Analysis of Results	Good	Sophisticated	Basic	No	Good
Explanation of Simulation Processes	Good	Basic	Don't Know	No	Good
Learning Capability	Planned	Sophisticated	No	No	No

Figure 2.26

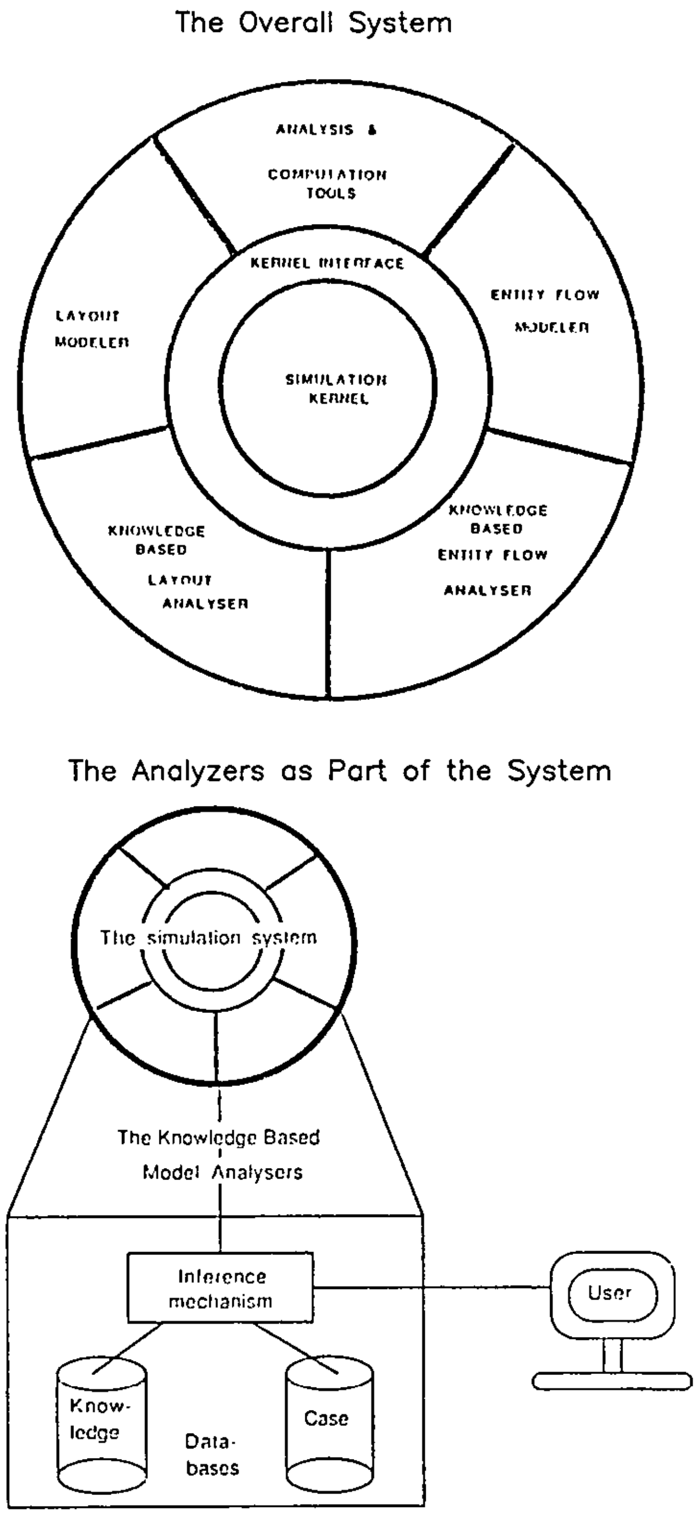
Overview of Commercially Available AI Software Tools for Simulation

LUT–FMS Research Group



[293]

Figure 2.27	*An Intelligent Simulation Environment for FMS*	LUT-FMS Research Group



[283]

Figure 2.28	*Structure of an Integrated Intelligent Manufacturing System Simulator*	LUT–FMS Research Group

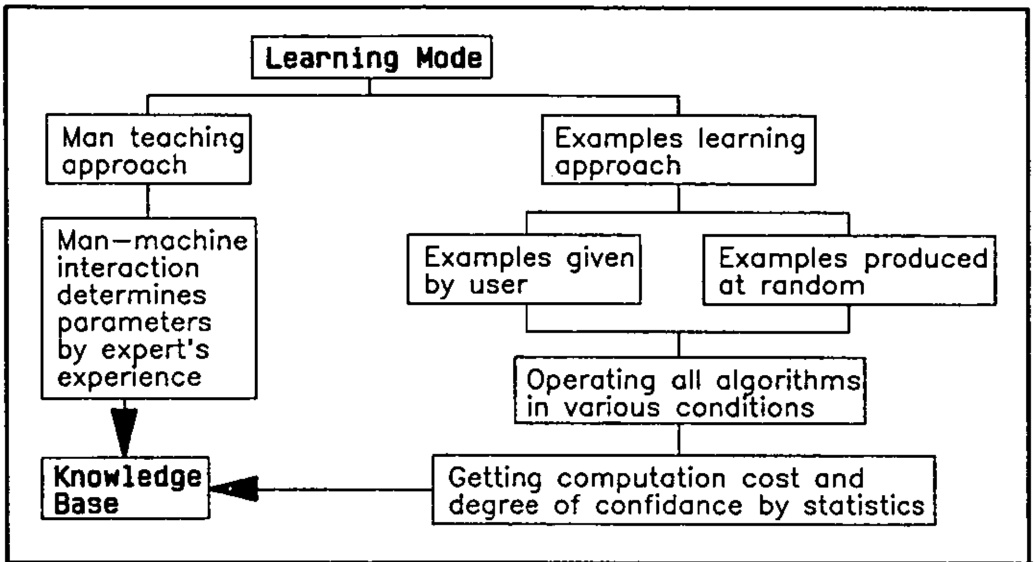


Figure 2.29 — *The Learning Mode* — LUT - FMS Research Group

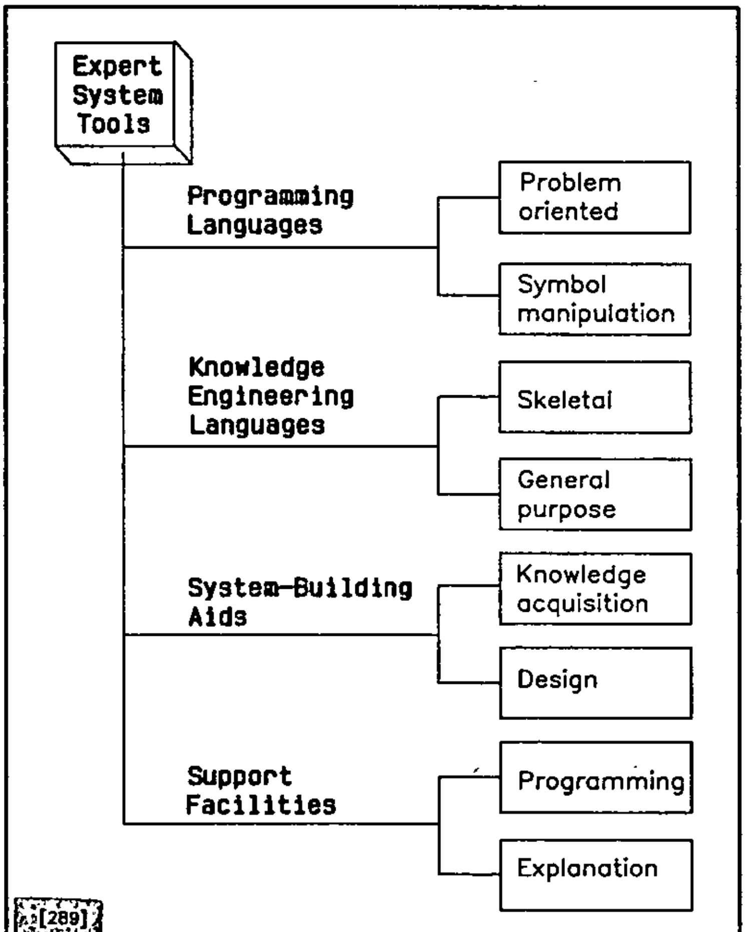


Figure 2.30 — *Classification of Expert System Tools* — LUT-FMS Research Group

Chapter 3
STRUCTURE AND PERFORMANCE OF FLEXIBLE MANUFACTURING FACILITIES

3.1 Introduction

This chapter is concerned with the major forms of flexibly automated batch manufacturing systems which are currently in use. Each of the particular system variants is examined with examples being quoted of current installations. In particular the emphasis of the chapter is directed toward studying the structure and the corresponding performance of these system configurations.

3.2 Concepts of Manufacturing System Configuration

Today's ever changing manufacturing requirements and the resultant need for flexibility make part variety and volume a fundamental concern in the design of a manufacturing system [40]. The development of new technology, including machine design, material handling methods and control techniques, has provided a solid fundation for meeting the requirements.

In principle, the basic choices of system design are dictated by the production volume per part number and production variety, whereas the size, configuration, commonality and life cycle of parts determine the processing, tooling, fixturing and machinery of the system (Figure 3.1) [15]. Apart from this, the phased implementation of equipment, such as machinery , material handling and controls, is a significant consideration in the design process.

To build a manufacturing system that is coherently flexible to meeting the changing production needs, the modular adaptability of the technological building blocks is of great importance. Manufacturing equipment modules, workpiece management modeules, tool management modules, auxiliary equipment modules and software tool modules should be able to be individually selected for varying degree of sophistication and adaptability, yet possess the ability to interrelate with each other at a level beyond individual capabilities (Figure 3.2) [198].

The type and number of each of these modules as well as their physical arrangement, is determined by the manufacturer's objectives, method of production, size of parts and a number of other factors. In the following sections six design choices for the manufacturing system configuration are described and discussed on the basis of the established flexible manufacturing concepts.

3.3 The Unmanned Station (UMS)

A UMS is a machine tool with tool and workpiece magazines and has the capability to operate without direct manning for relatively long periods and to machine a variety of components in this mode of operation (Figure 3.3) [198]. The manufacturing equipment employed in an UMS can basically take two forms: CNC lathes or machining centres. In a typical CNC lathe, there is a slant-bend with one or two turrets carrying tools for turning outer peripheries, bores or for boring and drills (Figure 3.4). There are two common types of machining centres, those with vertical and those with horizontal spindles. The horizontal type is used most though, owing to their greater flexibility, and because swarf falls naturally away from the workpiece.

To check tool wear and warn of tool breakage, some form of probe is required to support the unmanned station [275] [197]. The probe is inserted in the chuck at the beginning of the machining sequence, and is brought to position the workpiece or to detect tool wear and whether a tool has been broken.

If the station is required to carry out unmanned operation for relatively long periods and machining a variety of components in this mode, the concept of detachable tooling has to be introduced. The basic idea is that a magazine of tools should be mounted near the machine, and that some form of machanism should change the tools as needed [146]. Figure 3.5 shows the block tooling concept employed by Sandvik [125].

To achieve high level of unmanned operation in an UMS, the machine is equipped with mechanical handling devices which can take the form of either a multi-station pallet carrier or an automatic pallet changer (APC)[197]. Once multiple pallet carriers are used, a device that identifies which workpiece is coming on to the table is required. Where the machining cycle time is fairly short, and multiple pallets would not provide sufficient stock to justify their use, twin-pallet APCs are adopted.

Since lathes generally operate with short cycles, robot handling and a buffer store at the machine are needed. An alternative is the CPC system developed by SMT Machine Tools in Sweden [197] (Figure 3.6).

Control systems for this type of installation include multi- part programme storage, tool management software, support for contact probes. In addition, conditional programming is essential to allow complex and high capacity fixtures.

A typical example of this type of workstation is reported in [23]. The unmanned system consists of a CNC lathe combined with an automatic chuck jaw changer, and a robot to allow for turning out mid-size parts in large variety, small and medium volume production. Since the CNC lathe can accommodate 30 turning tools and secondary machining tools, parts requiring secondary processes can be machined through all processes in two chuckings. This results in a significant enhancement of system productivity.

An example of the UMSs for prismatic parts is shown in Figure 3.8. The interesting feature of the station is that the tool kitting concept is used for the management of tool flows. The pallet is so designed that it can accommodate four workpieces, three tools and a memory card.

Unmanned stations are not only installed for milling, drilling or 'lathe' machining, but also for gear cutting. The aim is to increase productivity and flexibility. The example shown in Figure 3.7 consists of a hobbing machine with a six-axis multiple processor control; a six-axis loading gantry for hob changing, fixtures and workpieces; storage areas for changable grippers; tool and workpiece clamping devices and a magazine for the accommodation of pallets for disc and shaft-shaped raw and finished parts. The station is designed for the lot size ranging from one to two hundred [103].

3.4 Flexible Manufacturing Cells (FMC)

An FMC is made up of a small number of CNC machine tools combined through automatic and unidirected work handling. It is considered as a system concept for medium volume manufacture with very restricted variety.

Figure 3.9 and Figure 3.10 show the structure of this type of system for cylindrical and prismatic components respectively. In the former case, the cell can be based on a group of existing individual machine tools supported by local controolers and an inspection station. An industrial robot is used for workpiece transfer. The cell layout is dictated by the robot specification. This type of cell is justified on the basis of zero variety but ease of configuration and low capital cost. In the latter case, the cell construction is superficially similar to a transfer line. Its status as a manufacturing system is determined by the system's software specification. The cell is highly mechanised with uni-directional workpiece flow and can offer some degree of workpiece variety [198].

Due to the variation of manufacturing requirements, these type of cells can also take the following forms [15]:

- The automated cell is applicable to high-volume production of a small, well-defined and homogeneous family of parts (Figure 3.11) (Figure 3.12) [33] [135] [159] [20]. Robotics or specialized material handling links a small number of flexible CNC machine tools (Figure 3.13) (Figure 3.14) [78] [125]. The cell normally has a fixed process, and parts flow sequentially between operations. Its high productivity is achieved through the application of the specialized work handling device, power clamping of parts, special tools and other forms of automation. Generally this type of installation is economically viable in its own right, and does not have any systems implication [15].

- The FMS cell satisfies a manufacturing requirement for medium volume output. It can be either a self-contained manufacturing unit, or a step in the automation of a manufacturing area since several cells may be linked to form a large scale multi-cell system [156]. The distinguishing characteristic of this type of cell is the automated flow of raw material to the cell, total machining of the components across the machines within the cell, and finally the removal of the finished parts.

An example of the automated cell for cylindrical parts is the Okuma turning cell [125]. It consists of two CNC lathes supported by an industrial robot. The cell is set up to machine flanges, housings and shafts. The minimum batch size is 20 pieces and the typical cycle time is about 4 minutes. The cell can operate unmanned for up to two shifts provided pallets can be supplied and removed automatically, one man being required on the day shift. It is claimed that the metal cutting time during the three shifts has been increased to 80% from 50-55% where two machines are used independently and loading/unloading is facilitated manually and the cell's productivity/man is 4.4-4.8 times that of manually-operated two machines.

Examples of the FMC for prismatic parts are shown in Figure 3.15, Figure 3.16 and Figure 3.17 respectively [158] [120] [12]. The FMC Pegard has installed for Caterpillar incorporates the Pegy tool changing robot. The two machining centres in the cell are both partnered by a Pegy, putting 160 tools at the disposal of each. Though essentially horizontal machines, the two machining centres are equipped with right angle heads so that they can machine all five faces of the part without relocation. A rail-guided pallet shuttle transfers parts between machines. Similar to the Pegard cell, the Lheon FMC also consists of two machining centres, each equipped with an ATC that hold sixty tools in order to meet the requirements for machining a great variety of workpieces.

In contrast to the above mentioned cells, the Werner Kolb FMC has a particularly sophisticated tooling system. It includes a numerically controlled double-portal handling system, which is responsible for the transfer of tools between the machine tool magazines and the central tool storage. A double gripper capable of pivoting on two axes is used to help speed-up the tool change cycle. A similar tooling solution is used by the Howden cell (Figure 3.13) [78].

An example of the FMS cells can be found in [206]. It incorporates a machining centre and a turning centre, supported by two gantry robots. An AGV system delivers parts and tools to the machining cell and to the robotic deburring station and washing station. The tool preparation area is also equipped with a gantry robot.

3.5 Flexible Manufacturing Systems (FMS)

An FMS is normally made up of a group of CNC machine tools which are unordered in a process-independent layout, and therefore can offer varying degrees of flexibility in processes and routing (Figure 3.18). In general, auxiliary equipment like deburring, washing and inspection machines are also provided. Automatic transport and loading/unloading of workpieces are necessary features and in considerable cases bi-directional work flow is of evidence. Recent development in manufacturing software has demonstrated the frequent inclusion of automatic tool management systems in most of the installations [298]. In addition, this type of system accommodates a comprehensive computer control function and requires NC data control, scheduling and production control. It is possible that in some particular cases dynamic variations be dealt with by the system's software [40].

It has widely been recognised that an FMS installation can be built with significant capability to respond to short-range changes in manufacturing requirements, the status of manufacturing equipment or the system extension. Tremendous flexibility in workpiece variety, batch size and batch distribution is capable of realisation [40].

There is evidence of wide application of FMS technology all over the world [98]. The example depicted in Figure 3.19 is a system to machine pump bodies for large automotive engines and industrial units [126]. This installation is primarily justified on the attainment of high flexibility of components variety and batch sizes, though only five machining centres are utilised. During the three shifts of production allowed, merely two men are needed in just one shift. As a result, manning level is greatly reduced. Work flow is accomplished by a rail guided AGV, while no automatic tool transport is applied since adequate tools can be provided manually for the ATC of each machine to carry out all required operations at the beginning of the manned shift.

The Normalair-Garrett system (Figure 3.20) has the distinction of being the first FMS built and operated in Britain [298]. It has been designed to produce a product consisting of a kit of parts. The staff support the manufacturing operations, carry out inspection of the components produced, and assemble the final product. The interesting technical feature of the system is that it is also the first system in the UK to use a secondary chain magazine to support the primary disk magazine of the machining centres.

As shown in Figure 3.21, the FMS installation at Strathclyde consists of five horizontal machining centres and one special contour boring and facing head machining centre [66]. The components are fed to each of the machines by two flow-through pallet stands linking the machine tables with the main run of the work transportation track. In contrast, the SCAMP system (Figure 3.22) uses a roller conveyor system to transfer components between buffer stores and machines [16]. Each machine is served by a industrial robot for the loading/unloading of components. A vision system is also employed for checking the asymmetry of particular components prior to those components being loaded into individual machines.

For large-scale FMSs, automatic warehouses can be included. Figure 3.23 and Figure 3.24 show two examples [149] [155a] [127]. Although AGVs are the most common material handling devices, Figure 3.25 illustrates the use of a computer controlled gantry crane for the handling of very large and heavy workpieces [157]. In addition to the standard tool magazine of 80-tool capacity, each machine in the WMW system also has a supplementary tool magazine with 20 tools to provide additional capacity. Designed for the production of printing frames as one-offs or in small batches, this system is expected to achieve the following ecomonic benifits: 58% reduction in number of machine tools, 76% reduction in labour force, and 70% reduction in throughput time.

A more sophisticated system installed by Renault Machine Outils is shown in Figure 3.26. It is justified on the achievement of the manufacturing flexibility, the product flexibility and the inventory control flexibility [96]. Another FMS built in France is the Citroen FMS for prototype automobile parts (Figure 3.27). An interesting feature of the system is its work and tool flow control strategies as shown in Figure 3.28 and Figure 3.29 respectively [214] [213]. The AGVs are used for the transfer of both workpieces and tools. A kitting strategy is employed for the management of tools, i.e. all tools have to be changed between operations for two successive parts [18].

An installation which demonstrates the potential for flexible manufacturing at one extreme of performance, i.e. large volume, short cycle time, is introduced by Cross International (Figure 3.30). It can machine 80 variants in four families and eventually may be able to process 140 different parts. The line will produce parts at the rate of 780/hour. The cycle time for each component is 9.2 seconds [210] [147]. This is in contrast to the classic 'batch of one' FMS based on machining centres which do more complex work but require more sophisticated control.

The AIMS project at Rolls Royce, Derby, has been designed and evolved by the company over a number of years. The layout is shown in Figure 3.31. Only 12 of the 26 machine tools are CNC. The achievements are quoted for all performance: lead time reduced by 60%; machining operations reduced from 21 to 5; inventory reduced by 24%; number of machine tools reduced from 57 to 26; types of machine tools reduced from 17 to 8; and scrap reduced by 39%.

3.6 Multi-Cell Flexible Manufacturing

Multi-cell flexible manufacturing facilities are a large- scale automated processing network, in which a number of cells are linked functionally through a common material handling system. Usually work flow within cells is possible, its features being similar to those of FMCs. In most cases, varying functions such as fabrication, machining and assembly are performed by different cells separately. Auxiliary equipments like inspection machine and washing machine are also typically employed to form individual cells.

The control of this type of manufacturing configuration is of a hierarchical nature, with cell processors being coordinated by a host computer. Scheduling and control of production can play a key role in the efficient and effective operation of the system. Figure 3.32 illustrates a generic structure of these types of systems.

A typical example of multi-cell systems for machining and assembly is depicted in Figure 3.33 [103]. The workpiece spectrum being produced is comprised of a collection of various sized backflow prevention valves. The chip formming machining of the caseparts is carried out on a CNC lathe and a CNC machining centre with changable pallets, a measuring station, and a washing station. A six-axis industrial robot performs workpiece handling with the assistance of an automatic gripper-changing system. In the assembly area, a five-axis gantry robot equipped with a gripper and tool changing system does the work for all handling and assembly tasks. Work flow between the two cells and the system storage is performed by inducting guided transportation vehicles.

Figure 3.34 shows the control structure for a system formed by several FMCs [129]. In such a multi-cell system, tasks are divided among the individual cell computers and the coordination computer. Machine-related functions, such as NC program supply and the machine tool programs, any material flow control within the cell and the associated data storage facilities remain at the cell level. Execution of the higher functions, such as job scheduling, tool requirement management, material flow control and the dialogs with the setting-up and clamping locations, etc. are the responsibility of the coordination computer.

A similar control hierarchy can be seen in Figure 3.35. The three cell computers are all under the control of the plant host computer. The flexible machining cell is designed to produce more than 60 parts, mostly as one-offs, whereas the automated assembly cell is intended to handle nine sub-assemblies in batches of one. For a detailed layout of the machining cell, see Figure 3.36 [206].

Multi-cell systems can also be used just for machining. An example is the multi-cell facility consisting of seven grinding cells for blade manufacture (Figure 3.37). Each cell is made up of two grinding machines served by a industrial robot. The cell also has its own automatic blade cleaning and inspection equipment. Major advantages this approach offers are reduced cost, shorter lead times, improved product quality, consistent levels of output, simplified shop control, lower manning levels and the ability to react to changes in demand or specification [17].

A recent multi-cell system shown in Figure 3.38 is installed to produce shafts and wheel assemblies for turbochargers [156] [290]. The system comprises seven cells, each accommodating two or three machines and a five-axis gantry robot (Figure 3.39). The gantry robot can also be used to change tools, fixture parts and change workholding jaws in some cells. Work transfer between cells is accomplished by three wire-guided AGVs. Each cell has its own local area controller which co-ordinates all the activities in the cell as shown in Figure 3.40. Communications between cell controllers is also possible. This system is justified to produce 50 part numbers in small batches or even one-offs.

The multi-cell system approach can also be used purely for assembly. The installation shown in Figure 3.41 is designed for PCB production [82] [95]. Figure 3.42 shows the materials flow, control and communications in the multi-cell system.

3.7 Large-Scale DNC Systems

A large-scale DNC system is the latest development in the field of batch manufacturing. It is a highly integrated manufacturing installation comprising various work stations all under central computer control. The work stations can be any forms of manufacturing units, e.g. individual machine tools, DNC cells, FMCs, or even FMSs. This approach provides a solution for the automation of a total manufacturing area (Figure 3.43).

The advantages of such a large-scale DNC system are obvious (Figure 3.44). Firstly each of the unit is functionally self- contained, and this makes the control of each of the units and the total manufacturing area simple. Secondly, the manufacturing units can be put into operation at different times. In the third place, these units have a standardized interface to the coordination computer. Fourthly the units can be supplied by different machine manufacturers. And lastly, since the incorporation of autonomous units into the system can be staggered, furthur extension of the system is very convenient. For a generalised structure of large-scale DNC systems, see Figure 3.45.

A large-scale DNC system made up of two FMCs is installed at Takisawa Machine Tool [21]. The first FMC (line A) consists of three horizontal machining centres, a track type unmanned trailer and fifty pallet stands, and is designed for machining 80 types of large-sized components. The second FMC (line B) is installed for the production of 20 types of medium and small-sized parts, and consists of 2 horizontal machining centres and a trackless type unmanned trailer (Figure 3.46). As can be seen in Figure 3.47, the work flow within the system is of a parallel nature for the two cells.

Control of the system is performed at three levels (Figure 3.48). The host computer handles synthetic data processing, such as engineering and business as well as management control. The process computer, on the other hand, manages a wide range of processing from the machining scheduling, control data, controllers for the two FMCs and outgoing parts, to the output of controlling conditions. The controllers for the two FMCs control the transfer, incomming and outgoing parts to and from the warehouse, and the automatic operations of machining centres.

The large-scale DNC systems manufactured by Yamazaki are very sophisticated in terms of the manufacturing units contained. Figure 3.49, Figure 3.50 and Figure 3.51 show the three systems installed at Oguchi, Minokamo and Worcester respectively. The installation at Oguchi consists of two FMS lines called A and B [163]. The machine tools used in each line include eight and ten units respectively. Each line employs a track type AGV for the transfer of workpieces between machine tools. The line A has a machining capacity of 800 workpieces per month of 23 parts for headstocks of NC lathes and machining centres, whereas the line B machines 51 types of parts, totalling 600 workpieces on a monthly basis. This system represents a major early investment in system software and is highly automated, with both work flow and tool management being computer controlled.

The Minokamo system, introduced for CNC lathe manufacture, has a main manufacturing area which consists of five FMS lines [22] [124]. The plant layout makes it possible to coordinate ancillary manufacture, FMS, unit assembly and machine tool assembly by means of AGVs and a digitally controlled warehouse. It is well justified on the reduction of equipment and labour requirement, and lead times.

The Yamazaki factory at Worcester can be regarded as a development of the installation at Minokamo in a number of areas. It is a more compact plant than the Minokamo plant and the factory level control system is more developed. Flexible machining is concentrated in three lines, i.e. small prismatic parts line, large prismatic parts line and rotational parts line, using seven, three and three machine tools respectively [28] [30]. The turning line is of particular interest as it uses turning systems with live tooling in contrast to the Minokamo plant which employed CNC lathes and small vertical machining centres. The rotational part palletising and work handling is also more sophisticated.

An equally influential impact has been made by the construction of a family of factories to produce electrical servomotors, industrial robots and machine tools in small highly integrated combinations of manufacturing and assembly areas. Figure 3.52 and Figure 3.53 depict the two large-scale DNC systems installed by Fanuc for the production of motors, and robots and CNCs respectively [19] [11]. These installations have given the lead on the use of flexible manufacturing technology for medium batch manufacture with significant variety. The use of a major DNC type network approach to overall control (Figure 3.54) is a powerful feature [29].

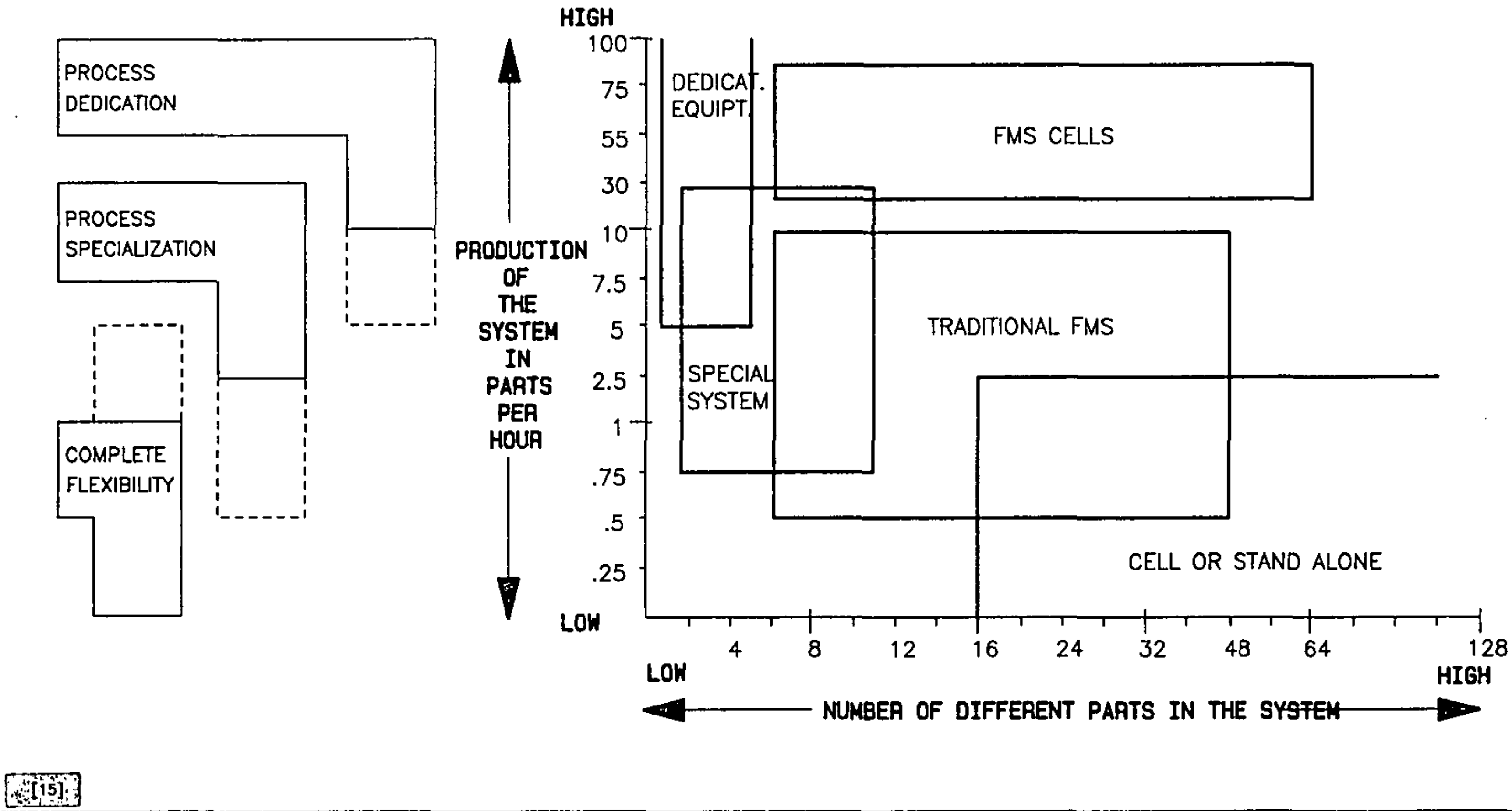


Figure 3.1

MANUFACTURING CONCEPTS
Productivity vs. Flexibility

LUT - FMS
Research Group

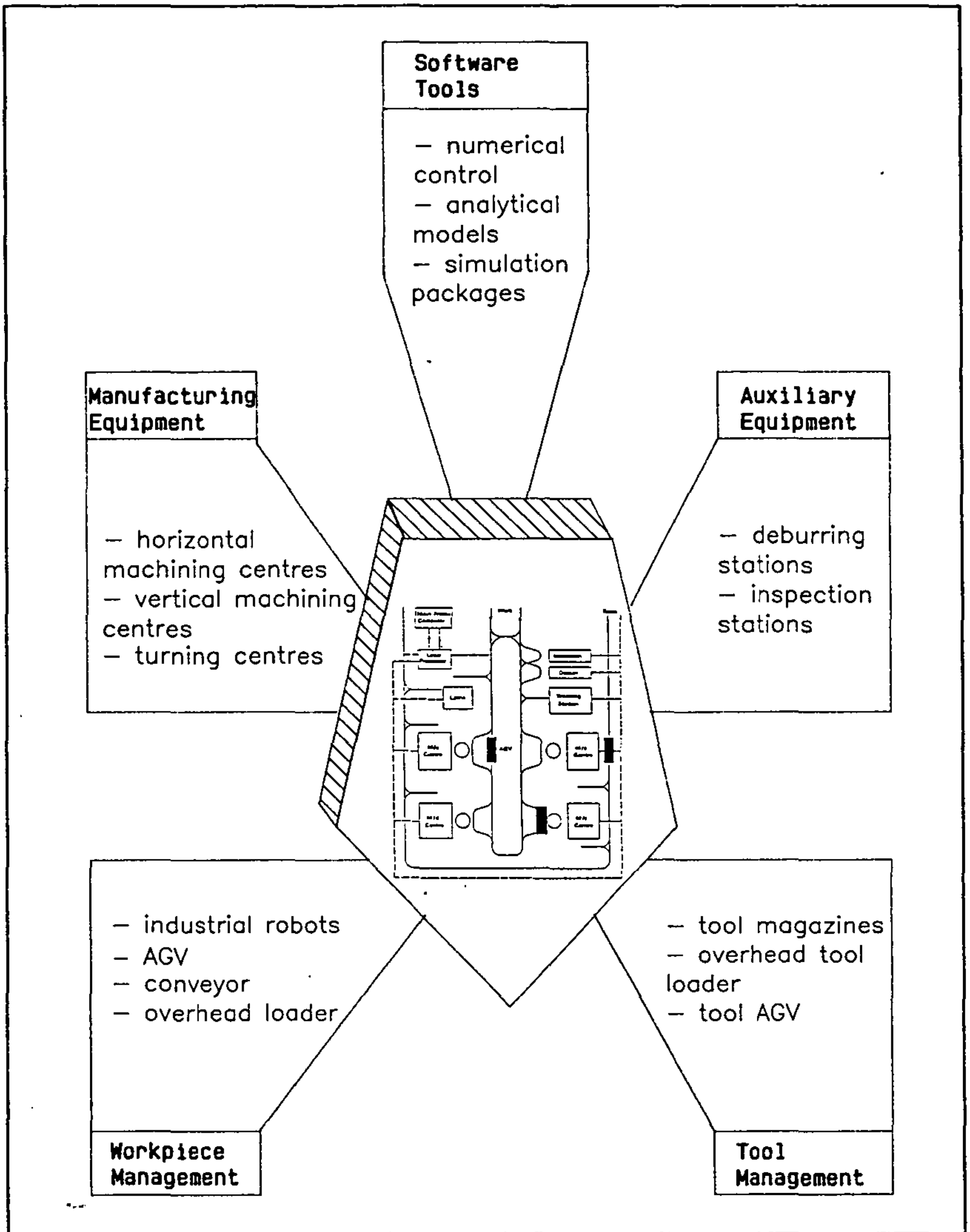


Figure 3.2

Building Blocks for Manufacturing Systems

LUT-FMS Research Group

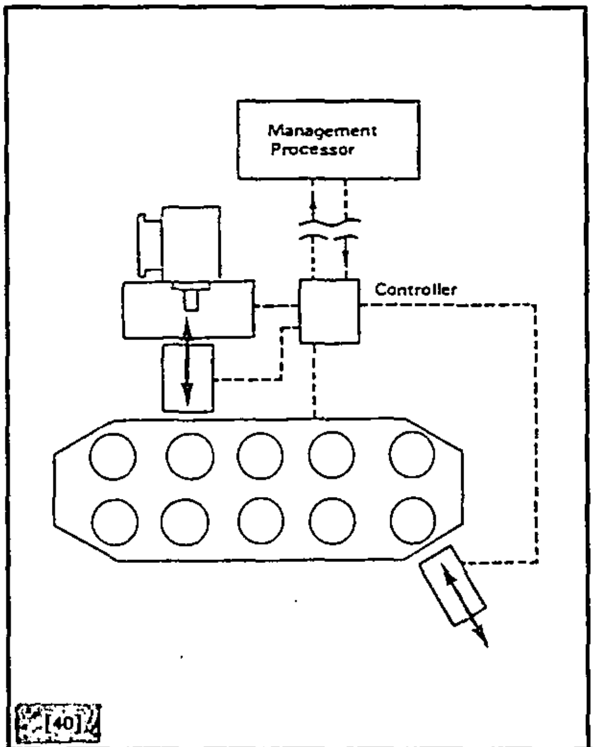


Figure 3.3	*Structure of the Unmanned Station*	LUT-FMS **Research Group**

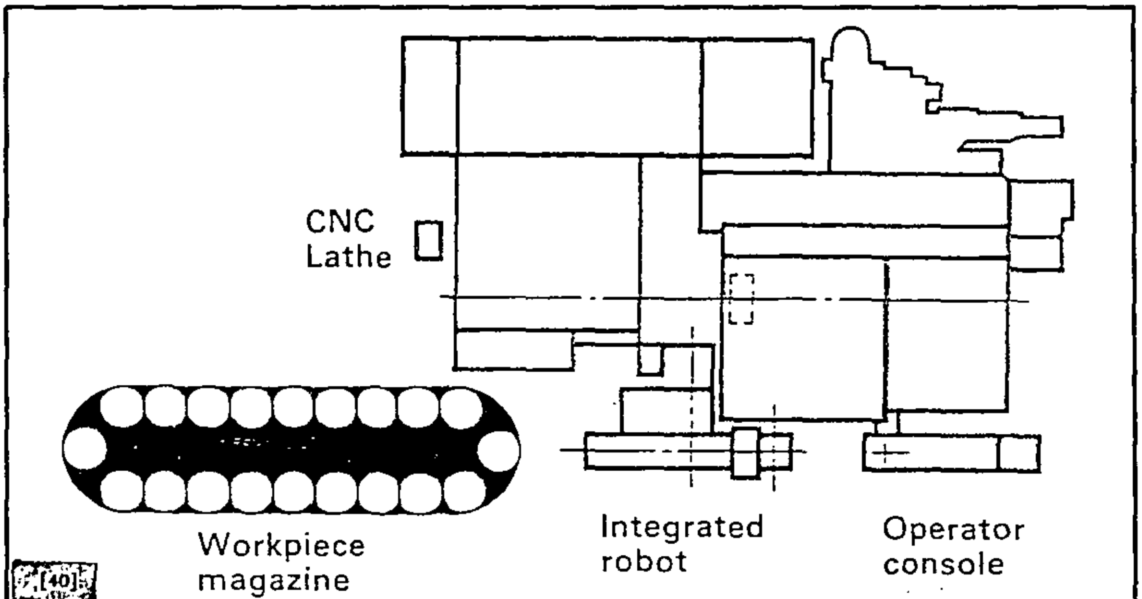


Figure 3.4 **Lucas Bryce**	*An Unmanned Station for Turned Parts*	LUT - FMS **Research Group**

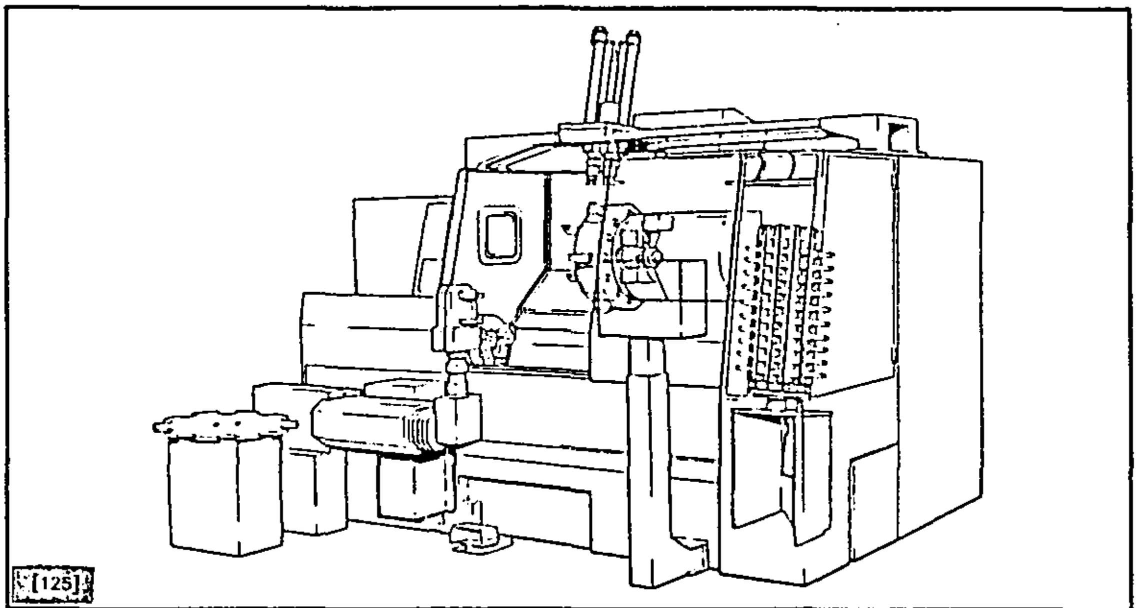


Figure 3.5 **Sandvik**	*An Unmanned Station with Block Tooling*	LUT - FMS **Research Group**

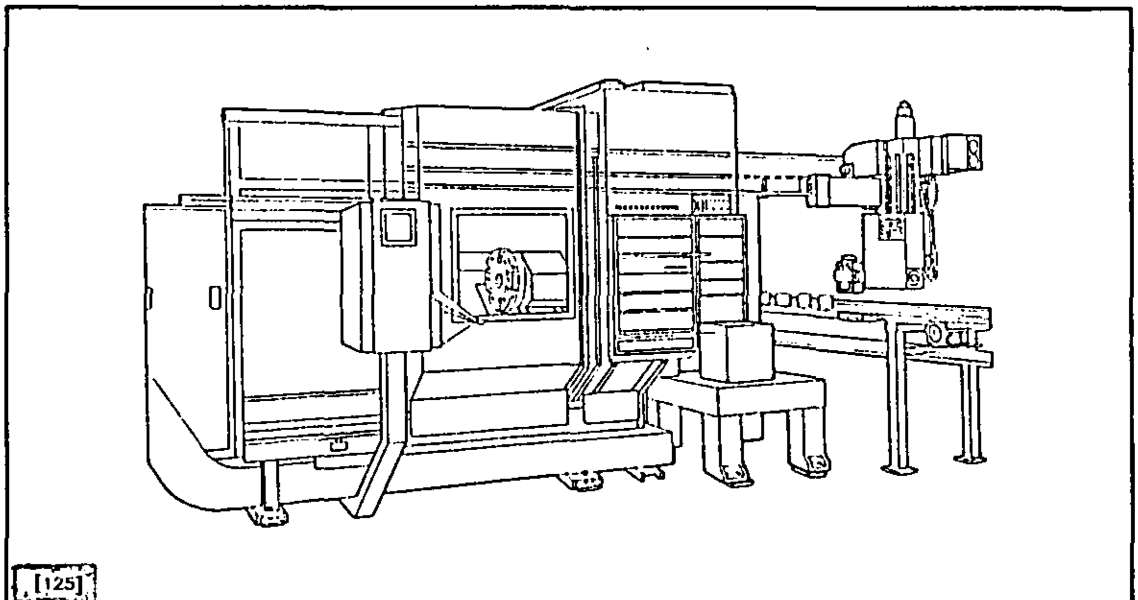


Figure 3.6 **SMT Machine**	*An Unmanned Station with the CPC Handling*	LUT - FMS **Research Group**

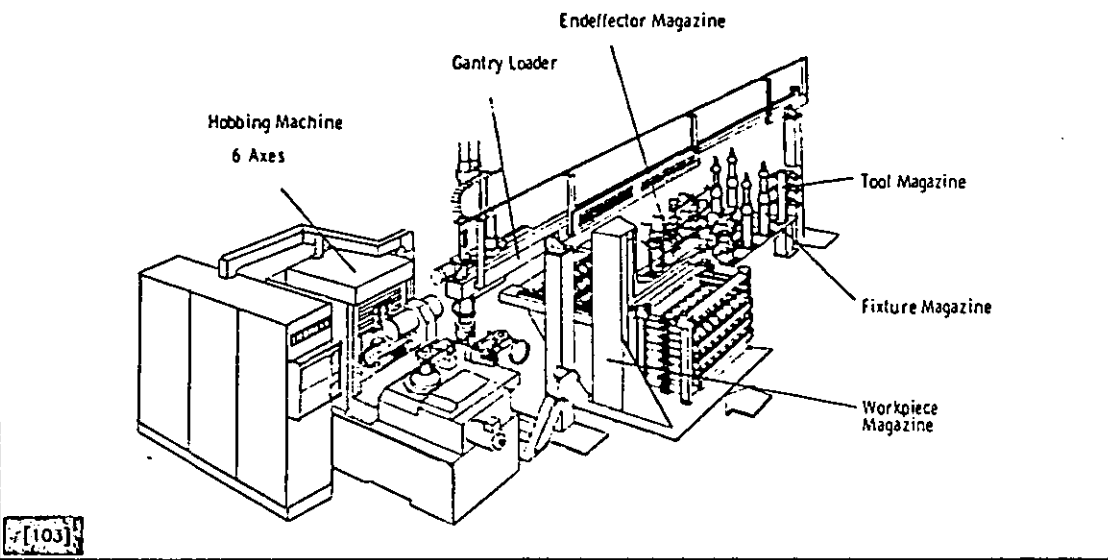


Figure 3.7

An Unmanned Station for Gear Cutting

LUT - FMS
Research Group

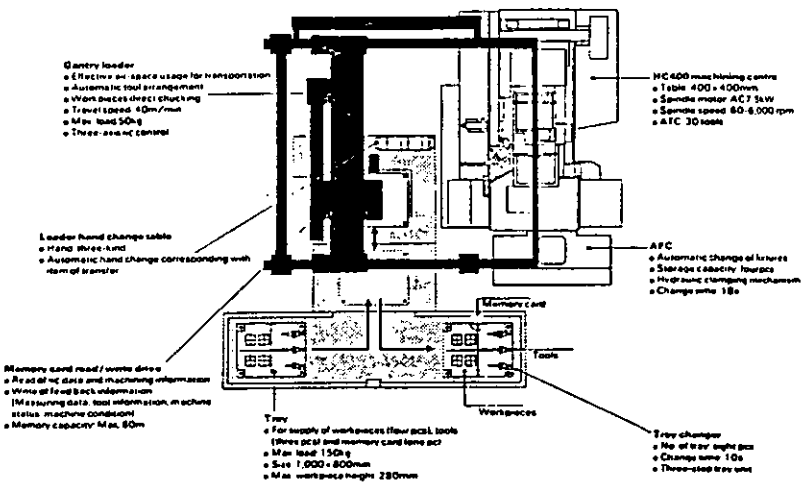


Figure 3.8
Hitachi

An Unmanned Station for Prismatic Parts

LUT - FMS
Research Group

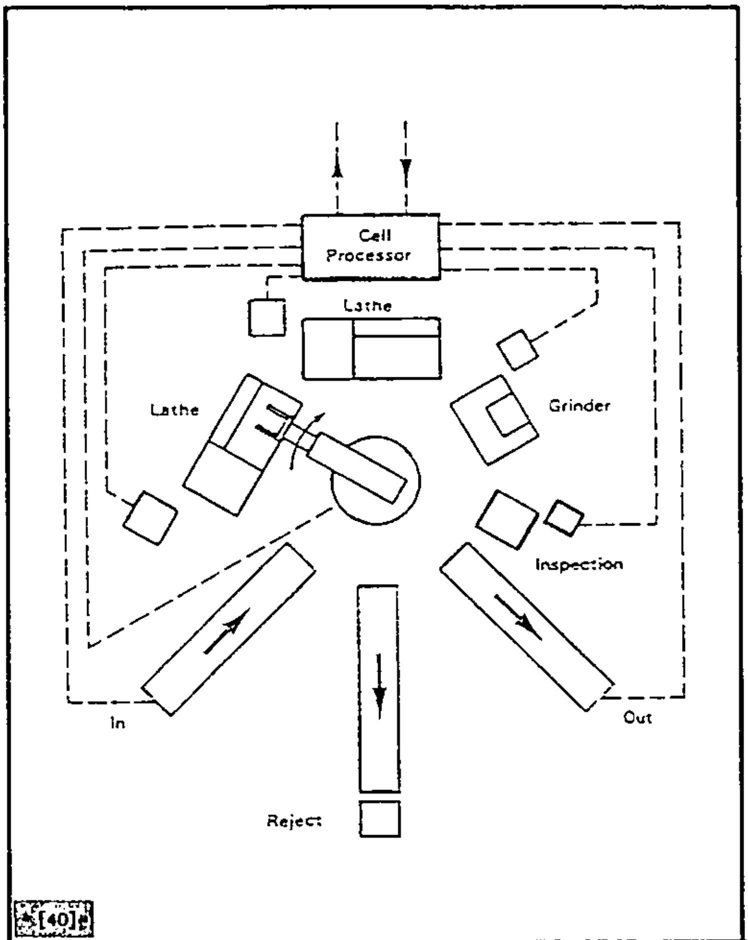


Figure 3.9	*Structure of the FMC for Cylindrical Parts*	LUT-FMS **Research Group**

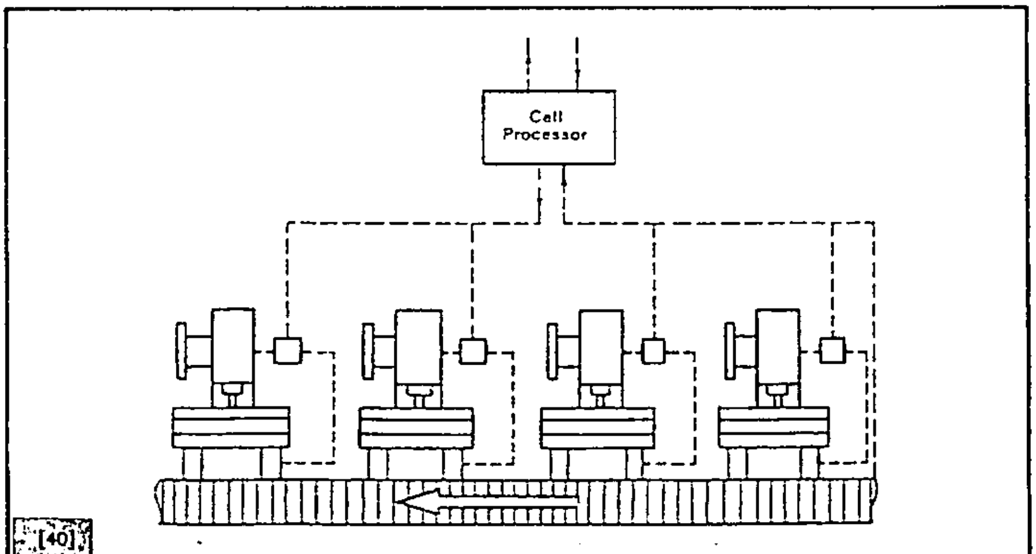


Figure 3.10	*Structure of the FMC for Prismatic Parts*	LUT - FMS **Research Group**

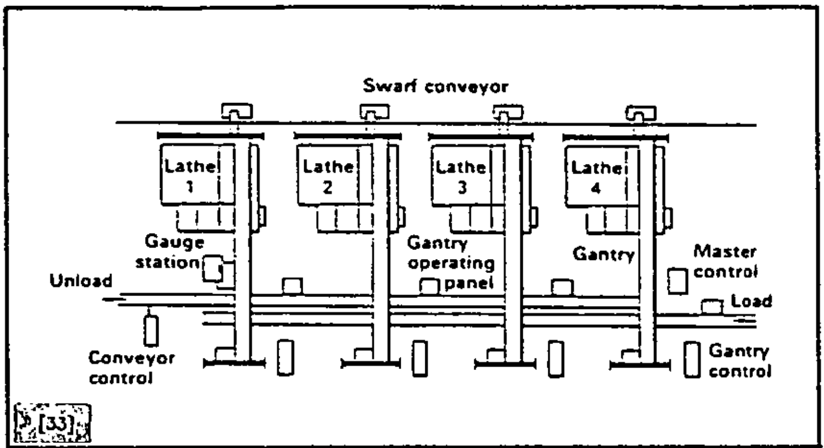


Figure 3.11 **Perkins** | *An FMC for Weight Components* | **LUT - FMS Research Group**

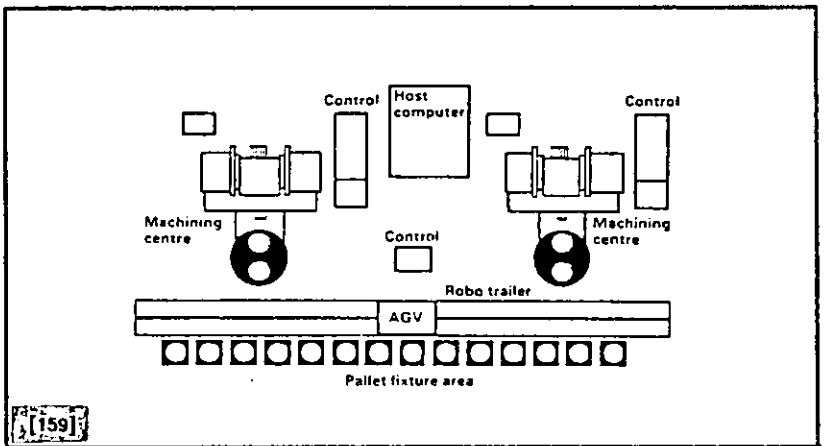


Figure 3.12 **KTM** | *An FMC for Prismatic Parts* | **LUT - FMS Research Group**

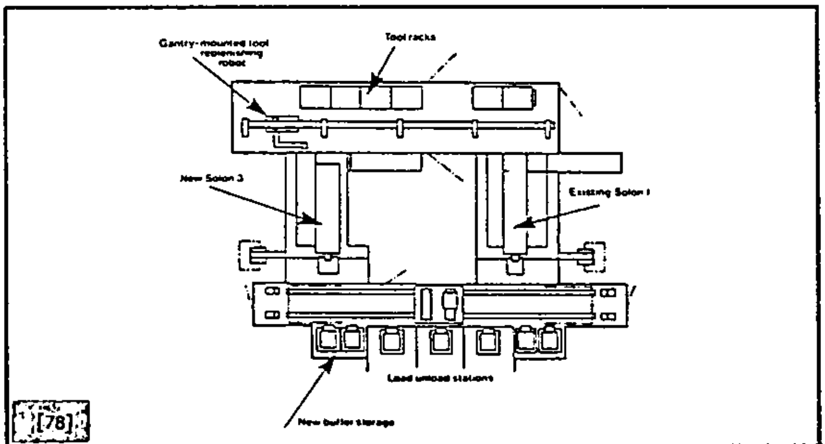


Figure 3.13 **Howden** | *An FMC for Compressors & Blowers* | **LUT - FMS Research Group**

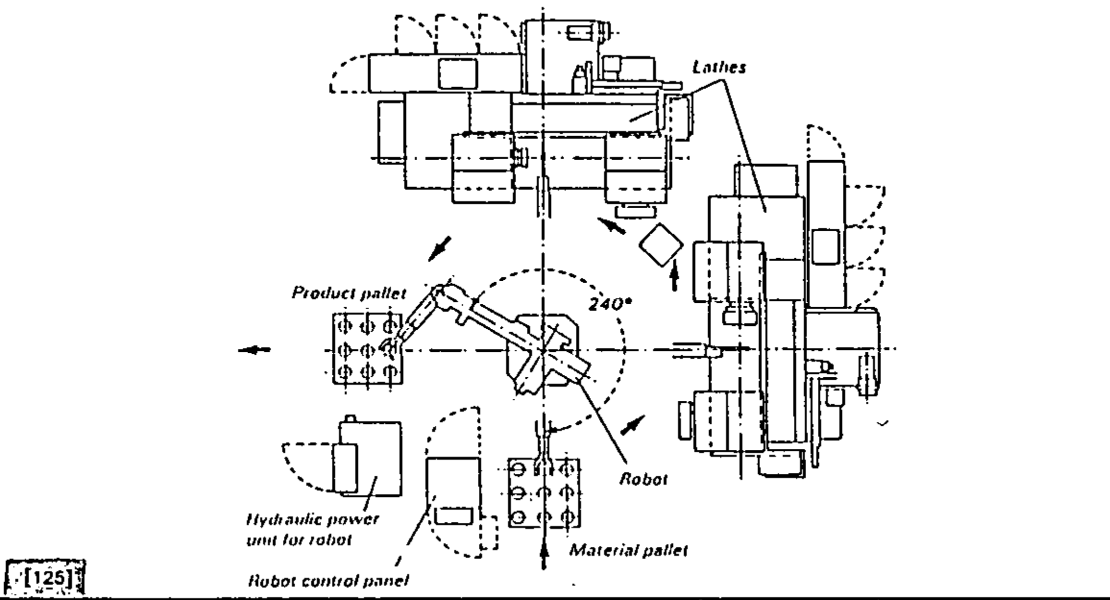


Figure 3.14 **Okuma**	*An FMC for Flanges & Shafts*	LUT - FMS **Research Group**

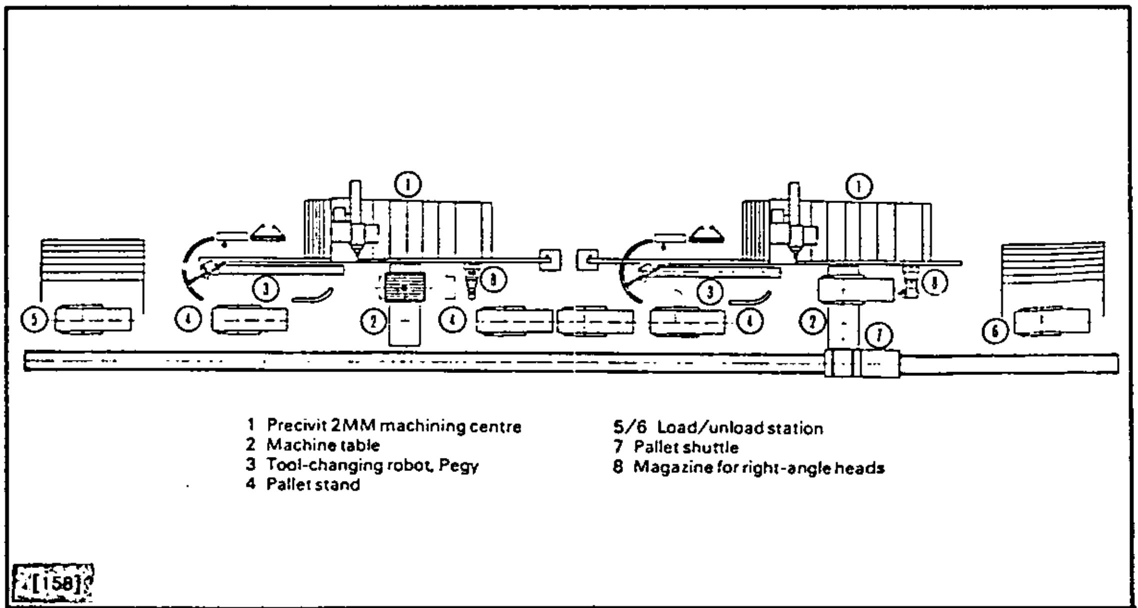


Figure 3.15 **Pegard**	*An FMC for Prismatic Parts*	LUT - FMS **Research Group**

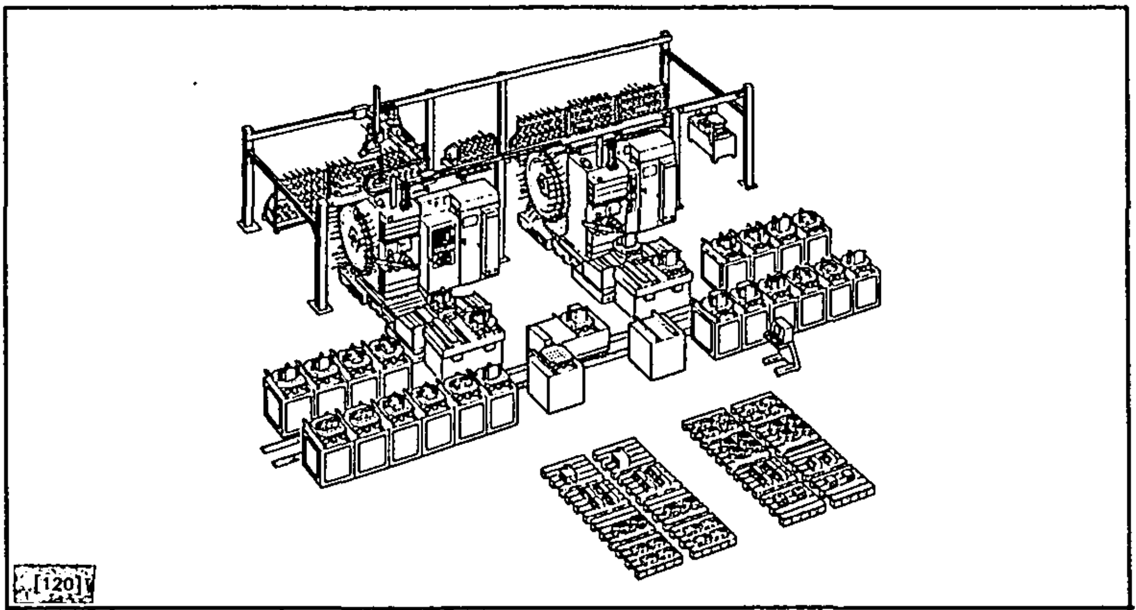


Figure 3.16 **Werner**	*A Double Workstation FMC*	LUT - FMS **Research Group**

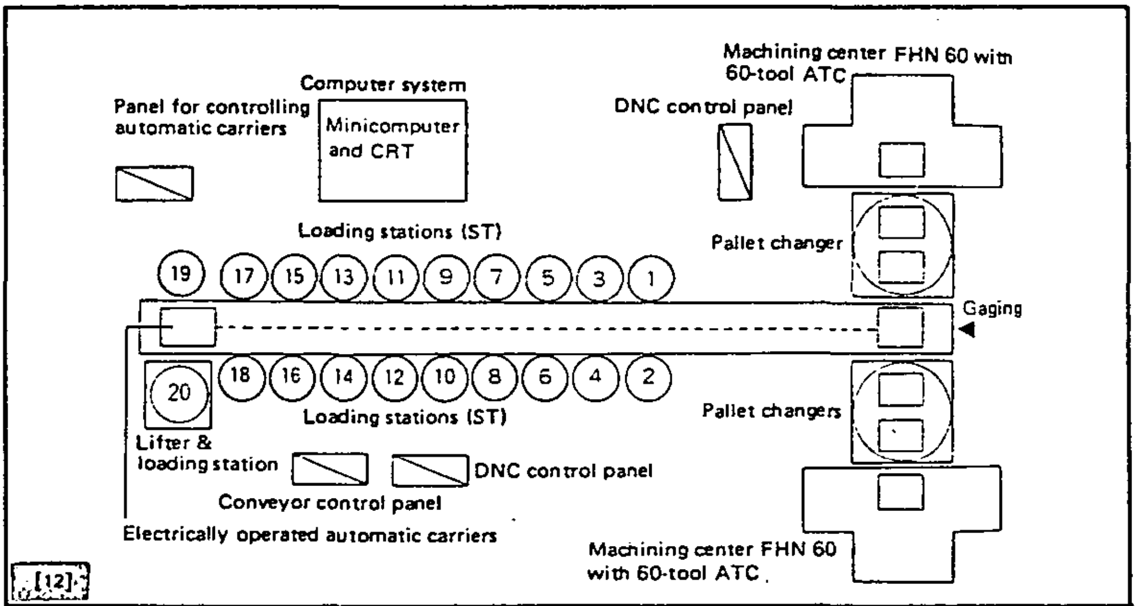


Figure 3.17 **Lheon**	*An FMC for Gear Boxes & Frames*	LUT - FMS **Research Group**

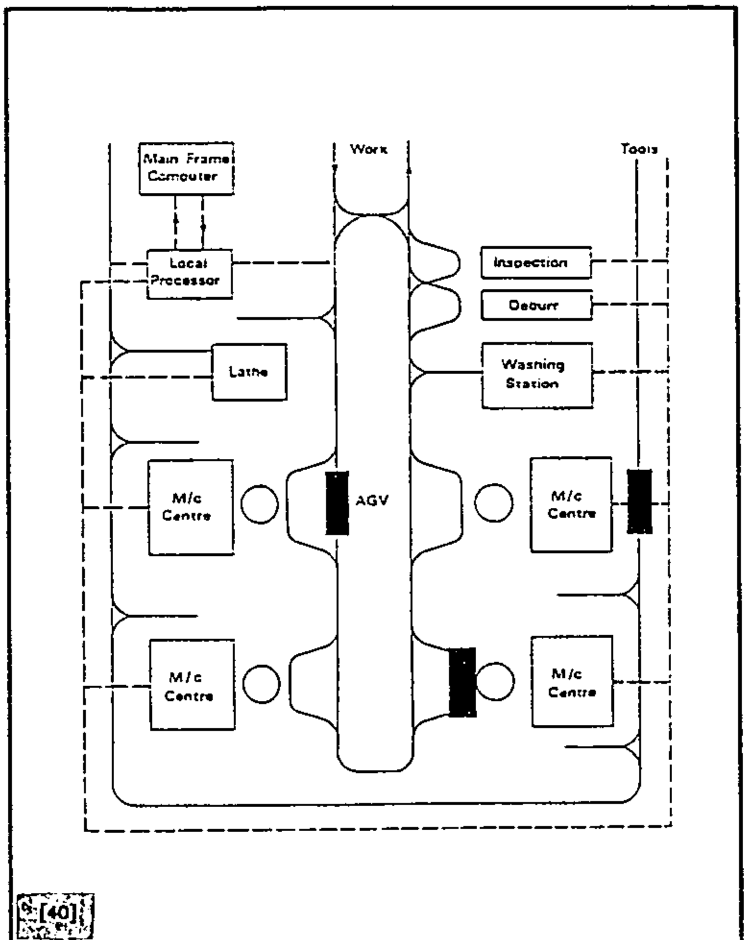


Figure 3.18	*Structure of Flexible Manufacturing Systems*	LUT-FMS Research Group

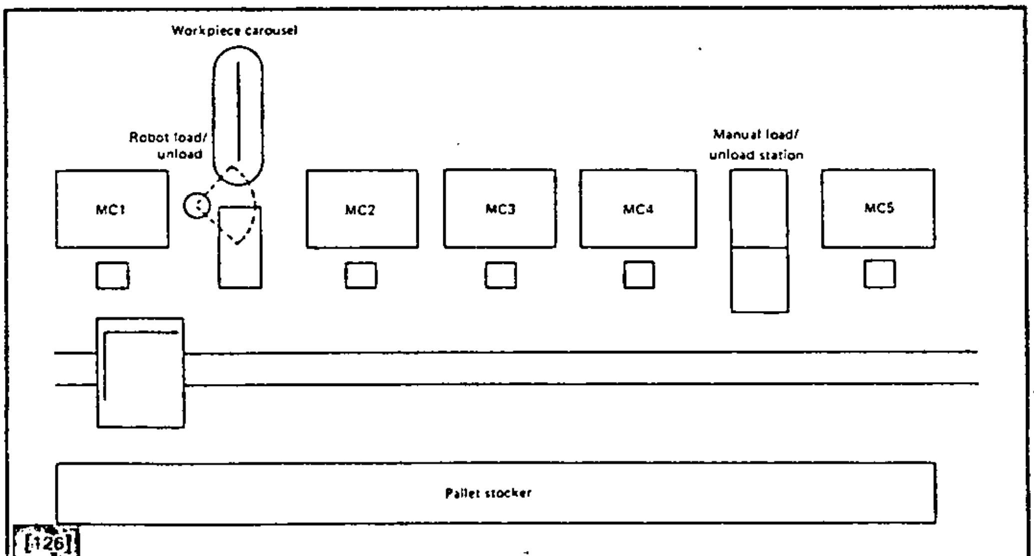


Figure 3.19 **Diesel Kiki**	*An FMS for Pump Bodies*	LUT - FMS **Research Group**

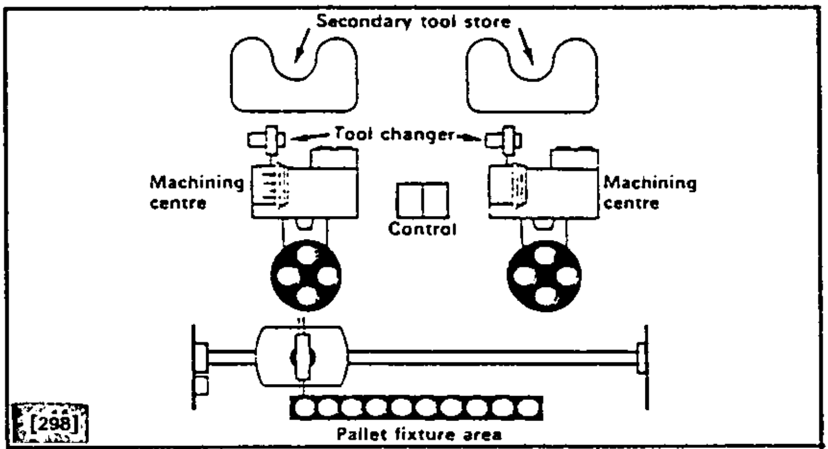


Figure 3.20
NGL

An FMS with a Secondary Tool Magazine

LUT - FMS
Research Group

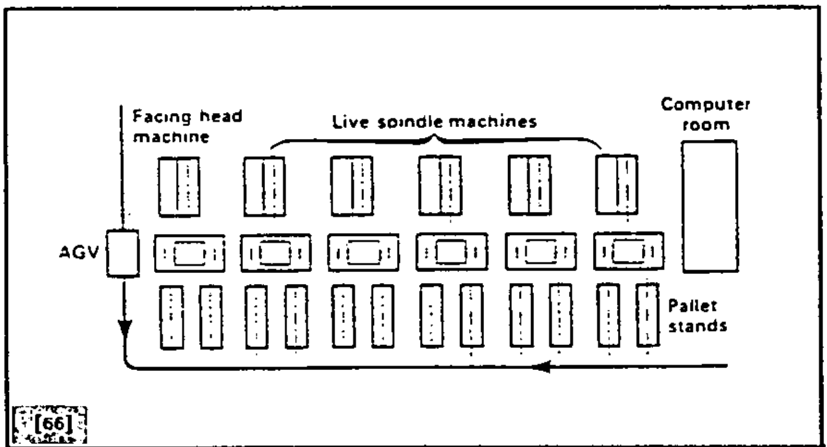


Figure 3.21
Anderson Strathclyde

An FMS for Gear Cases and Boom Castings

LUT - FMS
Research Group

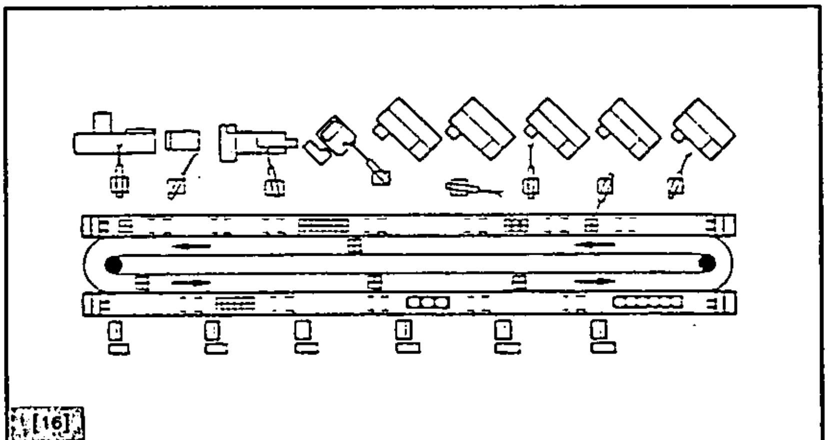
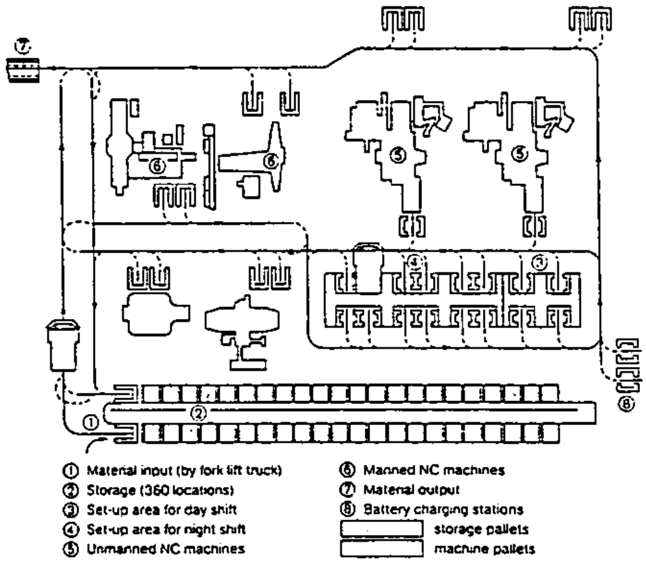


Figure 3.22
600 Group

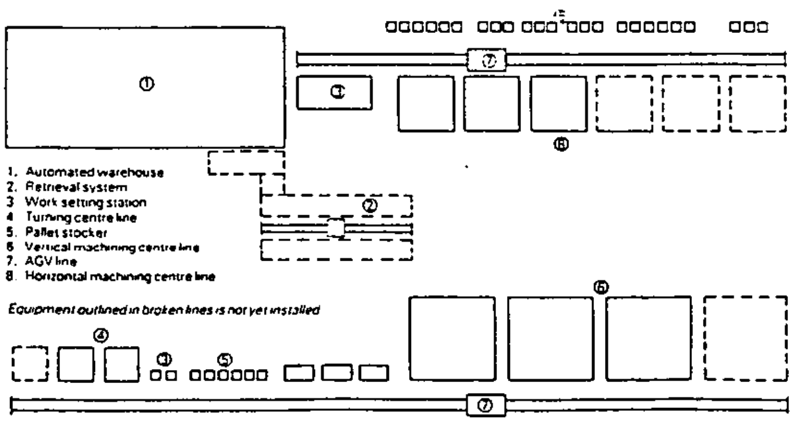
A Highly Flexible FMS for Turned Parts

LUT - FMS
Research Group



[149]

Figure 3.23 **Westfalia Separator**	*An FMS for Pump Housings*	**LUT-FMS Research Group**



[127]

Figure 3.24 **NKK**	*An FMS for Very Large Workpieces*	**LUT - FMS Research Group**

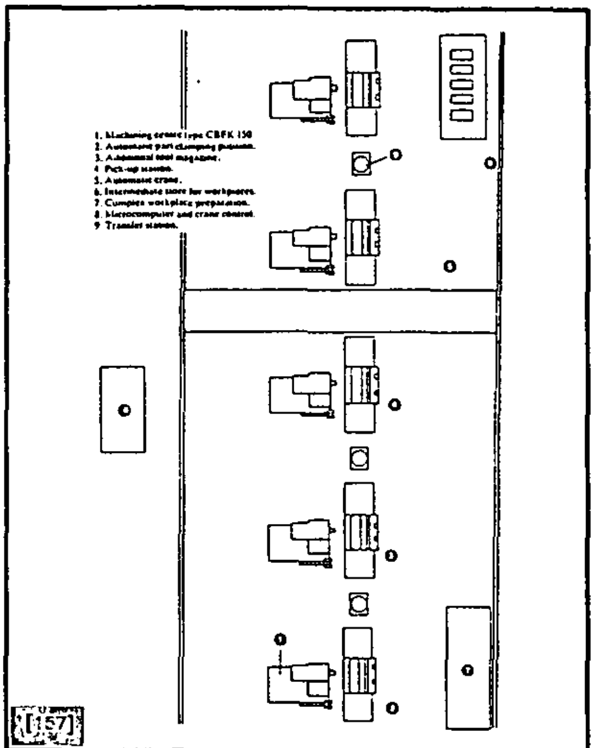


Figure 3.25 **WMW**	*An FMS with a Computer Controlled Gantry Crane*	**LUT-FMS Research Group**

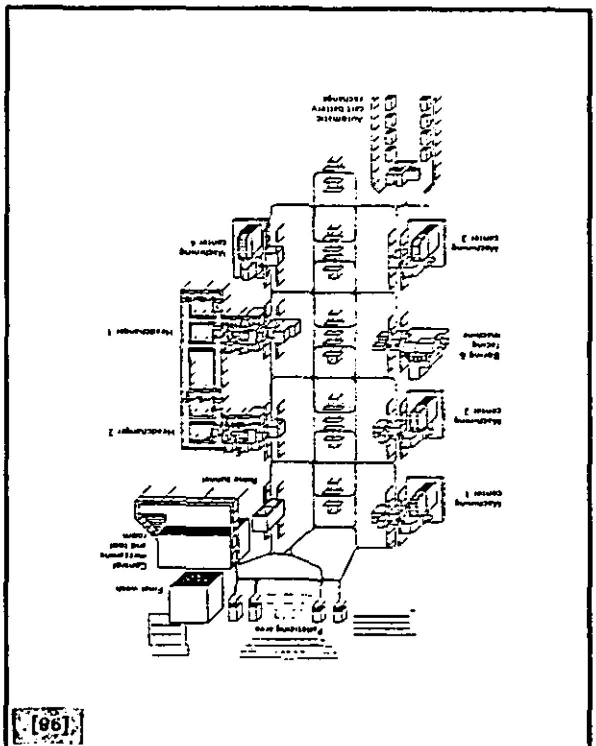


Figure 3.26 **Renault**	*An FMS for Gearboxes*	**LUT-FMS Research Group**

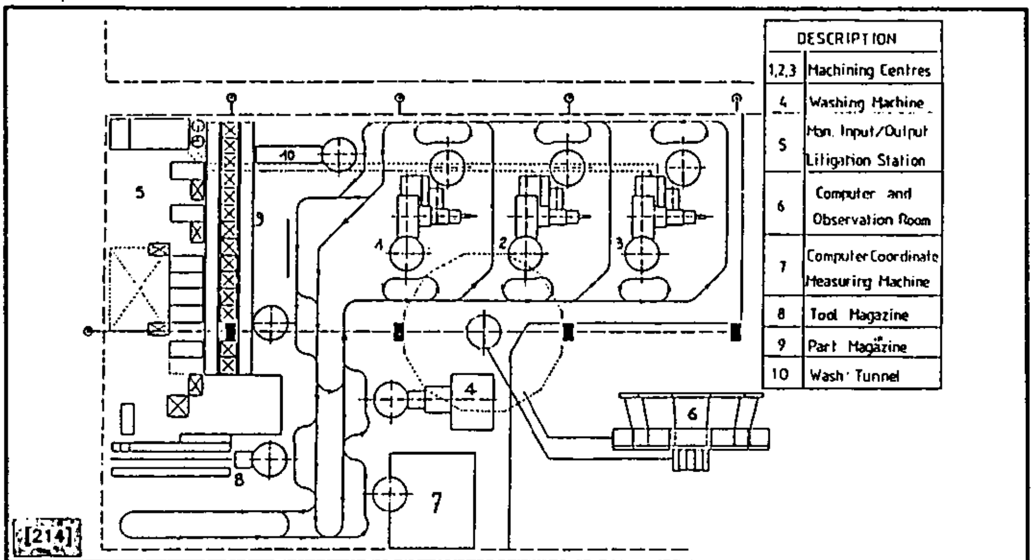


Figure 3.27 **Citroen**	*An FMS for Prototype Automobile Parts*	LUT - FMS **Research Group**

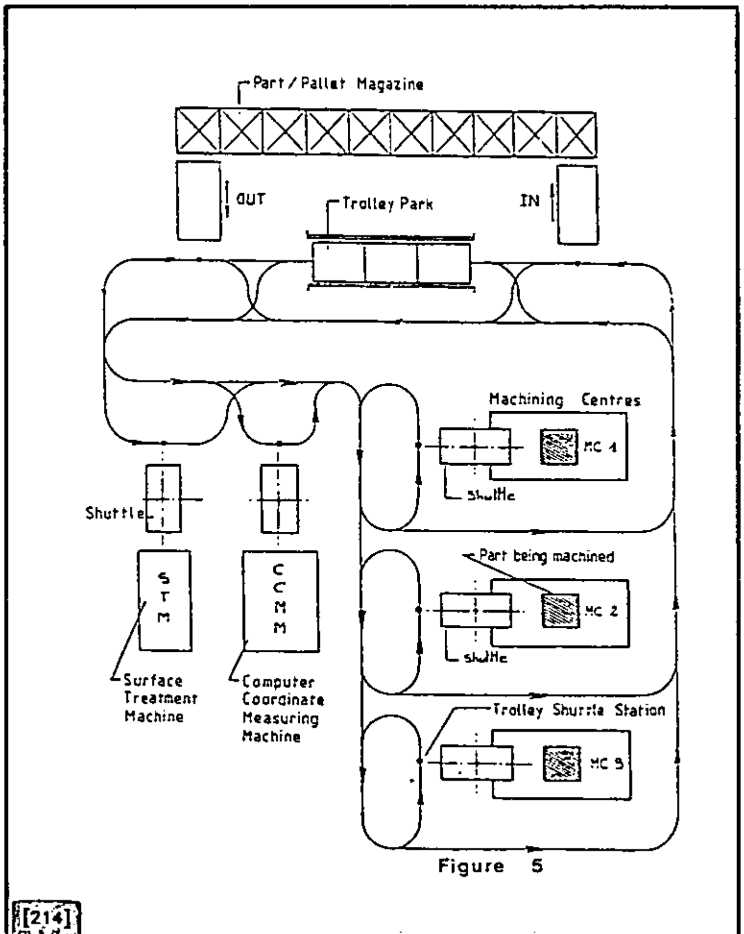


Figure 3.28 **Citroen**	*Work Flow Principles in the FMS*	LUT-FMS **Research Group**

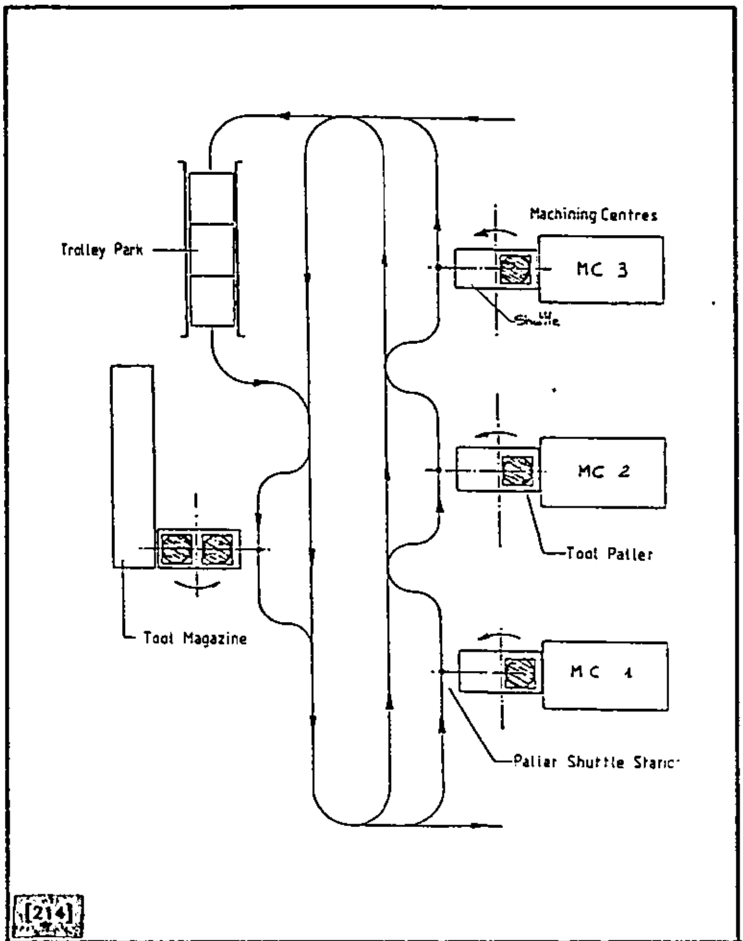


Figure 3.29 Citreon	*Tool Flow Principles in the FMS*	LUT-FMS Research Group

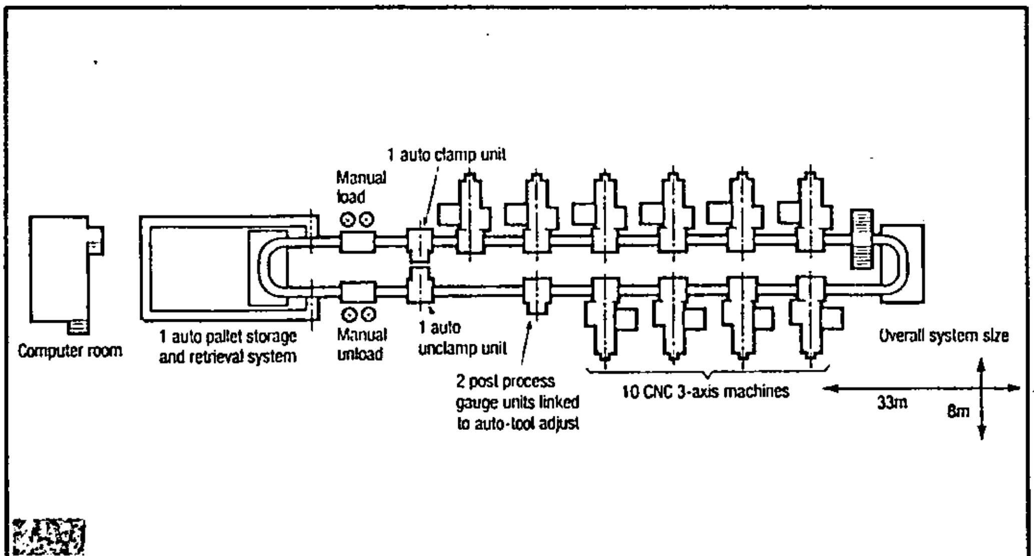
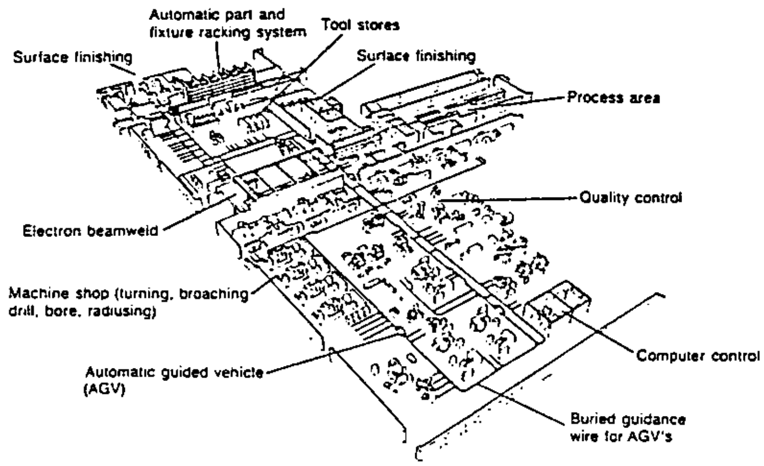


Figure 3.30 Cross International	*An FMS for High Volume Machining*	LUT - FMS Research Group



[57]

Figure 3.31 **Rolls Royce**	*A Large-Scale FMS for Cylindrical Parts*	LUT - FMS **Research Group**

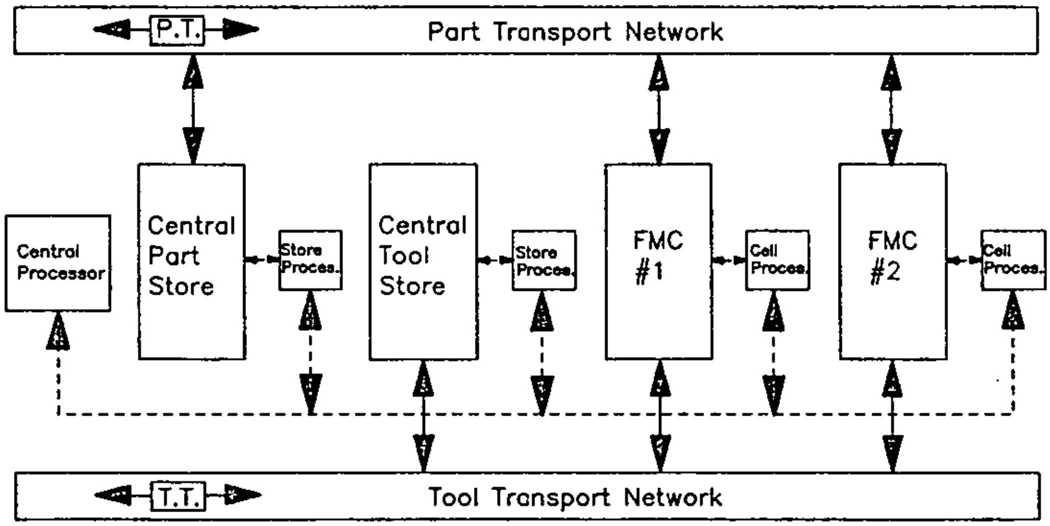


Figure 3.32	*Structure of Multi-Cell Flexible Manufacturing Systems*	LUT - FMS **Research Group**

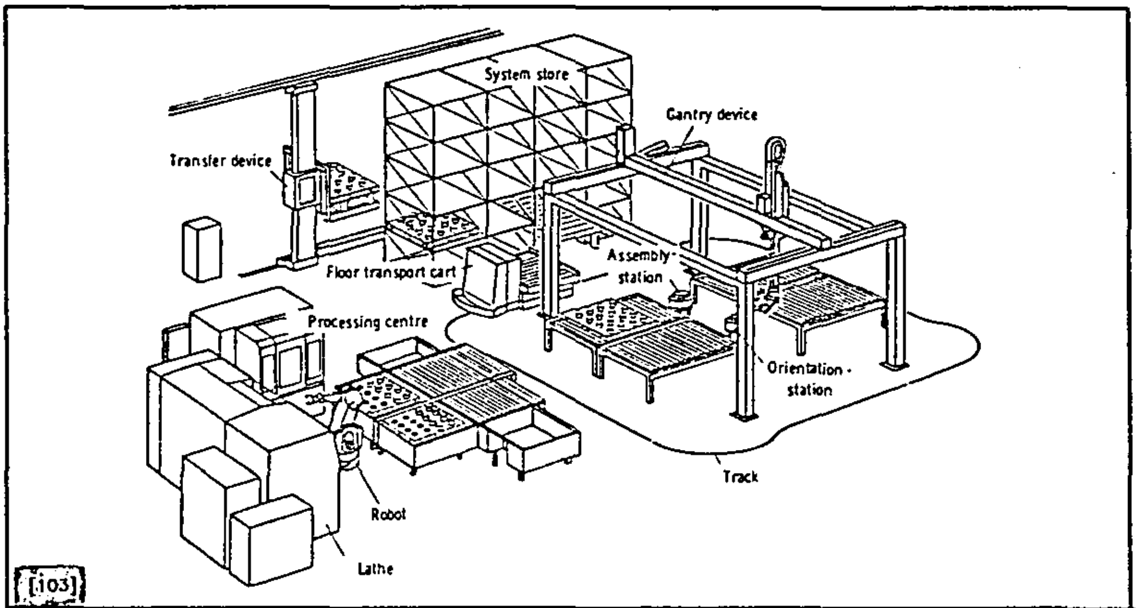


Figure 3.33 **RWTH**	*A Multi-Cell System for Machining & Assembly*	LUT - FMS **Research Group**

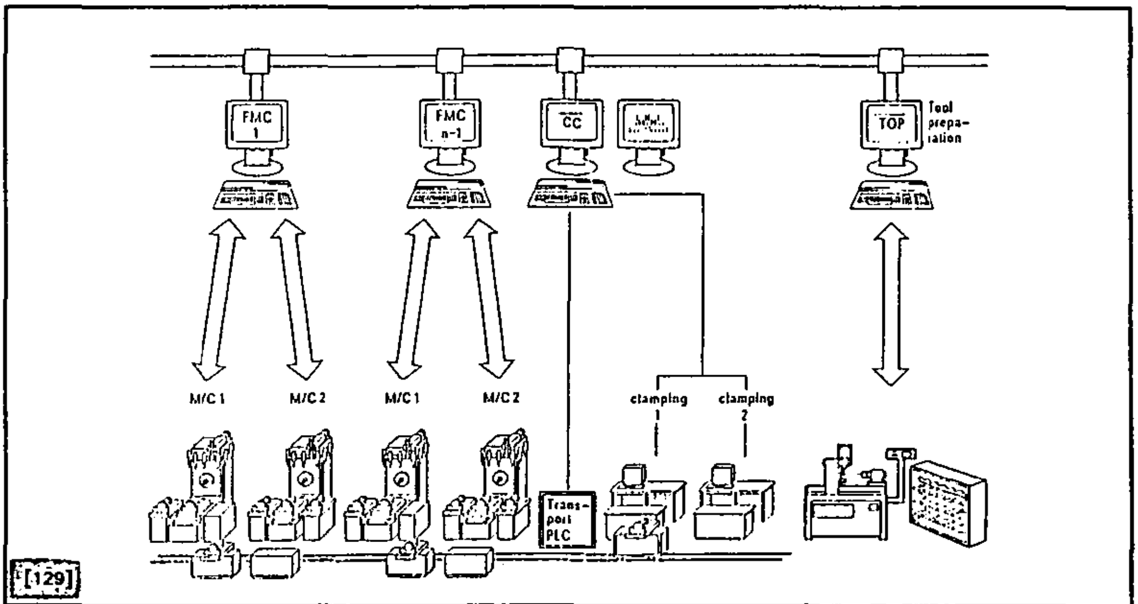
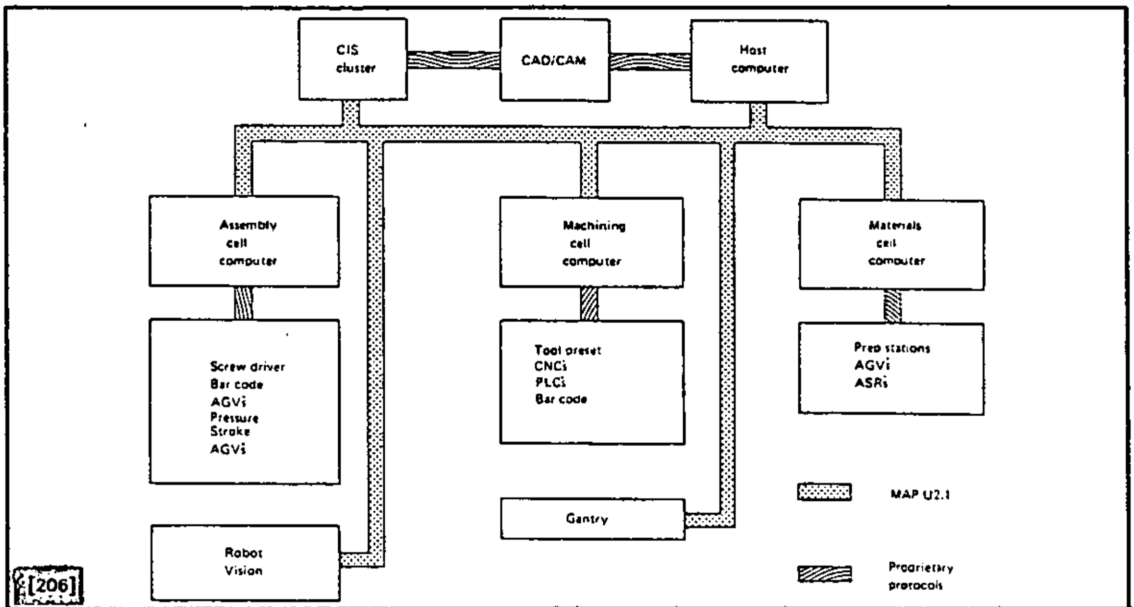
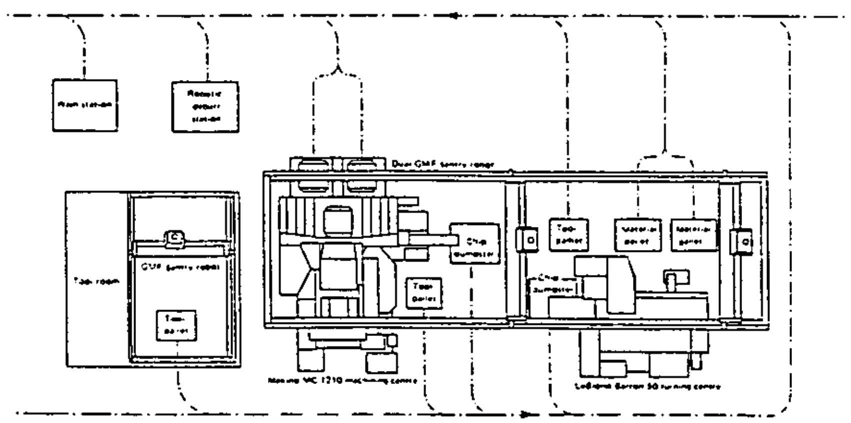


Figure 3.34 **Siemens**	*Control Hierarchy of the Multi-Cell System*	LUT - FMS **Research Group**



[206]

Figure 3.35 **GMF**	*Control Network for a Multi-Cell System for Robots*	LUT - FMS **Research Group**



[206]

Figure 3.36 **GMF**	*The FMC in the Multi-Cell System*	LUT - FMS **Research Group**

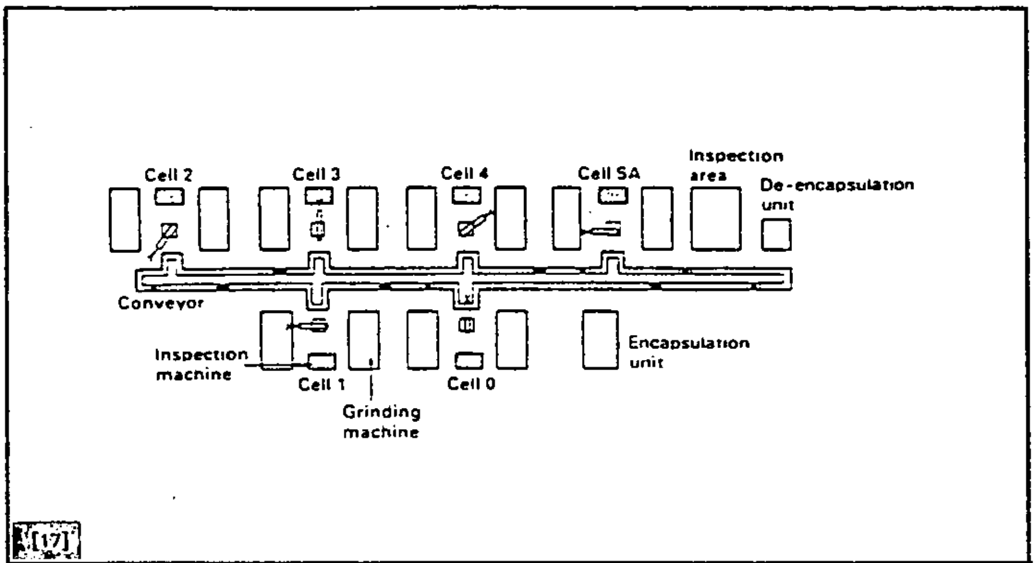


Figure 3.37 **Rolls Royce**	*A Multi-Cell Grinding System*	LUT - FMS **Research Group**

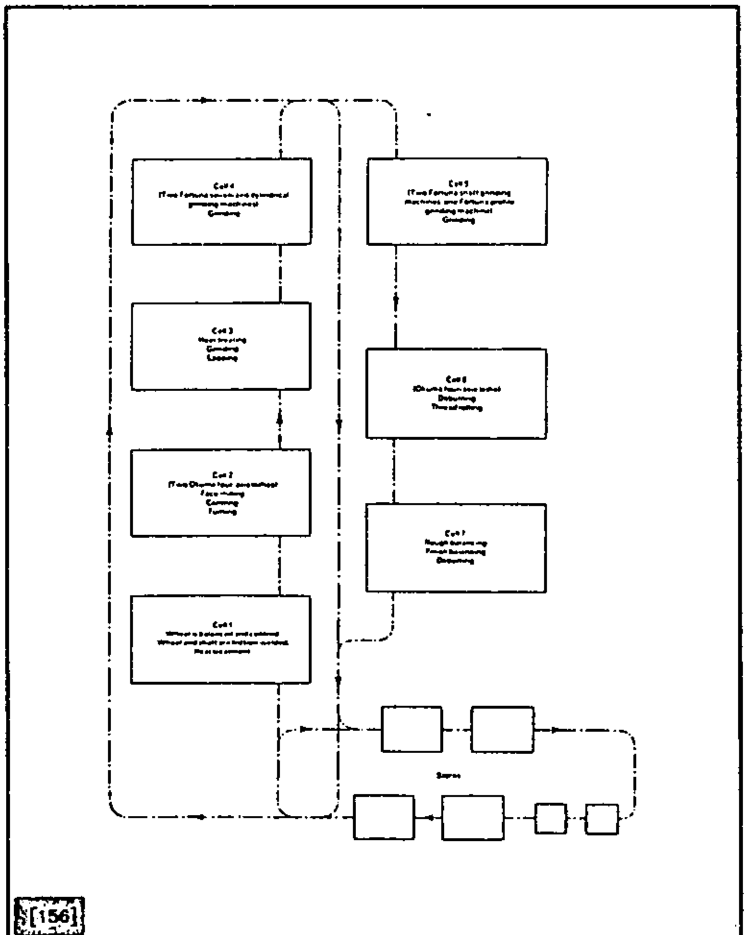


Figure 3.38 **Holset**	*A Multi-Cell System for Turbochargers*	LUT-FMS **Research Group**

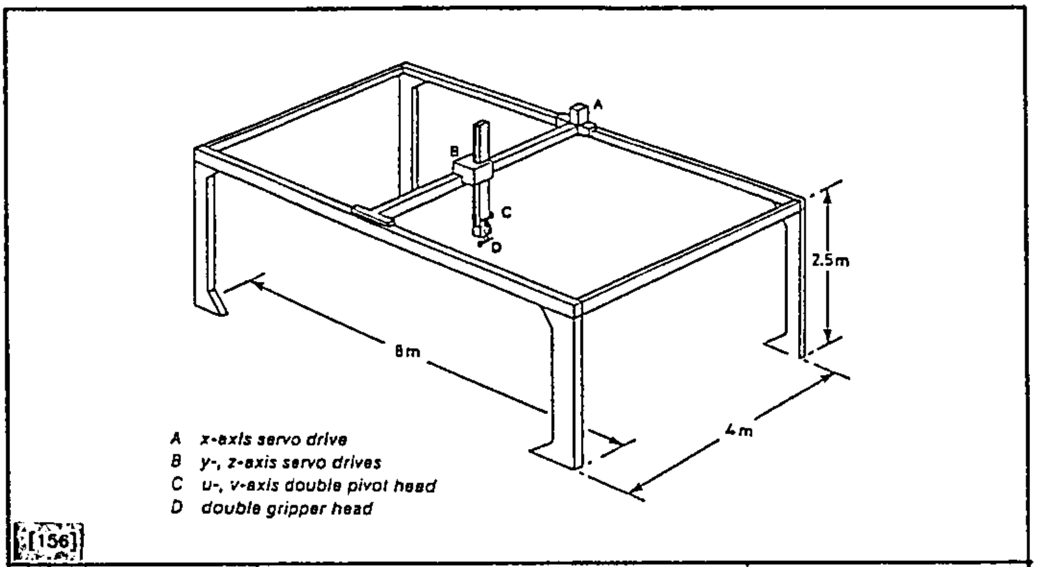


Figure 3.39 **Holset**	*The Cell Gantry Robot in the Multi-Cell System*	LUT - FMS **Research Group**

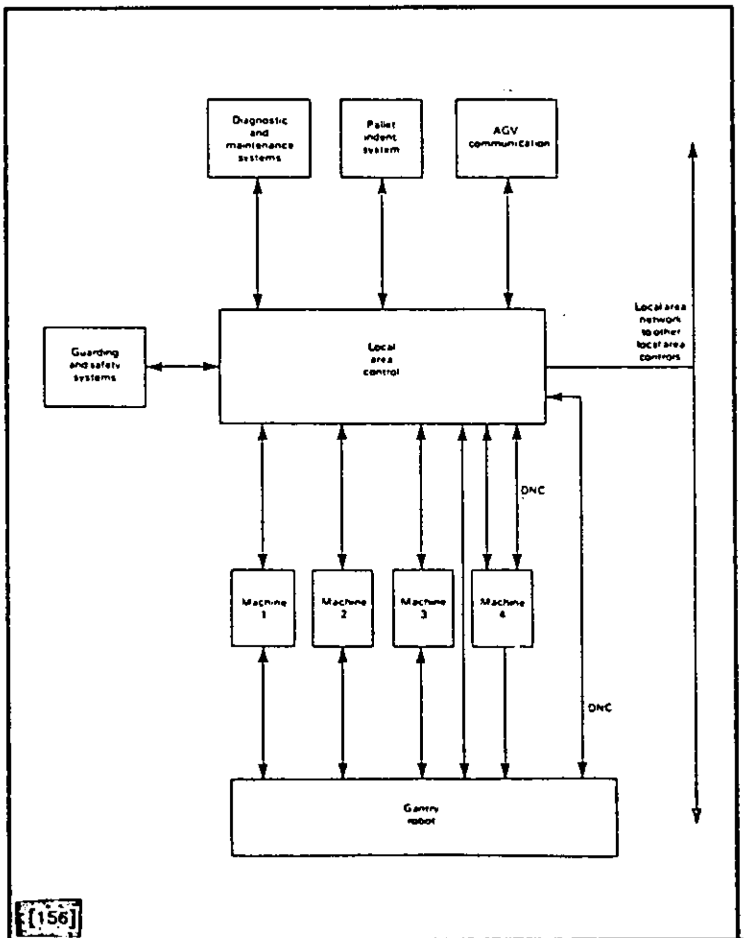
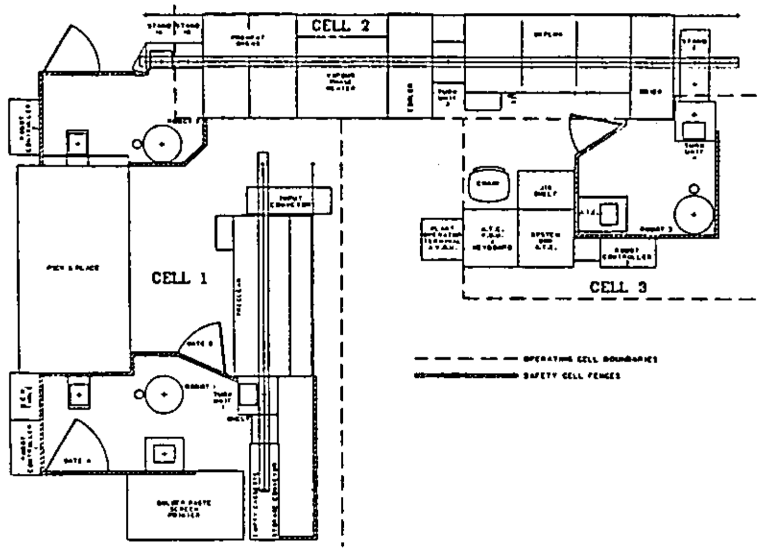
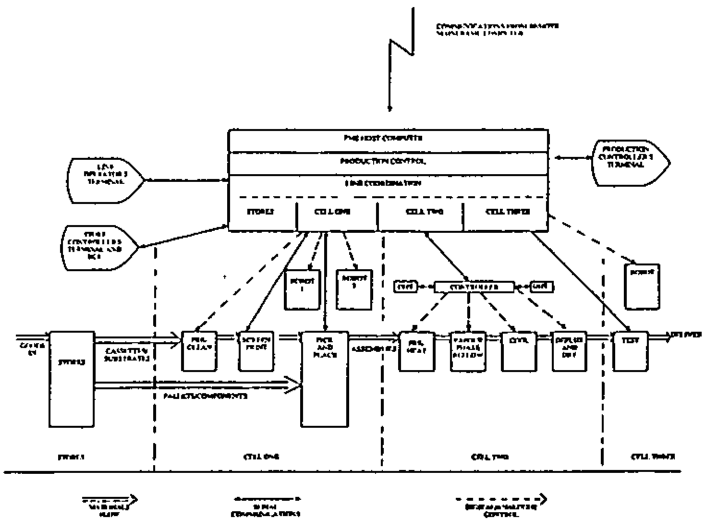


Figure 3.40 **Holset**	*Control Configuration in the Multi-Cell System*	LUT-FMS **Research Group**



[82]

Figure 3.41 **British Aerospace**	*A Multi-Cell System for PCB Production*	LUT - FMS **Research Group**



[82]

Figure 3.42 **British Aerospace**	*Materials Flow, Control & Communications in the Multi-Cell System*	LUT - FMS **Research Group**

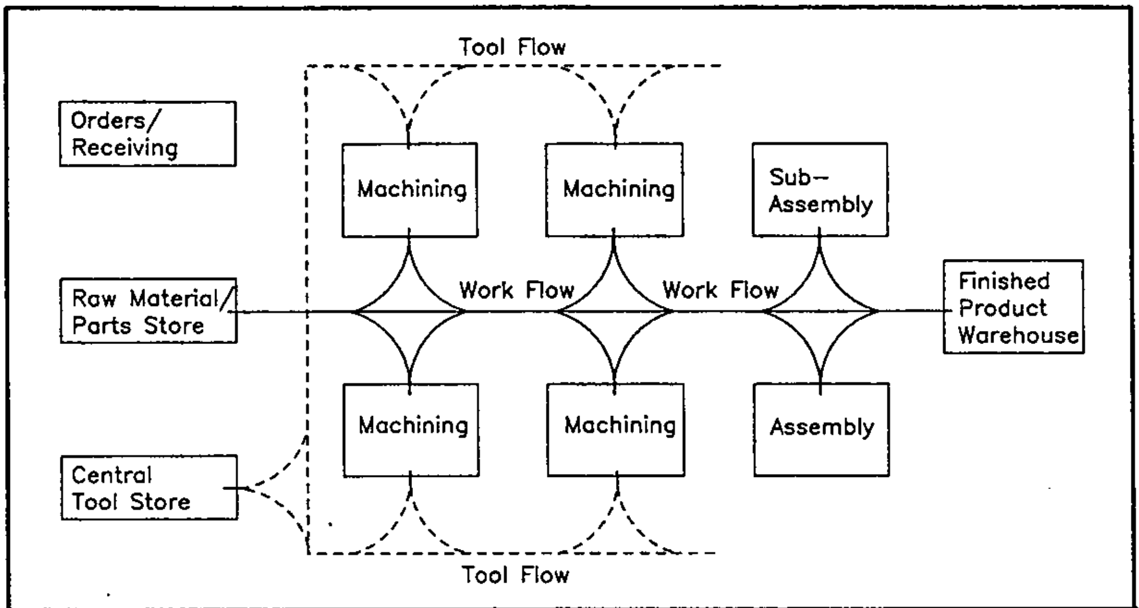


Figure 3.43

Typical Elements of a Manufacturing Area

LUT - FMS Research Group

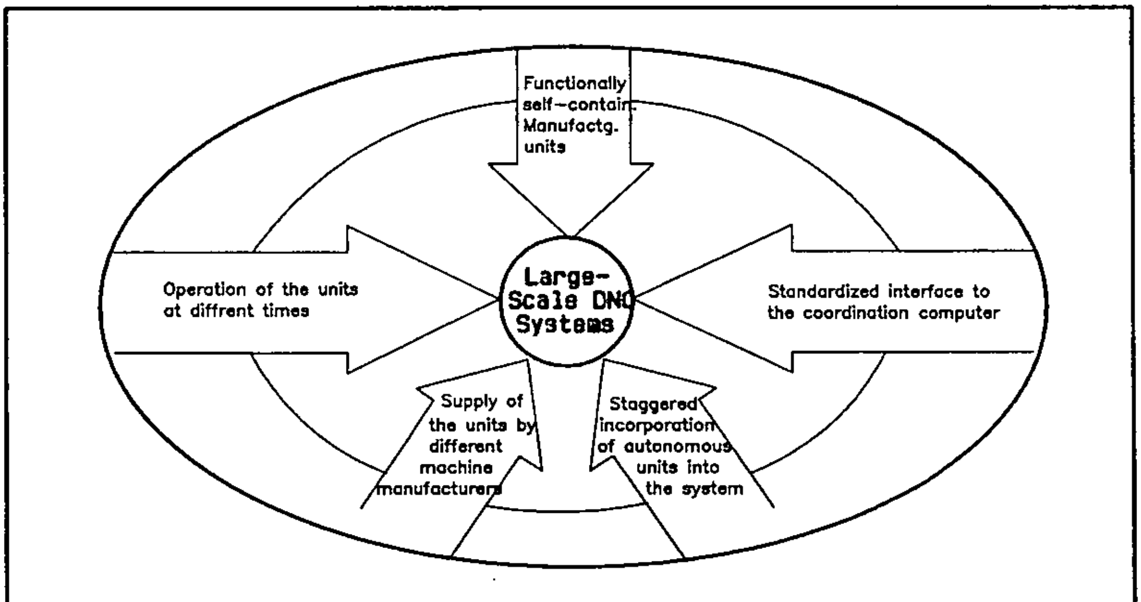


Figure 3.44

Potential Advantages of a Large-Scale DNC System

LUT - FMS Research Group

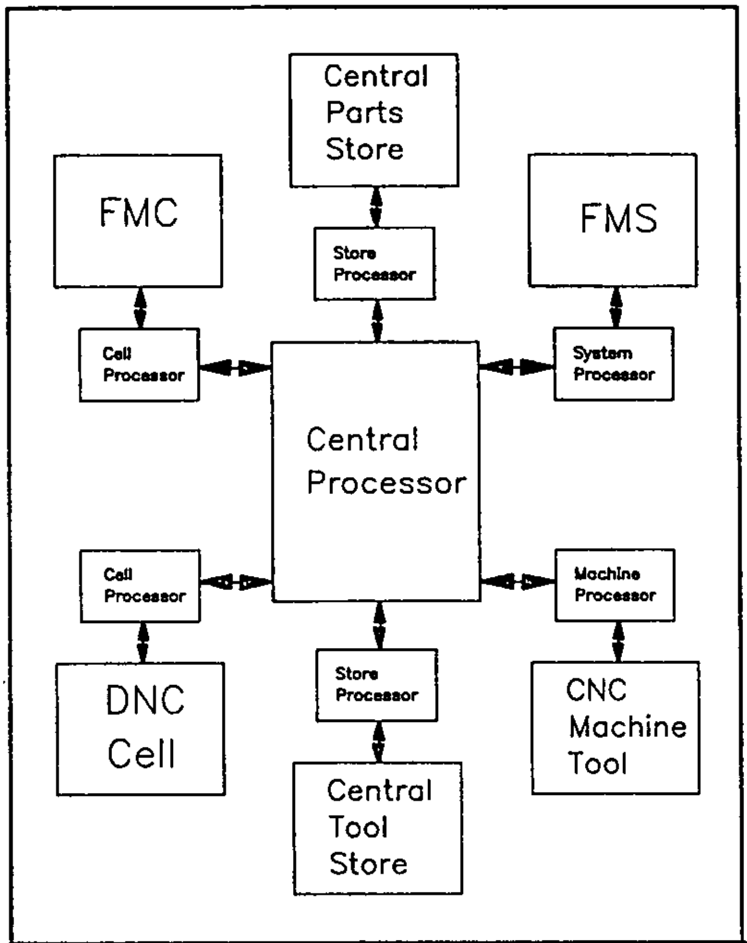


Figure 3.45 *Structure of Large-Scale DNC Systems* LUT-FMS **Research Group**

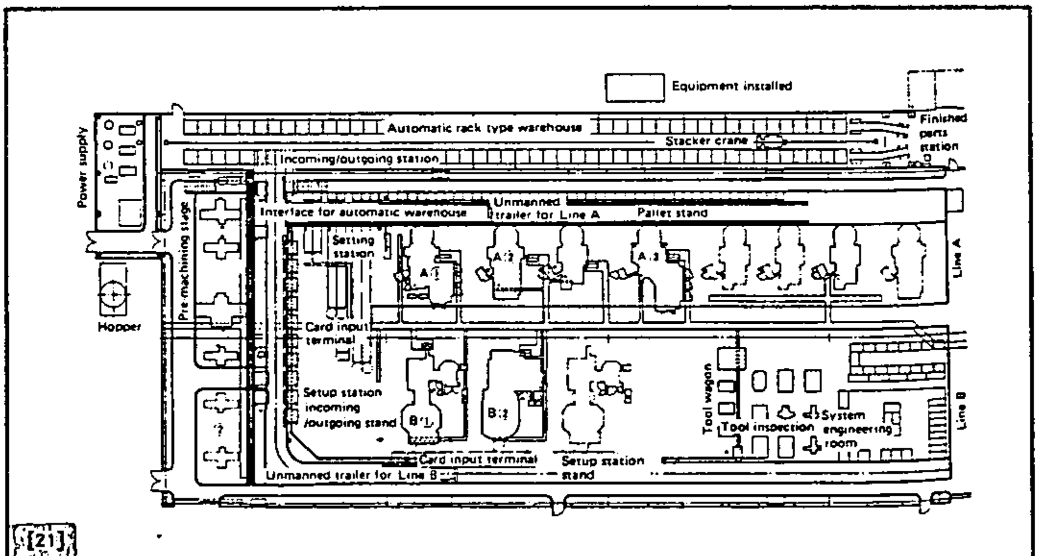


Figure 3.46 **Takisawa** *A Large-Scale DNC System Made up of Two FMCs* LUT - FMS **Research Group**

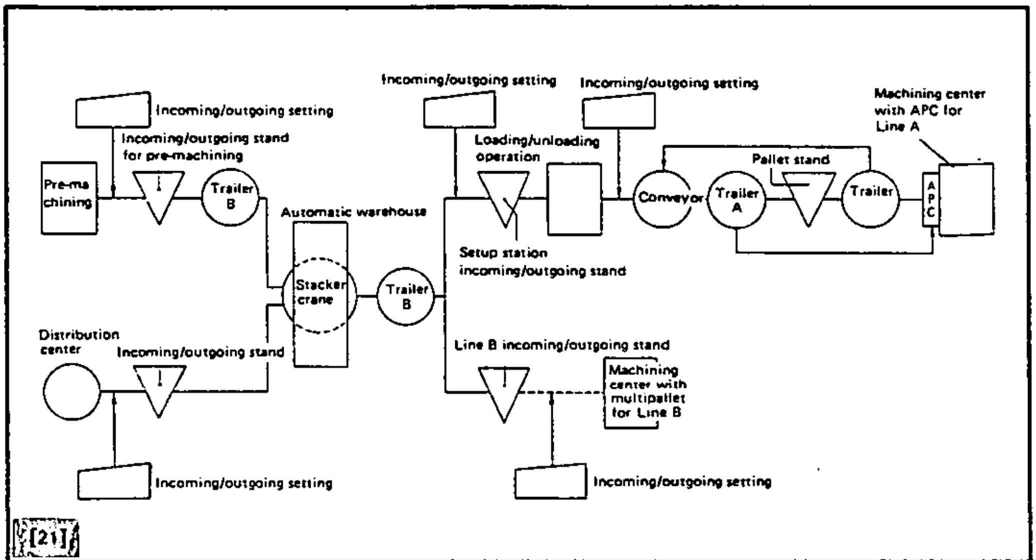


Figure 3.47 Takisawa	*Work Flow in the Large-Scale DNC System*	LUT - FMS Research Group

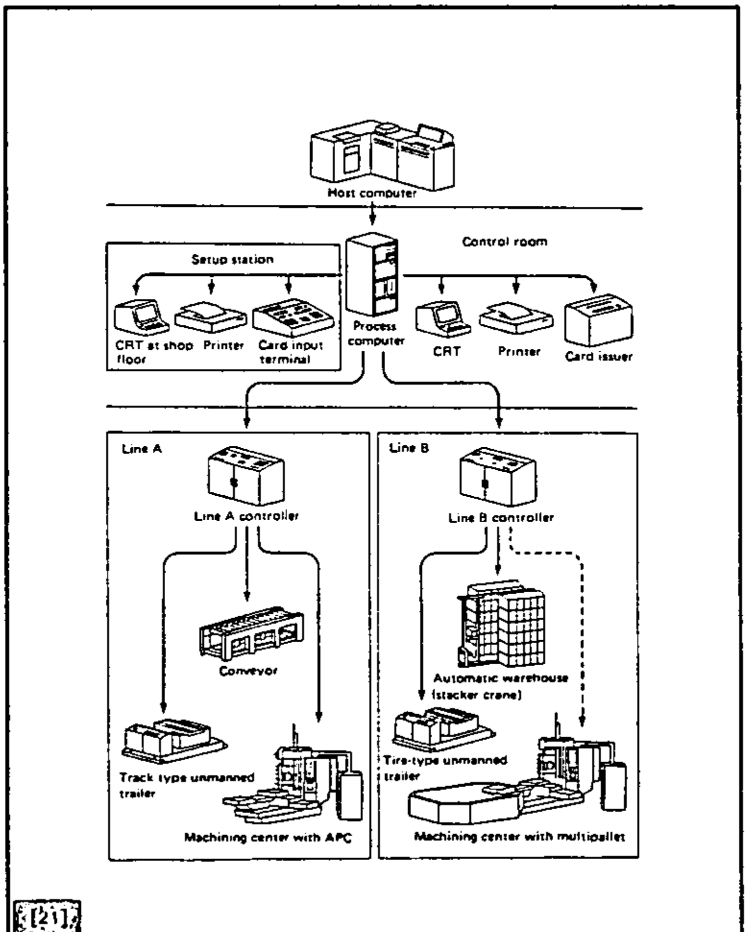


Figure 3.48 Takisawa	*Control System for the Large-Scale DNC System*	LUT-FMS Research Group

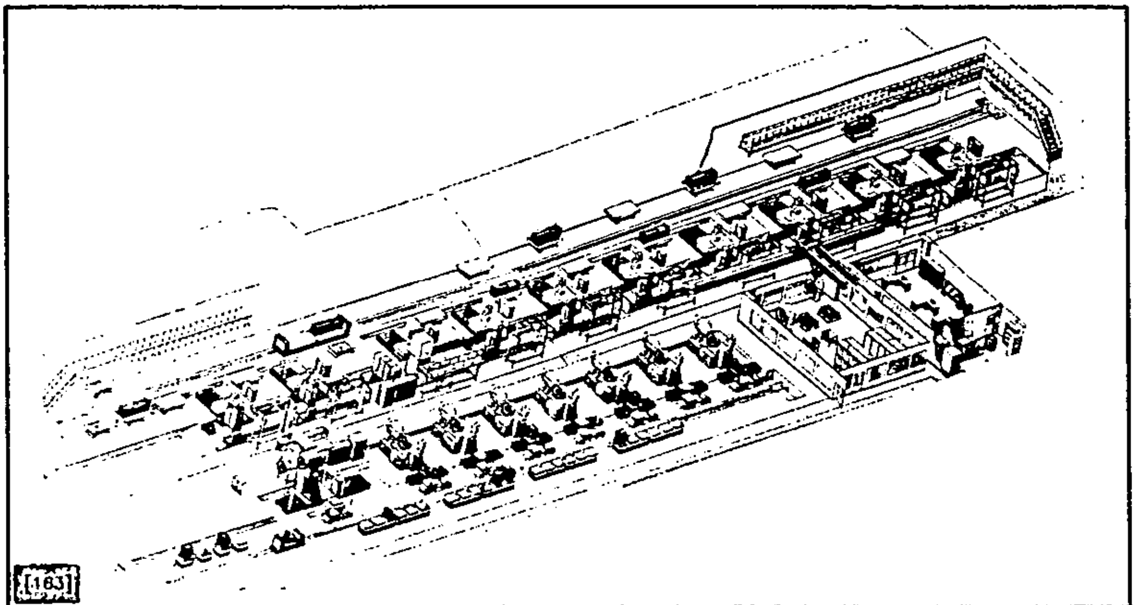


Figure 3.49 **Yamazaki at Oguchi**	*A Large-Scale DNC System Consisting of Two FMSs*	LUT - FMS **Research Group**

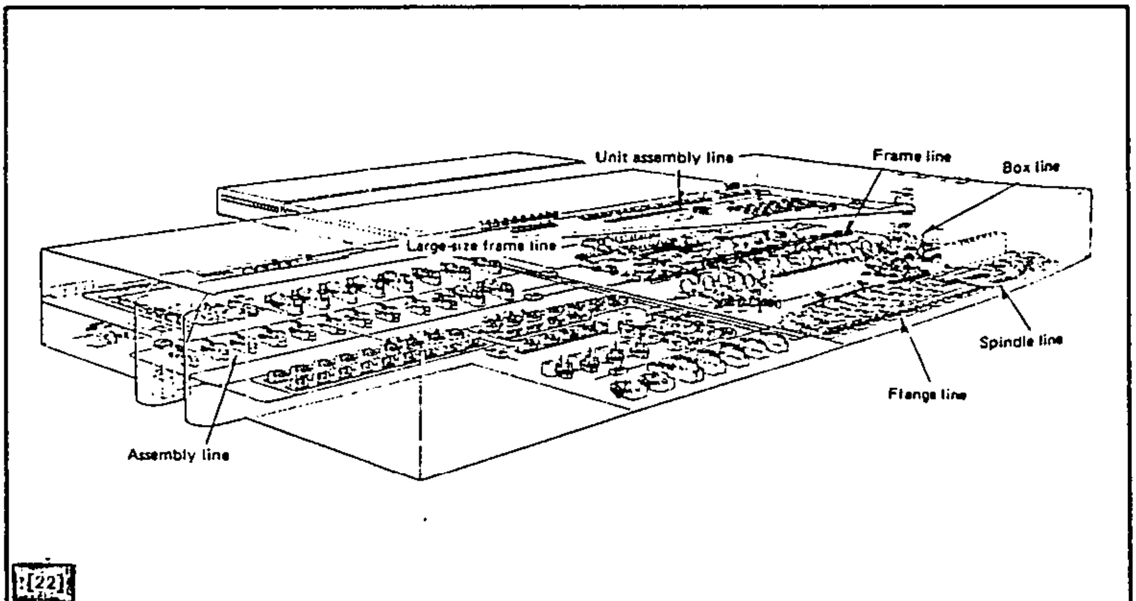


Figure 3.50 **Yamazaki at Minokamo**	*A Large-Scale DNC System Made up of Five FMSs*	LUT - FMS **Research Group**

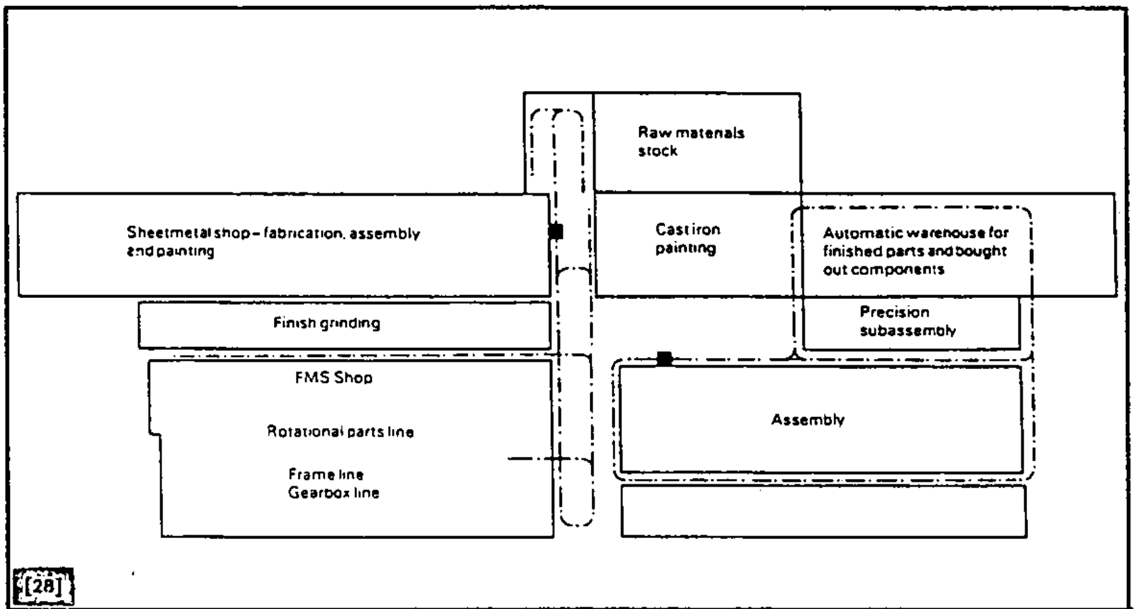


Figure 3.51 **Yamazaki at Worcester**	*A Large-Scale DNC System for CNC Lathes & Machining Centres*	LUT - FMS **Research Group**

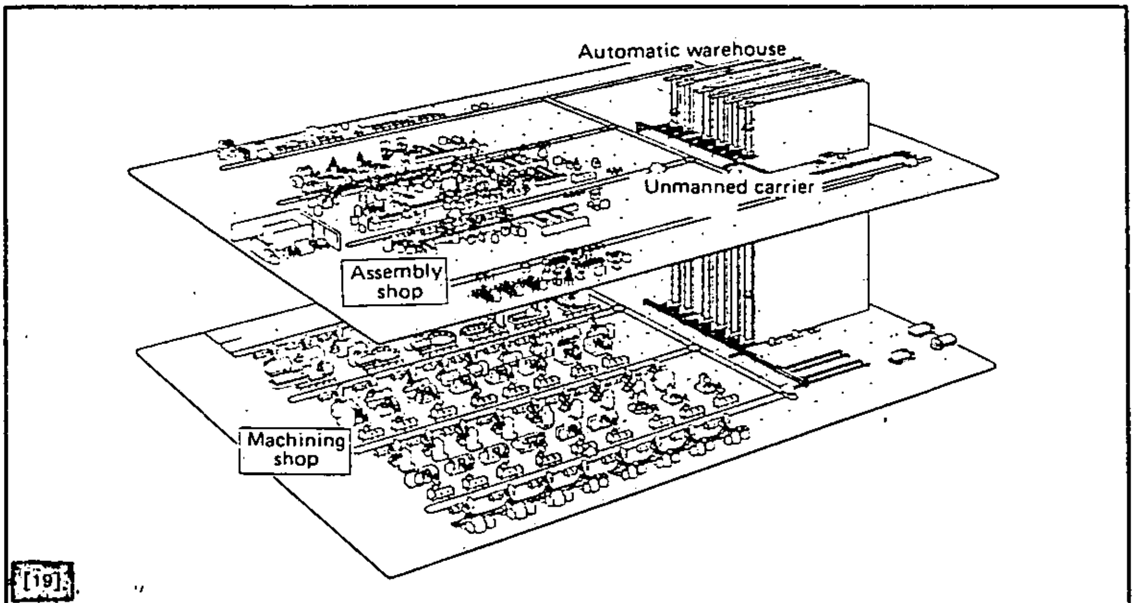
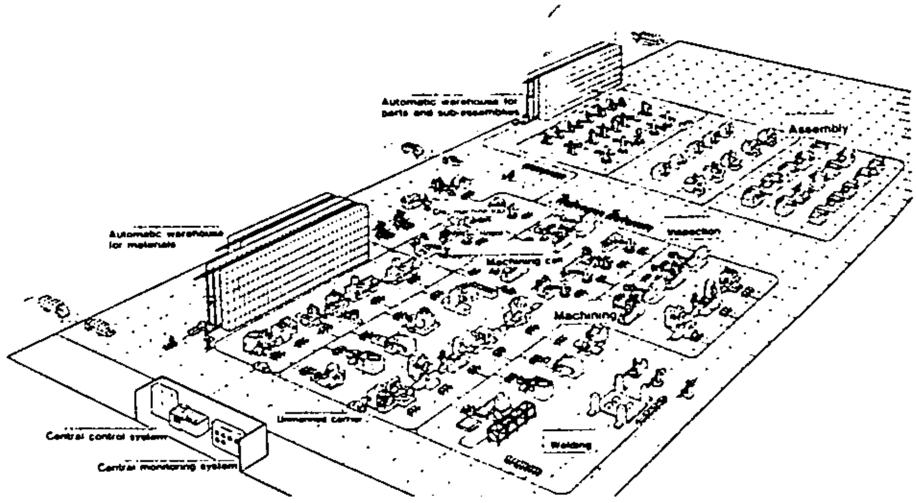
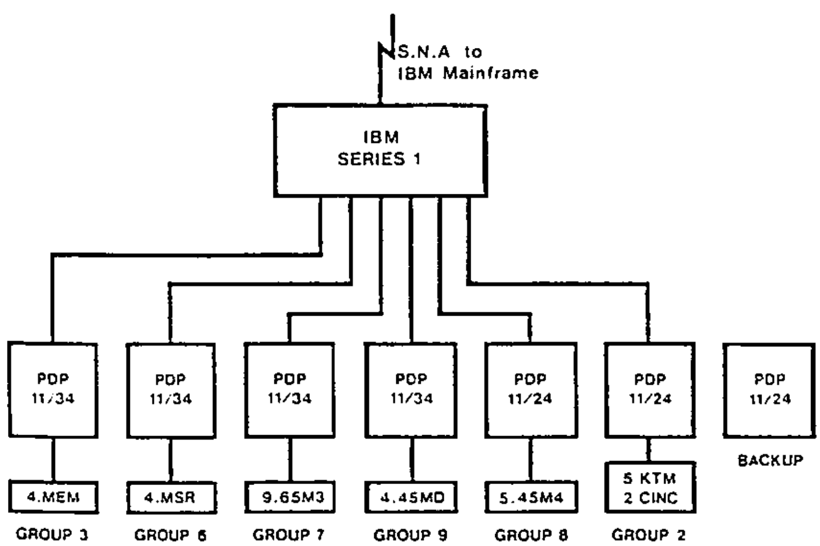


Figure 3.52 **Fanuc**	*A Large-Scale DNC System for Motors*	LUT - FMS **Research Group**



[11]

Figure 3.53 **Fanuc**	*A Large-Scale DNC System for Robots Production*	LUT - FMS **Research Group**



[29]

Figure 3.54 **British Aerospace**	*A DNC Network of Two-Level Digital Control*	LUT - FMS **Research Group**

Chapter 4
THE MODELLING CHALLENGE

4.1 Introduction

This chapter investigates the modelling challenge arising in the design of flexibly automated manufacturing systems. Firstly specifications for these advanced manufacturing systems are discussed. Then the requirements of modelling in flexible manufacturing are identified and examined. Following this, some consideration is given to the examination of the difficulties associated with the modelling of advanced manufacturing systems. Finally the need for this work and the framework for the study are indicated explicitly.

4.2 Manufacturing Specifications

The high capital cost and complexity of work are now accepted features of the design of flexibly automated manufacturing systems. The design study requires a large commitment of manpower and skill for the correct specification and integration of manufacturing elements to enable efficient operation of the system.

Figure 4.1 shows the design process for manufacturing systems. The first step is to specify the present and future manufacturing needs the system has to satisfy. This influences the choice of manufacturing system configurations and prevents the system from being under or over designed. Specifications for flexible manufacturing can be categorised into three major groups: mission specifications, performance specifications and system elements specifications [71].

Mission specifications are the most important needs that the system has to meet. They include:

- Drawings of the parts to be produced.
- Process plans and fixturing concepts.
- Production requirements and available production time,or system throughput, and surge capacity required.
- Delivery date.

Performance specifications are concerned with the behaviour of the system desired. They are:

- Part manufacturing costs.
- Part lead times.
- System WIP inventory level.
- Capability to produce spare parts or change product mix as market requirements demand.
- System availability.
- Desired redundancy.
- Accuracy requirements.
- Level of skilled labour required.

System elements specifications are the optional information on specific attributes of the elements to be used. These include the following:

- Physical capacity of the system, such as horsepower , tool storage capacity, maximum part dimensions and weight, etc. - Machinery desired, e.g., horizontal or vertical machining centres, dedicated machines, head changers, etc.
- Auxiliary equipment, such as inspection machines, washing stations, etc.
- Desired pallets and fixtures.
- Description of material handling devices to be used.
- Desired software capabilities.
- Control strategies for the operation of the system.
- Controllers that are to be adopted.
- Applicable industrial standards, such as those for NC part programs and languages, and the needed computer interfaces.

4.3 Requirements of System Modelling in Flexible Manufacturing

In the increasingly competitive world of manufacturing, modelling has begun to tbe accepted as very powerful tool for the design and control of complex production systems. In the last few years, modelling has gone from a tool of 'last resort' to being viewed as an invaluable design and problem solving methodology which is used more and more by engineers, designers and managers.

Today, as a result of fierce, world-wide competition, industry is being forced to turn to expensive factory automation and careful re-examination of existing operating policies and procedures. Unfortunately, even the most careful analytical design of these highly automated, computer-controlled manufacturing facilities sometimes fails to prevent major and expensive blunders such as AGVs that pile up in traffic jams and major mismatches in capacities between different parts of a proposed system. The complexity of these highly integrated manufacturing systems has caused organizations to increasingly turn to modelling for dynamic analysis of these systems prior to implementation. The stakes are too high and the costs too great to do otherwise. Traditional design and analytical methods have too often proven inadequate to study the complex interactions and dynamic behaviour of integrated manufacturing systems.

Although extensive research has been carried out on the design tools for flexible manufacturing and a cross section of modelling methods have become available, the potential benifits the technology promises to offer still can not be fully realised as a result of the difference between the design performance and the actual performance [166]. Thus it is of crucial significance to establish a new methodology for designing and modelling much more efficient and high productive advanced manufacturing systems [43].

In the following two sections, the requirements for the modelling of these advanced manufacturing systems are examined with respect to the modelling functions and modelling method attributes, based on the discussion on the manufacturing specifications in the previous section.

4.3.1 Modelling Functions

By modelling it is meant that a simplified description is abstracted from a relatively complex reality for the purpose of gaining insight into the behaviour of the system or testing of particular hypotheses [107]. Thus, in addition to the specification of modelling objectives, modelling consists of the process of abstraction, the conducting of experiments using the model and the analysis of results obtained from the experiments. This process is illustrated in Figure 4.2.

The functions that have to be included in a model for a flexible machining cell can be summarised as follows (Figure 4.3) [2], based on the discussion on the manufacturing specifications:

(1) ***Modelling of alternative flexible manufacturing configurations***: Depending on the specification of different production requirements by different manufacturing organizations, there can be differing design solutions [15]. Even with the same specification, alternative configurations are possible [54]. Therefore, a modelling mothodology must be able to allow for the modelling of alternative system configurations.

(2) ***Modelling of machining stations***: Since the capanilities of a filexible machining cell are uniquely identified by the machines it contains, careful modelling of the variety of machines is of vital importamce. In general, horizontal-spindle machining centres are the key metal-removing machines in a machining cell, though any particular line may employ a variety of special-purpose machines to support these basic machines. Typical examples are multiple-spindle machines (such as head changers) to most efficiently produce hole patterns and special single-purpose machines (such as broaching, planing, hobbing, turning and even grinding machines) to accomplish machining operations not performed by machining centres [15] [125].

For prismatic parts, the usual choices of machines for a machining cell are between various vertical and horizontal machining centres and special-purpose machines, such as head changers and head indexers. To accommodate a mix of strictly prismatic parts with other prismatic parts requiring large bores or circular bearing surfaces, vertical turret lathes can be used.

With respect to strictly rotational parts, i.e. bars and shafts, standard CNC lathes with both bar and chucking ability can be integrated to form a rotational machining cell [125] [16], but currently this concept only exists on a small scale in flexible machining cells [275] [197].

To perform the required operations, all the machining centres must have tool storage capabilities, either in the form of a drum [120] [163] or tool chain [298]. Tooling requirements for the workpiece variety of a machining cell usually put extreme demands on storage capacity. It is not uncommon to need more than 100 pockets in a tool magazine [84].

Vertical turret lathes must be equipped with pallet shuttles before they can be integrated into a flexible machining cell. External turning, facing and boring operations can all be performed with little need for a tool changer. Usually a four to six tool block-indexable turret with dedicated tools is more than enough to complete the necessary truning work content. Tool changers have to be provided, however, if there is a need for a variety of different turning and grooving tools.

(3) *Modelling of auxiliary stations:* In addition to the machining stations, auxiliary stations need to be incorporated into the cell in order to support the main machining functionality of the cell. These stations include load/unload stations, inspection stations and washing stations. The principal requirements of a load/unload station include a clean support for the pallet in a position accessible to the transporters, access around the pallet to permit the loader to remove and load workpieces [125] [12].

The inspection process can be performed on- or off-line and each has its advantages. An on-line inspection machine can be programmed to identify machining errors and implement tool offset changes directly through the central computer [214].The greatest benefit of an on-line inspection system is the quick identification of manufacturing problems. An off-line system has inherent lags due to remote location, part fixturing or locating delays and perhaps lack of automated inspection.

The washing stations may or may not be separate entities in a machining cell, because they could be integral with the load/unload stations. Here, washing is considered chip removal from the parts, fixtures and pallets [96].

(4) *Modelling of different material handling systems:* There are two prinsipal forms of part transport: parts must be introduced into the cell, and they must be transferred between stations within the cell. It is usually not convenient to combine these two functions because movement into the cell involves raw parts whereas within the cell involves part, fixture and pallet assemblies.

Since mounting parts on fixtures is usually a manual operation, introduction of parts into the cell, is performed manually. Various cranes and robots can be empoyed to maneuver parts too heavy to lift manually. These facilities would be located near the load/unload stations; and bins, magazines, or pallets of raw parts should be stacked nearby to facilitate the loading function [66].

With the cell, there are three major pallet-movement designs, i.e. AGVs [135] [158] [120] [12] [298] [66] roller conveyors [33] [16] and robots [125] [157]. Guidance and control of AGVs can take three basic forms: rail-, antenna- and tow chain-guided. For the rail-guided AGVs, sensors located at appropriate points along the rail identify the precise location of the vehicle and can be used to position it to the required tolerance to transfer pallets to a machine or unload station. The antenna-guided vehicles are usually battery-powered and can move along a flat floor. A wire embedded below the surface is detected by the antenna on the AGV. Position sensors still must be used to control pallet transfer. The third AGV design uses a tow chain in a trough under the floor. The chain moves continuously and the AGV movement is controlled by extending a drive pin from the vehicle down into the chain. At specific points along the guideway, computer-operated cam-type stop mechanisms raise the drive pins to halt the AGV movement.

A roller conveyor system can be designed to move pallets from the load stations to pallet changers located on the machines. Individual sections can have separate drives to control placement of pallets near machines. In contrast to wire- or tow-guided AGVs, a conveyor system limits access to the major elements of the cell because it must be raised above the floor level so that it is aligned with the pallet changers on the machines.

Robots are a special consideration for workpiece transfer and are generally applicable where spacing between machines is short and workpieces plus fixtures are relatively lightweight. They are mostly used when machines are clustered in a circular work cell so that one robot can serve several machines. They are often used with unfixtured rotational parts [125] [198].

(5) *Modelling of various temporary buffer storages:* In addition to on-shuttle and off-shuttle queues at stations, several different kinds of buffer storages can be incorporated into a cell. These storages are necessary to gain flexibility in sequencing production through the cell and to allow for contingencies on the line, such as machine or tool failure. A obvious form of buffer is a separate loop of track or conveyor where pallets can be shuttled to allow others to proceed past them on the direct route [71].

In addition to the separate temporary storages, empty transporters can also be used to serve to buffer unwanted pallets. Extra loading stations also can act as buffers of limited capacity.

(6) ***Modelling of various control functions and decision rules for the management of work flow in the cell:*** The control of a machining cell with regard to the work flow is usually facilitated at the following three levels [59] [41] [250] [254] [269], and therefore rules concerning these functions have to be modelled:

- *Level 1:* The decisions that are made at this level include the selection of part mix for a particular time interval, planning of the production requirement compatible with cell capacity.

- *Level 2:* This level is mainly concerned with the batching of components, balancing of workload assignment to the cell resources, and the timing and sequencing of parts relaesing into the cell.

- *Level 3:* At this level, decisions have to be made with respect to the workpiece/transporter movement, route management and reacting to disruptions.

(7) ***Modelling of the tool flow strategies in the cell:*** Since economic and effective solutions to the tool flow requirments of flexible machining cells are becoming increasingly important and there is clear evidence of major hardware developments by machine tool builders towards increasingly sophisticated networks for the flow and exchange of preset tools between tool stores [214], modelling of tool life checking, too stores, tool transport system and tool flow stratagies can be of crucial importance in these installations.

The tool stores are usually organized in a three-level hierarchy, i.e. the machine primary tool store, the cell secondary tool store and the factory central tool store. In a flexible machining cell, the first two types of tool stores must be modelled.

Tool transfer is mostly between the primary tool stoares and the secondary tool store at the cell level. It may accomplished either by using the workpiece transfer system or a separate tool transfer system. A number of alternatives may be possible depending on the nature of the tool flow and the machining installation under consideration. The tool transfer devices mainly take the form of either an AGV [26] [214] or gantry robot [28].

The difficulties encountered in managing the tool flow, point to the need for strategies to deal with specific operational problems, such as tool assignment and tool issue in the activity flow networks. Each of these operational strategies and their relationships with the loading, scheduling and tool management strategies contribute to a total tool management solution [84] [304]. Thus methodologies must be developed to allow for the modelling of these strategies and their interactions.

4.3.2 Modelling Method Attributes

As shown in Figure 4.1, modelling plays a key role in the design of a manufacturing system after the specification of manufacturing goals and system elements. It helps in designing and evaluating alternative configurations. The design of alternative configurations is carried out by choosing the type and number of workstations, the work/tool transport devices and work/tool storage facilities, and laying out physically all the elements chosen [227]. The evaluation of these configurations is realised through the following procedures [71]. Firstly an evaluation matrix is constructed to show all the criteria which are considered important to the evaluation process. Then operational strategies have to be developed, such as loading, release and control rules. Next the operatin of particular configurations should be modelled in order to provide performance measures for economic analysis. And finally the economic analysis itself is performed to estimate return-on-investment, payback period and other economic factors.

With the above end in view, the attibutes a modelling methodology should possess can be brought together into the following (Figure 4.4):

- ***Flexibility with model building:*** The approach should allow alternative systems to be modelled with different criteria. This, for example, is required when the specifications of the system design need to be modified as a result of the unsatisfaction of the performance output.

- ***Details contained within the model:*** As a result of the complexity of the design process and the need for a structured design system, models can be required to run at various levels of detail for different purposes. For example, in the very initial design stage, modelling is usually required at an aggregate level, while for the subsequent implementation, detailed decisions have to be made by the model.

- ***Efficiency of the modelling process***. Since flexible manufacturing systems are complex systems and a large number of variables can be involved in the design and modelling process, the modelling methodology needs to be efficient in order to provide an adequately quick estimation of the system performance with the available computing equipment.

- ***Transparency of the modelling process***. To establish the credibility of the model, knowledge embedded in the model should be able to be entered by the user, and the modelling process should be transparent to enable the user to understand what the model is trying to do and how it is trying to do.

- ***Confidence associated with the system outputs***. To enable the fullfillment of the potential benifits offered by the system, unrealistic assumptions should not be made in the model and the modelled performance should be as consistant as possible with the realised performance.

4.4 Difficulties of Modelling in Flexible Manufacturing

It is universally accepted that modelling plays an essential and crucial role in the design of advanced manufacturing systems. However, modelling is of value only if the insights generated in the process can be used to impact reality. This is achieved by ensuring that the right features of the system are captured and the model results make sense [107].

With regard to the modelling of flexible manufacturing installations using powerful computer hardware and complex software, it is evident to point to the following difficulties (Figure 4.5) [45]:

(1) A system model for flexible manufacturing typically contains many types of knowledge that collectively represent a real system. Some types can be easily represented and are understandable, e.g., properties of system elements like the size and capacity of a machine. Other types of knowledge, such as how system elements behave, how they interact with each other, and how decisions are made, may be very difficult to be represented and understood. Some of the information can also get lost in the translation to computer code. Therefore, approximation is always required for modelling. This makes it impossible to formulate an exact model for flexible manufacturing.

(2) Since knowledge is usually neither explicitly represented nor well structured in a model, it is frequently difficult for the user to truly understand what the model can do and how it can do it. Embedded assumptions tend to be hidden, scattered, and fragmented throughout the computer program. The initial structure of the first version of a model is often lost as more complexity is added or modifications are made. As a result, it is difficult to assure that the model is an adequate representation of the dynamic system. The user can have little confidence in either the predictions or the design advice the model might suggest.

(3) As the system designer proceeds from aggregate analysis through more and more detail down to the level of control system design, models are required to allow smooth transitions. However there is little evidence that existing models can reason over multiple levels of detail. It is often necessary to start over with new data formats, conventions and terminology at each step.

(4) Because of the mathematical nature, the modification of analytic models requires considerable expertise and effort. As for discrete event simulation models, usually the potential problem is that the system elements are hidden in masses of code, or worse, distributed across the code. As a result, users of the model will require very large degree of effort to alter them in a coherent fashion. The model may thus inhibit rather than promote the easy changes required to provide a good environment in which to investigate alternatives. Many commercial systems, however, have user friendly interfaces and powerful graphics support that reduce the problem [165].

(5) Although some models can be expected to produce adequate data that describe the behaviour of the modelled system, if the data are not presented effectively, it can be difficult to see the most important behavioural properties of the system. For example, large manufacturing system simulations can generate enormous output data. Determining the main global features of the system's performance from such output may not be easy, and important trends may be overlooked.

4.5 Framework for the Study

Based on the recognision of the above modelling challenge, this thesis explores the possibility of developing a new method for the modelling of advanced manufacturing systems. This can be achieved by assessing the currently available modelling methods and then identifying techniques that can be applied.

The emphasis of the study will be placed on the structure and design of a modern modelling system based on the identified techniques. This new modelling system should meet the above discussed system modelling requirements and overcome the difficulties posed by conventional modelling approaches.

The potential value of the new modelling method is to be assessed by conducting realistic experiments on the new system, and comparing the results with those obtained from the existing modelling methods.

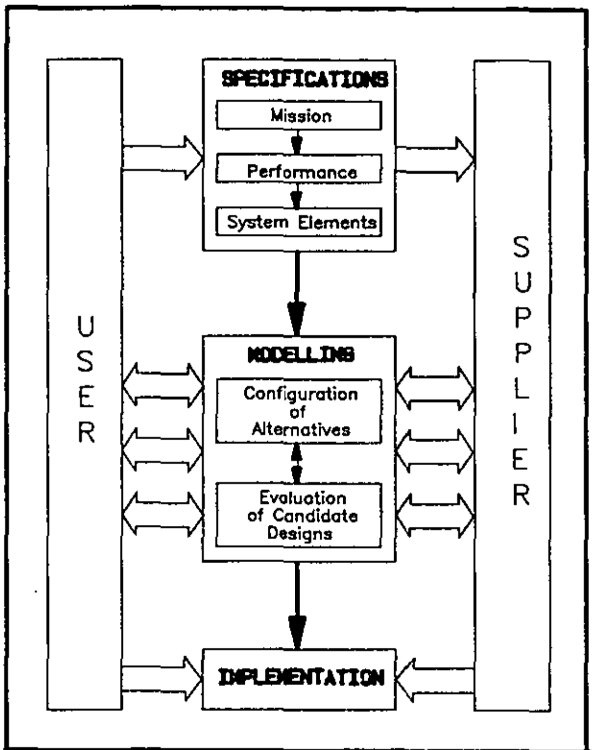


Figure 4.1	*The Design Process for Manufacturing Systems*	LUT-FMS Research Group

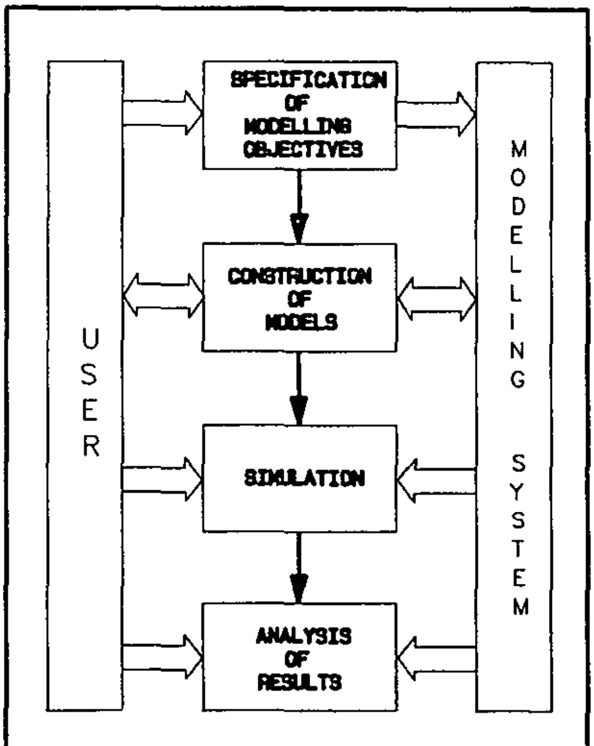


Figure 4.2	*The Modelling Process in Manufacturing Design*	LUT-FMS Research Group

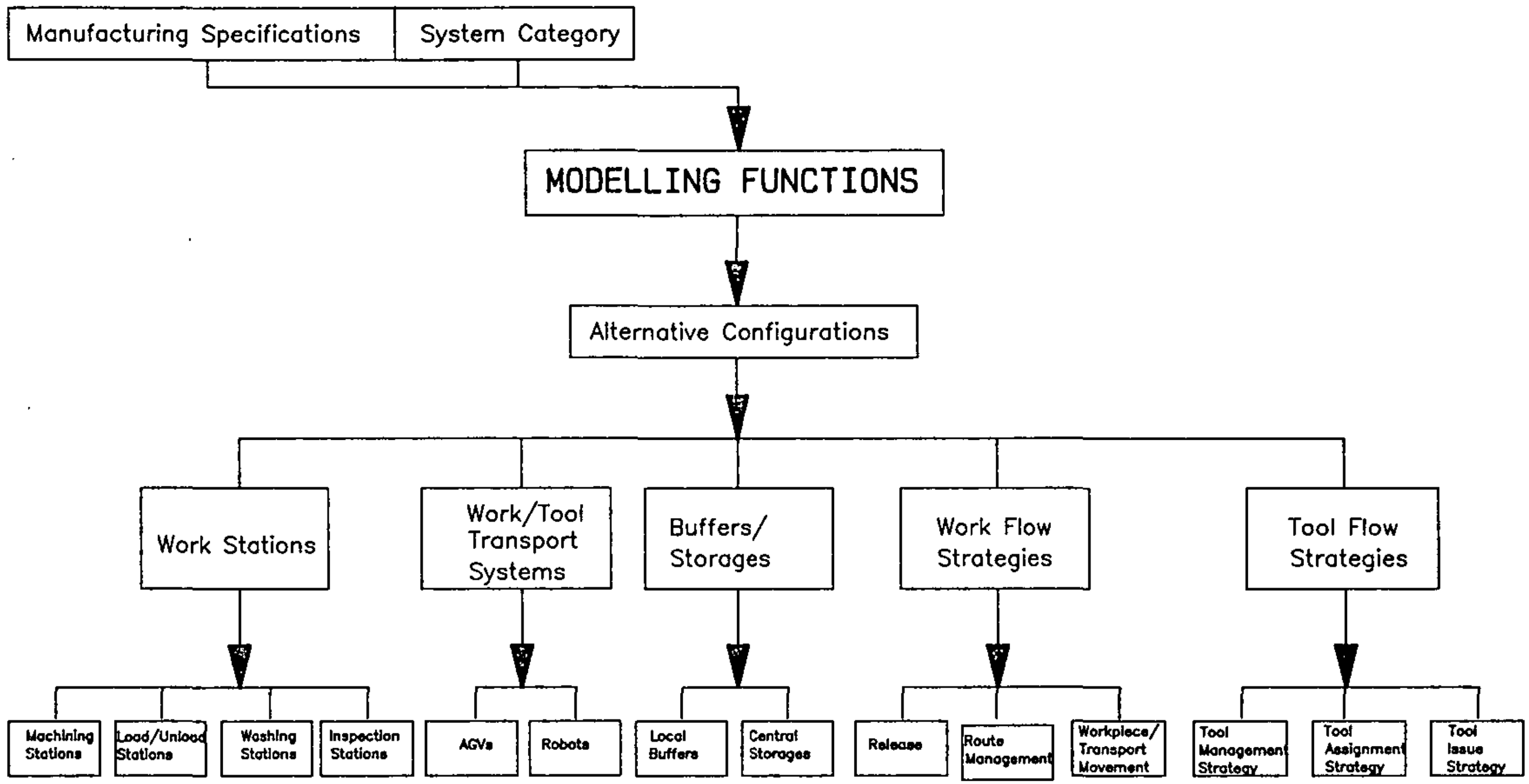


Figure 4.3

Modelling Functions for Manufacturing Design

LUT - FMS
Research Group

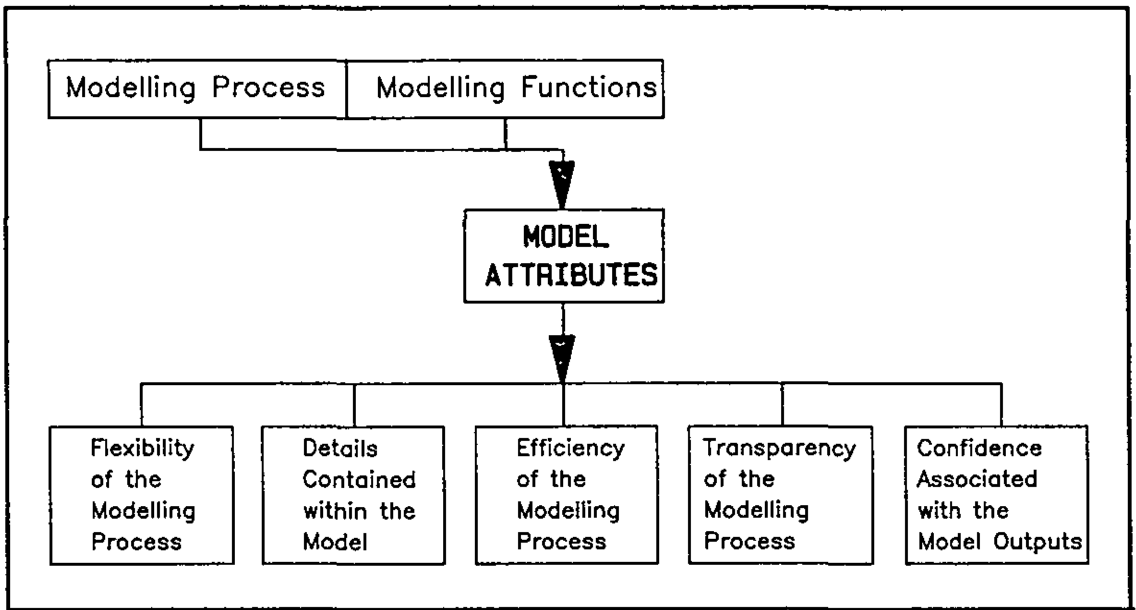


Figure 4.4 *Model Attributes for Flexible Manufacturing Design* LUT - FMS Research Group

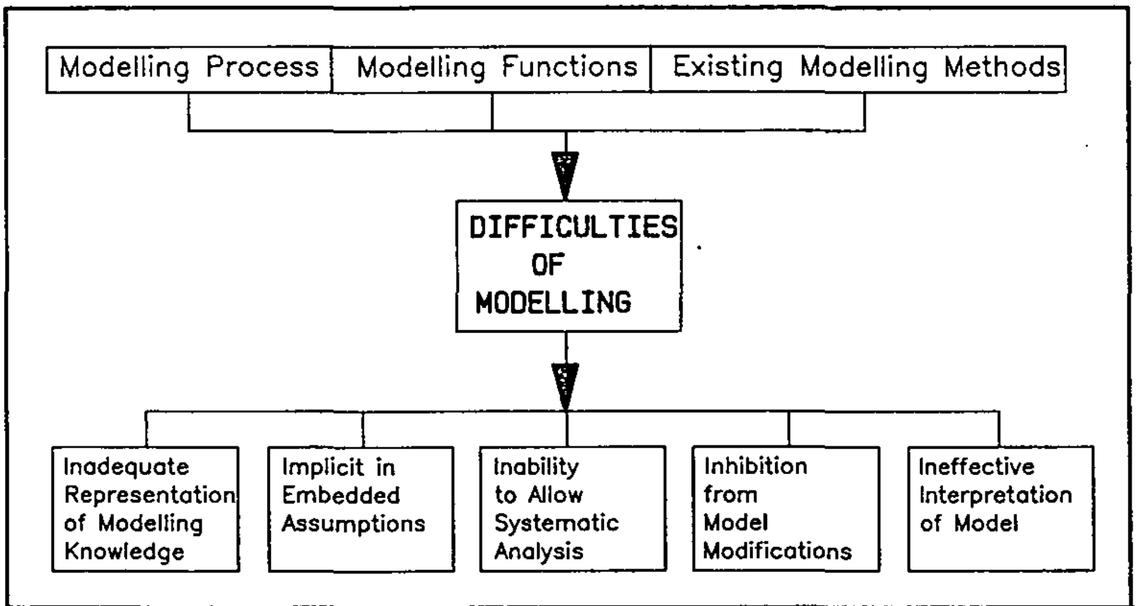


Figure 4.5 *Difficulties of Modelling in Flexible Manufacturing Design* LUT - FMS Research Group

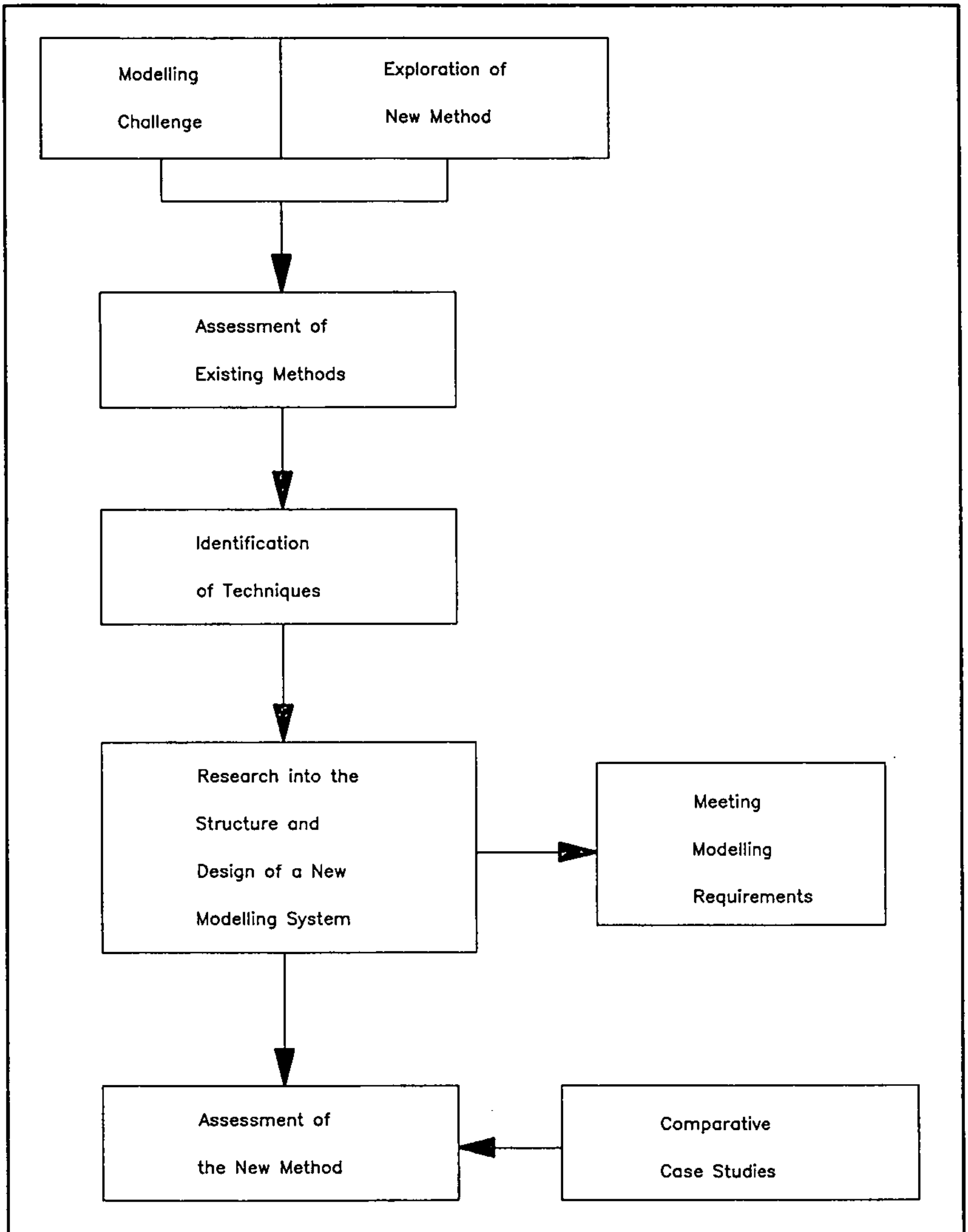


Figure 4.6

Framework for the Study

LUT-FMS Research Group

Chapter 5
COMPARATIVE ASSESSMENT OF MODELLING METHODS

5.1 Introduction

This chapter assesses the major modelling methods that are currently available for the analysis of flexible manufacturing facilities. A classification scheme is proposed for these methods along several dimensions and the criteria to be used in the assessment are outlined. The techniques to be discussed include static capacity analysis, queueing networks, Petri nets, simulation and perturbation analysis. Each of these techniques is examined and comparison among them is then conducted according to the identified criteria.

5.2 Classification of Modelling Methods

Models and the corresponding techniques can be classified with regards to different criteria. Solberg [60] classifies models in terms of the form, the system objective, the time nature and the variability of the model (Figure 5.1). Wilhelm and Sarin [295] and Looveren et al. [171] conduct classifications based on the various decisions that can be made by the models (Figure 5.2) and (Figure 5.3).

Drawn on the requirements of modelling for the design of advanced manufacturing systems, modelling methods can be classified along the following dimensions (Figure 5.4):

- ***Modelling Objectives:*** evaluative, hybrid evaluative and generative, and generative. The distinction is conducted according to the way models are used to deal with the system objective.

In considering the modelling methods for the design of manufacturing systems, it is vitally important, from the practical user's point of view, to distinguish between *generative* methods which *find* 'good' candidate decisions, and *evaluative* methods which *evaluate* a given set of dicisions. With generative models, certain variables of the system description are left unspecified initially and some specific algorithms are employed to determine what they should be; while with evaluative models, input parameters, control rules and the like are either built into the structure of the models or are taken as given. Although an experienced analyst might be clear about this distinction, models which are built with these different types of methods can well be mis-used by industrial engineers.

Essentially, models based on the generative methods can help to quickly resolve complex situations, but on the other hand they suffer from the 'black box' syndrome. Besides, they may not work well in some contexts and do not allow easy modification of decisions. These models also appear to remove the decision maker from the decision-making process and may be threatening. Typical examples are the static capacity analysis [165] [217], and the linear and non-linear programming techniques, such as those used by [254] [182]. For the purpose of designing a manufacturing system, these techniques may be used to determine the number of cell elements and the number of buffer spaces [34], and especially they can be very useful when financial factors are considered in the analysis [233].

Evaluative methods, in contrast, are more a tool to *help* the designer to make decisions by sharpening his intuition about the system, i.e. they provide insight rather than decisions. With evaluative models, usually decisions can be more easily modified by the user through trying out different input parameters. However, it may take a long time to find 'good' decisions using these models. Most design tools developed for advanced manufacturing systems are based on evaluative methods, such as the queueing network method [248] [249] [268], heuristic algorithms [84], semi-Markov process [237], the discrete event simulation method [43] [164] and the Petri nets method [52].

There are some important exceptions to these two methods. For example, evaluative models can be integrated with generative models, that is, the output from an evaluative model can be used to modify the decisions chosen by a generative model [292] [151] [184]. These types of methods may be classified as hybrid evaluative and generative methods . Another interesting exception is the purturbation analysis technique which generates directions for improving the existing decisions that are evaluated by a simulation model [132].

- ***Abstraction Level:*** structural, approximate and detailed. Here modelling methods vary in terms of the logical details which can be contained within the corresponding models.

A structural method can be used to design models from the static point of view. These models are useful for describing and formalizing manufacturing systems in terms of concepts and relationships. They can be used as a reference and a guide all along the design process. A typical example of structural modelling methods is the GRAI method [91]. Recently, research into the use of IDEF0 concepts for designing structural factory models is underway on the Xerox Workstation in close link with the work reported in this thesis [145a].

Approximate modelling methods may be defined as a class of tools which can be used to design models for providing a quick estimate of how a manufacturing system behaves and how its components interact, or provide decisions arising in the design of a manufacturing system. The basic requirements of these methods are the effectiveness of the system performance output provided and the efficiency associated with the modelling process. Methods of this class can either be static or dynamic. The static capacity analysis [164] [217], mathematical programming [254] and heuristic algorithms [84] [304] are the examples of static methods, whereas queueing networks [248] [268], Petri nets [175] [8], semi-Markov process [237] and simple simulation approaches [85] are dynamic methods.

Theoretically, every modelling method could be used to design a model containing the desired details. However, this is contrained by two major factors. One is the current status of the theory of the method, and another is the computational requirement. A detailed modelling method should be a tool which can be applied to develop detailed models with its current thoery and available computational vehicles. The current status of theory for a method is characterised by its capability to fully represent the structure and interactions of a manufacturing system. Therefore the queueing network is not a detailed method because its current thoery does not allow for the building of complex models. On the other hand, since some complex mathematical programming models [254] can not be solved easily, their application in detailed manufacturing system modelling is limited. Recently, conventional simulation techniques have been extended to allow modelling of manufacturing systems in full-scale details, and this method has been termed *emulation* [60] [43]. For some large-scale mathematical programming models [182], if the computational constraint can be resolved, they can also be classified as detailed methods.

- ***Modelling Formalisms:*** algorithm, graph, Markov chain, simulation techniques, and artificial intelligence. The categorisation is made based on the way in which manufacturing knowledge is represented.

An algorithmic approach [84] concentrates on the development of computer algorithms which deal with the scheduling of the chain of activities in a manufacturing system. Unlike the simulation approach, the decision making within these modelling algorithms is of a hierarchical structure. At each level, the start and finish times for particular activities of certain entities are determined, and this result is used as given parameters at the next level. Iwata [142] structured his scheduling model at three levels. At the first level, the parts' machining schedules are determined by selecting an appropriate machine tool from candidates for each machining operation and simultaneously determining the loading sequence of parts on each selected machine tool. The decision making at the second level is to determine the schedule of tool allocation and tool delivery by considering the tooling availability at each machine and tool provision from the central tool store. At the third level, transport devices are selected and scheduled for transferring parts between machines.

The graph approach uses graphic tools to describe and formalise a manufacturing system's operation. With this approach, graphic conventions have to be defined, analysis procedures be developed, and the modelling principles for manufacturing applications be formalised. A typical example is the Petri nets approach, in particular the timed Petri nets [52]. Although in its infancy, this approach has the potential of providing quantitative indicators with respect to the performance of a manufacturing system. There are also graphic tools developed for assisting building simulation models, such as the activity cycle diagram [65] and the network diagram [188]. These tools, though graphic in nature, can not be used to provide quantitative insight into the behaviour of a manufacturing system, and therefore they should not be classified as graphic modelling methods.

The Markov chain approach is based on probability theory, which represents manufacturing processes in terms of mathematical queueing network equations. Although robust in its underlying theory, models based on this approach have to make certain unrealistic assumptions, and normally they can only be used to study the steady state of a system. Major examples include all the queueing network models developed to date [248] [282] [87] [59] [239], the operational analysis approach [172a] and the semi-Markov process approach [237].

The simulation approach uses computer programs to imitate the system's dynamic behaviour by taking advantage of the high processing speed of a computer. Various conventions have been developed to allow for the writing of these programs, such as the event based approach, the activity based approach, the process based approach and the three phase approach [209]. Based on these approaches, many general-purpose simulation languages, generalised manufacturing system simulators and specific manufacturing system simulation models have become available [65] [64].

Artificial intelligence has come to be seeing as an alternative to conventional modelling approaches. It uses more advanced knowledge representation methods to describe the operation of a system so as to allow for full-scale modelling of different aspects of the system. The major differences between the conventional somulation approach and the AI approach are the following:

- Numeric versus symbolic knowledge representation,
- Explicit versus non-explicit solution procedures,
- Integrated data and control structure versus separated knowledge and control structure, and
- Modelling of entity flow versus modelling of information flow.

Typically, the AI approach uses frames, objects, rules, logics, etc.[46] to represent the structure and modelling knowledge of a system and the use of the modelling knowledge is organized around a general control structure.

5.3 Criteria for the Assessment

Different modelling methods have different characteristics and address different aspects of the problem, and thus they can be used for varying purposes. With regard to the requirements of the application of these modelling methods in design of flexible manufacturing systems, the following criteria are proposed as being pertinent in the overall contex of this assessment:

- Principal characteristics and modelling capabilities.
- Limitations of the method.
- Typical model inputs and outputs.
- Application experience.

- Further developments.

In the following sections, major evaluative modelling methods will be compared according to these criteria. These include static capacity analysis, queueing networks, Petri nets, simulatin and perturbation analysis.

5.4 Principal Characteristics and Modelling Capabilities

Static capacity analysis is a technique which simply adds up the total amount of work allocated to each resource, and estimates the performance from these totals, or computes the gross requirement for the resource. A common example is to add up the processing time of all operations assigned to a station in order to estimate its utilisation, or determine the minimum number of stations for each station group. Figure 5.5 shows a static capacity model which can be used to determine the station requirement, transporter requirement, pallet requirment and storage requirement. Models based on this technique are static and simple.

Queueing networks can be used to develop models which account for dynamics, interactions and uncertainties in the system, but in an aggregate way. Both the input data required and the output measures produced are average values which assume a steady state operation of the system. However, these models tend to give reasonable estimates of performance and are extremely efficient. They can model stations, buffer storages, simple control rules and system features like tool sharing and workpiece blocking [88] [92]. In addition these models require relatively small amount of input data and do not use much computer time. Figures 5.6 and 5.7 summarize the single pallet type queueing network model [248] [251]. A more powerful queueing network model is shown in Figures 5.8 and 5.9. It takes into account multiple pallet types and parts routing proportions, which heavily influence the performance of the system [268].

Figure 5.10 and Figure 5.11 illustrate the Petri net approach to system modelling and the basic concepts of timed Petri net models [52]. Petri nets are useful to model systems where behaviour can be described as interferences between asynchronous and concurrent processes. The current theory of Petri nets applied to flexible manufacturing systems permits a dynamic, deterministic model of the system. Timed Petri nets, in conjunction with certain modelling conventions, appear to be a quite useful modelling tool (Figure 5.12). In particular activities requiring many different resources, such as machine tool, AGV, robot and cutting tools, can be modelled (Figure 5.13). Due to their graphical nature, Petri nets give clear and legible models which facilitate the dialogue between designers and users.

The simulation method mimics the detailed operation of the system through a computer program. Four basic approaches are available for developing these type of models[209], i.e. the event based approach (Figure 5.14), the activity based approach (Figure 5.15), the process based approach (Figure 5.16) and the three phase approach (Figure 5.17). Depending on the amount of information that is built into a particular model, simulation has the potential of allowing as much detail as desired or necessary to mimic the reality. Simulation can and has been used for all problem types. At the advanced stage of the system design, simulation is very useful to get a precise view of the behaviour of the system as a function of various candidate scheduling and operating policies. More detailed questions can be analysed and answered and system parameters determined [166].

Perturbation analysis is a technique which can provide additional information to that normally provided by a simulation model. The basic idea is to observe the detailed behaviour of the system, whether through simulation or from the actual system, for one set of system parameters. By doing some additional calculations, this technique can predict the system behaviour from the initial observation if these parameters were changed (Figure 5.18). The important advantage of the technique is that it is not necessary to re-run the simulation or system with modified system parameters. Therefore, it is a useful tool for fine-tuning design decisions. Figure 5.19 illustrates the modelling process using such an approach.

Figure 5.20 summarises the major characteristics and capabilities of these methods.

5.5 Limitations of the Method

Since static capacity analysis ignores all dynamics, interactions and uncertainties which appear in real systems, the main drawback of the technique is that for more complex systems it can be much too coarse a tool and seriously overestimates system performance, leading to the inability to help to make realistic decisions.

Although optimistic results have been obtained from the comparison between queueing network models and simulation models, there are severe limitations within a queueing network model. First, certain assumptions (e.g. exponential service rates, FIFO queue, no station breakdowns, etc.) necessarily made in the model formulation are unrealistic for most of the systems [249] [251]. Secondly it is clear that this method is inherently unsuitable to answer many of the detailed design questions, such as the transient effects of infrequent but severe disruptions and the assessment of various control policies.

Because of the graphic nature, applications of Petri nets are potentially limited by the inefficiency after incorporating detailed system features, such as many machines with finite buffers and real-time routing policies. Although this problem has recently been alleviated by the advent of coloured Petri nets [8] [175], the techniques for model construction, analysis and realisation are not yet fully developed as a result of the tool's newness. In addition current models also do not consider any uncertainties.

While the simulation method is potentially a powerful tool for modelling, understanding and designing advanced manufacturing systems, large-scale simulators do not provide the capabilities necessary to allow simulation to achieve its potential. First present simulators are justified by their inability to verify the completeness and accuracy of the models. Secondly models embedded in simulations can not be easily modified, and thus simulation may inhibit rather than promote easy changes required to construct alternative models. Thirdly contemporary simulation techniques fall well short on the dimension requiring comprehensibility of the results. Furthermore, although simulation models can be made very detailed, the price has to be paid is in terms of the programming time to create the model, the input time to generate detailed data requirements, and the computer time each time the model is run [178].

The main disadvantage of perturbation analysis is that it currently can not predict the effects of large changes in parameters. For example, the addition of a new machine tool can not be analysed using this technique. Therefore, it is of limited value for the evaluation of preliminary designs.

Figure 5.21 summarises the major limitations of these methods.

5.6 Typical Model Inputs and Outputs

The data input requirements for static capacity analysis are very simple and typically include production quantities, the system description, the planning horizon, and part routes. Major outputs that can be produced are the minimum number of machines needed, the expected system utilisation, and and the required number of pallets, etc. This can be seen in Figure 5.5.

A queueing network model requires quite simple data items to be input. They are number of pallets within the system, number of station groups and stations at each group, number of transporters, pallets' visiting frequencies to a station group and the probability of the use of transporters, and average station processing times and the average transport time [248]. The typical outputs include the average steady-state expected production rate, mean queue lengths, and mean machine utilisation figures [251]. Figure 5.22 shows the layout of the benchmark manufacturing cell used by the Emulator project [43]. The input information to the multiple pallet type queueing network model for this cell is shown in Figures 5.23 and 5.24 in the format which is used by the model. The output results are depicted in Figure 5.25.

Depending on the amount of information that is built into a particular simulation model, the data inputs and outputs can vary for different models. Basically the input information requirements are part data, machine tool data, transporter data, process data and control strategies. The outputs are statistics on the throughput of parts, machine and transporter utilisation, and part performance figures. Figure 5.26 illustrates the input and output requirements from the Emulator at Loughborough University [43].

Basically the input and output information requirements for Petri net models are similar to those of simple simulation models [52]. Whether this approach can provide additional information will depend on the development of the technique as a result of the tool's novelty.

Since perturbation analysis is performed based on a simulation model, it has the similar input and output requirements to those of a simulation model, i.e., it can provide performance measures for the modelled system. In addition, it is capable of generating directions for improving the existing decisions. For example, the technique can be used to study finite buffer situations to help determine the suitable size of the buffers.

Comparison of static capacity analysis, queueing network and simulation with regards to production rate is shown in Figure 5.27 [31]. Rathmill [218] compares these modelling methods according to the logic details which can be built within the models (Figure 5.28).

5.7 Application Experience

Static capacity analysis has been used to study the feasibility of an preliminary design and can help to quickly screen out many decisions. For instance, for a given design alternative, if even the maximum performance produced from the analysis is not accepted, it is not necessary to evaluate that alternative with a more detailed model [165].

Queueing network models can provide more realistic performance estimates than static capacity models. They can in general provide approximate indications of the adequacy of particular systems, which may be sufficient as a preliminary solution. These models are becoming popular in manufacturing system design and operation as a good compromise between the efficiency of the model and the accuracy of the predictions. They have shown to be especially useful in situations where management requires quick turnaround on initial designs [43].

At present, simulation is perhaps the most widely used computer based tool for performance evaluation of advanced manufacturing systems. It plays a crucial role in the successful implementation of the system and thus is well recommended after the use of static capacity models or queueing network models.

While the application of Petri nets in the modelling of flexible manufacturing has not been well established, Perturbation analysis has been used to help a system manager to improve his decisions without having to experiment on the actual system [266]. It is useful for fine-tuning design decisions, and therefore can be conducted after simulation experiment.

5.8 Further Developments

Although simple in nature, it can be foreseen that static capacity analysis would be extended to answer many other questions, such as tool management [50] and determination of WIP storages, as a result of the ease of its implementation.

Since the advent of the CAN-Q model, there has been considerable interest in the development of queueing network models. Future developments of these type of models would include the incorporation of various control strategies, multiple pallet types, tool management, workpiece blocking and uncertainties. Additionally queueing network models for multi-cell flexible manufacturing will be required as a result of the development of the manufacturing technology [81]. However, from the viewpoint of practical users, further development of these type of models is inhibited by the requirement of expertise in the field.

Due to their unique graphic nature Petri nets have the potential for realistic applications in flexible manufacturing. But the current theory needs to be further developed, and it seems necessary to computerise the use of the tool in order to solve large-scale problems.

In recent years computer simulation has been widely accepted as a necessary tool to succeed in designing a flexible manufacturing system, that will actually operate the way it is designed to do. However, so far this very complex design task has been made even more complex because of the complexity of just using the various simulation tools existing on the market. Therefore, there exists the need to develop user-friendly simulation packages that will reduce the complexity of this design task [64]. Additionally it is necessary to develop techniques, such as parallel processing [227], to increase the speed of execution of large-scale manufacturing simulation models. Rathmill [218] has concluded that there is a trend towards developing data driven generalised manufacturing simulators (Figure 5.29).

As a result of the novelty of the technique, there have not been many perturbation analysis models developed for the design of manufacturing systems. Hence, formalisation of the technique and the way it can actually be applied to impact system design need to be further investigated.

5.9 Conclusions

From the above assessment it can be concluded that each of the major currently available modelling methods has certain disadvantages when applied to manufacturing system modelling. Static capacity analysis is too simple to significantly influence the system design. The application of queueing networks, Perti nets and perturbation analysis is mainly constrained by the immaturity of the techniques themselves. Simulation is the most useful and widely applied method but there are certain limitations associated with the currently established techniques.

Recently, AI has come to be seen as an alternative to conventional methods in most of the fields. Shortly after the start of this work, the department aquired the Xerox Workstation and LOOPS knowledge engineering environment. All these have made it a valuable and possible research direction to explore the application of AI techniques in the modelling of advanced manufacturing system.

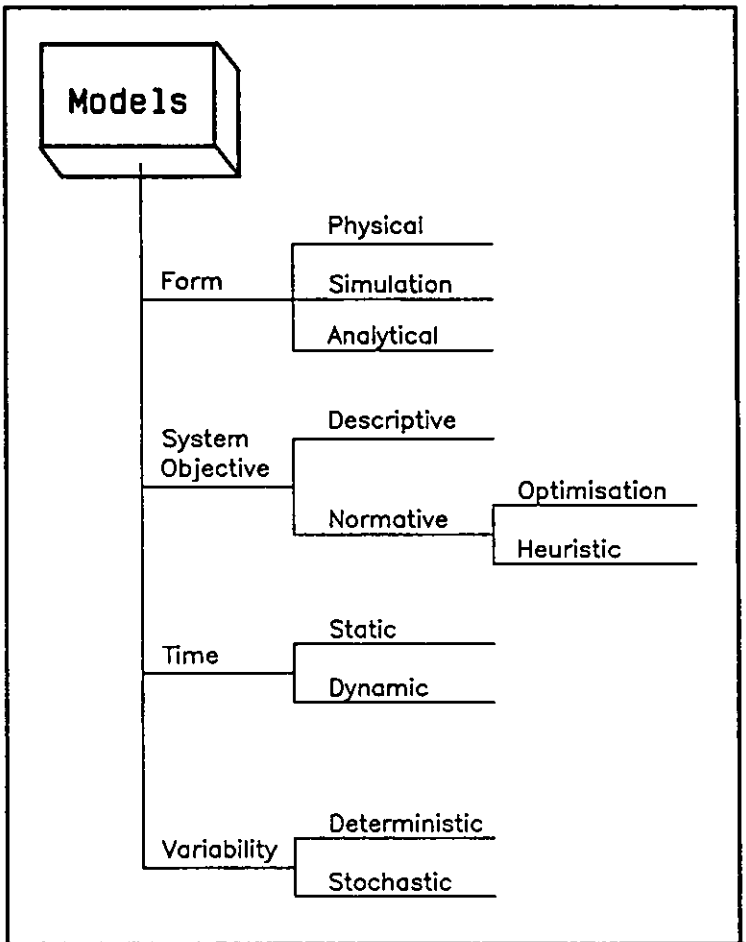


Figure 5.1	*Classification of Models - View 1*	LUT-FMS Research Group

Problem Category	Queueing Networks	Simulation	Mathematical Programming	New Methods	Markov Models
FACILITIES DESIGN					
Machines	*				
Transporters	*	*		*	
Fixtures	*	*			
Tooling		*			
Part Selection	*	*			
WIP Storage	*	*		*	*
Layout		*			
INTERMEDIATE RANGE					
Part Routing			*		
Machine Tooling		*	*		
DYNAMIC OPERATION					
Release Rules	*	*			
Breakdowns		*	*	*	*
Scheduling		*	*	*	

New methods include: Operational analysis, Perturbation analysis & Mean value analysis

Figure 5.2	*Classification of Models - View 2*	LUT - FMS Research Group

Technique / Problem	Queueing Networks	Mathematical Programming	Simulation Analysis	Others
Screening	*		*	Perturbation Petri nets
Selection		*	*	Multi-Criteria
Batching		*	*	
Loading	*	*	*	
Releasing		*	*	Control theory
Dispatching	*	*	*	Control theory

Figure 5.3	*Classification of Models - View 3*	**LUT - FMS Research Group**

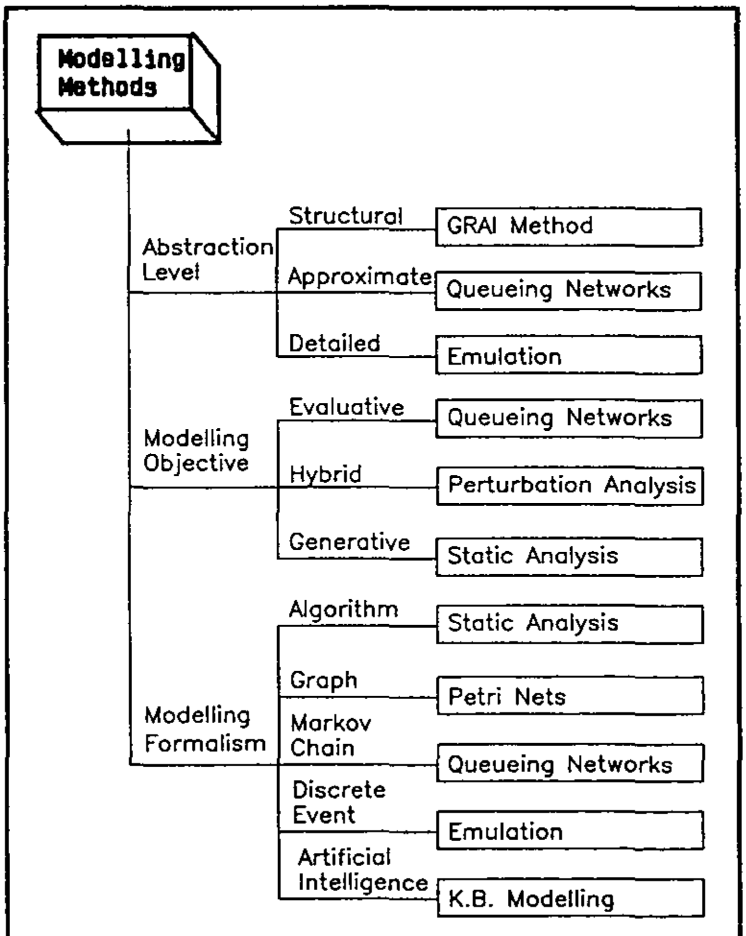


Figure 5.4	*Classification of Modelling Methods*	**LUT-FMS Research Group**

Notation:

$p \in \{1, 2, ..., P\}$ part types,

$s \in \{1, 2, ..., S\}$ station groups,

$o^{ps} \in \{0, 1, ..., O^{ps}\}$ operations of part p to be done on station s,

BS_p batch size of part p,

ET_s pallet exchange time at station s,

TT average transport time between stations,

WT_p total waiting time of part p in a cycle,

PH planning horizon,

$t^{ps}_{o^{ps}}$ processing time of operation o^{ps} of part p on station s,

COT_s cummulative operation time of station s,

NS_s number of stations in station group s,

CTT cummulative transport time,

NT number of transporters,

CT_p cycle time of part p,

NP number of pallets,

NB number of buffer spaces.

The Model:

$$COT_s = \sum_{p=1}^{P} BS_p * \sum_{o^{ps}=0}^{O^{ps}} (t^{ps}_{o^{ps}} + ET_s) \quad \forall s,$$

$$NS_s = COT / PH \quad \forall s,$$

$$CTT = TT(\sum_{p=1}^{P} BS_p * \sum_{s=1}^{S} O^{ps} - \sum_{p=1}^{P} BS_p),$$

$$NT = CTT / PH,$$

$$CT_p = WT_p + \sum_{o^{ps}=0}^{O^{ps}} t^{ps}_{o^{ps}} \quad \forall p,$$

$$NP = \sum_{p=1}^{P} BS_p * CT_p / PH,$$

$$NB = Max(0, NP - \sum_{s=1}^{S} NS_s - NT).$$

Figure 5.5	*The Static Capacity Analysis Model*	LUT-FMS **Research Group**

Notation:

- N total number of pallets,
- $i \in \{1, 2, \ldots, M\}$ station group number, and M represents the transporter group,
- s_i number of stations in group i,
- t_i average processing time of an operation by a station in group i,
- v_i total number of visits of parts to group i,
- $p \in \{1, 2, \ldots, P\}$ part type,
- $o^{pi} \in \{0, 1, \ldots, O^{pi}\}$ operation os part p on station i,
- BS_p batch size of part p,
- α normalising constant,
- q_i visit frequency of parts to station group i,
- w_i work load assigned to group i,
- P production rate,
- T average flow time,
- u_i average utilisation per station of group i,
- l_i average queue length at group i, including part waiting and in
- lq_i average number of pallets in queue at group i,
- d_i idleness of group i,
- $G(M, N)$ normalising constant,
- n_i number of pallets at group i at some system state,
- $\mathbf{n} = (n_1, n_2, \ldots, n_M)$ state of the system, and $\sum_i n_i = N$,

The Model:

$$v_i = \begin{cases} \sum_{p=1}^{P} (BS_p * O^{pi}) & \forall i \wedge i \neq M, \\ \sum_{p=1}^{P} BS_p * (\sum_{i=1}^{M-1} O^{pi} - 1) & i = M, \end{cases}$$

$$\alpha = 1 / \sum_{i=1}^{M} (v_i / \sum_{i=1}^{M} v_i) \quad \forall i,$$

$$q_i = \alpha * v_i / \sum_{i=1}^{M} v_i \quad \forall i,$$

$$P = q_M / t_M * G(M, N-1) / G(M, N),$$

$$T = Nt_M / q_M * G(M, N) / G(M, N-1),$$

$$u_i = \begin{cases} q_i t_i / t_M s_i * G(M, N-1) / G(M, N) & \forall i \wedge i \neq M, \\ 1 / s_M * G(M, N-1) / G(M, N) & i = M, \end{cases}$$

Figure 5.6	*The Single Pallet Type Queueing Network Model (1)*	LUT-FMS Research Group

$$
l_i = \begin{cases} \sum_{k=1}^{N} (q_i t_i / t_M)^k * G(M, N-k) / G(M, N) & \forall i \in \{1, 2, \ldots, M-1\} \wedge s_i = 1, \\ \sum_{k=1}^{N} k / k! * G(M-1, N-k) / G(M, N) & i = M \wedge k \le s_M, \\ \sum_{k=1}^{N} k / s_M ! s_M^{(k-s_M)} * G(M-1, N-k) / G(M, N) & i = M \wedge k > s_M . \end{cases}
$$

$$
lq_i = \begin{cases} l_i - u_i & \forall i \in \{1, 2, \ldots, M-1\} \wedge s_i = 1, \\ l_M - u_M * s_M & i = M. \end{cases}
$$

$$
d_i = \begin{cases} 1 - u_i & \forall i \in \{1, 2, \ldots, M-1\} \wedge s_i = 1, \\ G(M-1, N) / G(M, N) & i = M. \end{cases}
$$

$$
G(M, N) = \sum_{\substack{\mathbf{n} \\ n + n + \ldots + n = N}} \prod_{i=1}^{M} f_i(n_i),
$$

where

$$
f_i(n_i) = \begin{cases} w_i^{n_i} / n_i! & n_i \le s_i, \\ w_i^{n_i} / s_i! * s_i^{n_i - s_i} & n_i > s_i, \end{cases} \qquad w_i = q_i * t_i \wedge \forall i.
$$

Recursive Computation of G(M, N):

$$
G(m, n) = \begin{cases} \sum_{k=0}^{n} (q_m t_m / t_M)^k / k! * G(m-1, n-k) & m \ne M \wedge k \le s_m, \\ \sum_{k=0}^{n} (q_m t_m / t_M)^k / s_m! \, s_m^{(k-s_m)} * G(m-1, n-k) & m \ne M \wedge k > s_m, \\ \sum_{k=0}^{n} G(M-1, n-k) / k! & m = M \wedge k \le s_M, \\ \sum_{k=0}^{n} G(M-1, n-k) / s_M! \, s_M^{(k-s_M)} & m = M \wedge k > s_M. \end{cases}
$$

where

$$
G(m, 0) = 1 \qquad \forall m,
$$

$$
G(1, n) = (q_1 t_1 / t_M)^n \qquad \forall n.
$$

Figure 5.7	*The Single Pallet Type Queueing Network Model (2)*	LUT-FMS **Research Group**

Notation:

$p \in \{1, 2, ..., P\}$ part type,

$i \in \{1, 2, ..., M\}$ station group number, and M represents the transporter group,

s_i number of stations in group i,

N(p) number of pallets for part type p,

$r(p) \in \{1, 2, ..., R(p)\}$ route number of part p,

$o^{r(p)i}_{r(p)i} \in \{0, 1, ..., O^{r(p)i}\}$ operation of part p at station group i in route r(p),

$t^{r(p)i}_{o^{r(p)i}}$ processing time of operation $o^{r(p)i}$,

$rp^{r(p)i}_{o^{r(p)i}}$ routing proportion of operation $o^{r(p)i}$,

α approximation factor for station pooling,

V(p, i) mean number of visits to group i by part p,

T(p, i) mean processing time of part p at station group i,

R(p, i) mean response time (waiting + processing) of part p at station group i,

Q(p, i) mean queue length at station group i for part p jobs (including jobs in process),

QNEW(p, i) new mean queue length at station group i for part p jobs,

w(p, i) number of part p jobs waiting at group i,

b(p, i) number of part p jobs in process at group i,

W(i) number of jobs waiting at group i,

B(i) number of jobs in process at group i,

Q(i) mean queue length at station group i,

u(p, i) utilisation of station group i by part p jobs,

U(i) utilisation of station group i.

The Model:

$$V(p, i) = \sum_{r(p)=1}^{R(p)} \sum_{o^{r(p)i}=0}^{O^{r(p)i}} rp^{r(p)i}_{o^{r(p)i}} \qquad \forall p \wedge \forall i.$$

$$T(p, i) = \begin{cases} 0 & V(p, i) = 0, \\ \sum_{r(p)=1}^{R(p)} \sum_{o^{r(p)i}=0}^{O^{r(p)i}} t^{r(p)i}_{o^{r(p)i}} & V(p, i) \leq 1, \\ \sum_{r(p)=1}^{R(p)} \sum_{o^{r(p)i}=0}^{O^{r(p)i}} t^{r(p)i}_{o^{r(p)i}} \Big/ V(p, i) & V(p, i) > 1. \end{cases}$$

$$Q(p, i) = N(p) / M \qquad \forall p \wedge \forall i.$$

Figure 5.8	*The Multiple Pallet Type Mean Value Analysis Model (1)*	LUT-FMS Research Group

$$R(p,\ i) = T(p,\ i) + \ /s * [(N(p) - 1) / N(p) * Q(p,\ i)T(p,\ i) + \sum_{k \neq p} Q(k,\ i)T(k,\ i)]$$

$$\forall\ p \wedge \forall\ i,$$

$$X(p) = N(p) / \sum_{i=1}^{M} V(p,\ i)R(p,\ i) \qquad \forall\ p,$$

$$QNEW(p,\ i) = V(p,\ i)R(p,\ i)X(p) \qquad \forall\ p \wedge \forall\ i,$$

$$u(p,\ i) = V(p,\ i)T(p,\ i)X(p) \qquad \forall\ p \wedge \forall\ i,$$

$$U(i) = \sum_{p=1}^{P} u(p,\ i) \qquad \forall\ i,$$

$$Q(i) = \sum_{p=1}^{P} QNEW(p,\ i) \qquad \forall\ i,$$

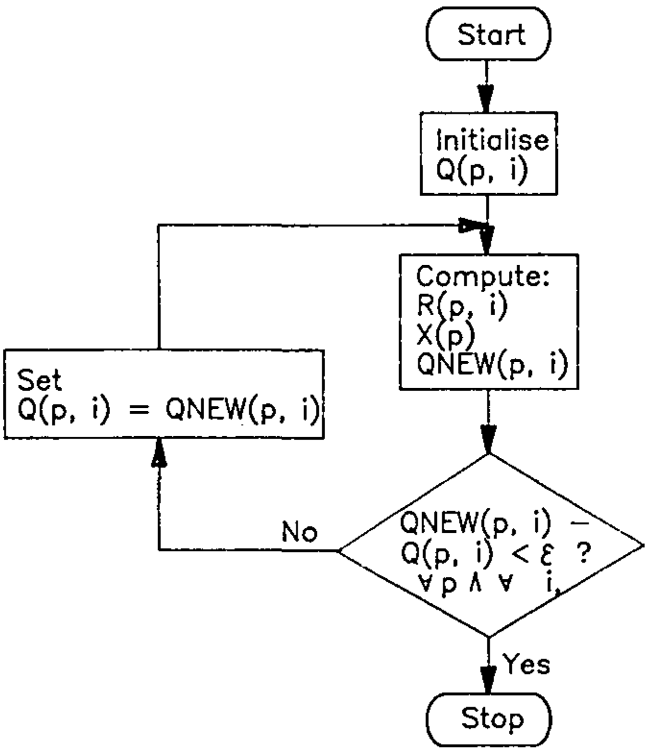
$$w(p,\ i) = [R(p,\ i) - T(p,\ i)] / R(p,\ i) * QNEW(p,\ i)$$

$$W(i) = \sum_{p=1}^{P} w(p,\ i) \qquad \forall\ i,$$

$$b(p,\ i) = T(p,\ i) / R(p,\ i) * QNEW(p,\ i)$$

$$B(i) = \sum_{p=1}^{P} b(p,\ i) \qquad \forall\ i.$$

Flow Chart for the Computer Program:



where ε is the computation precision factor.

Figure 5.9	*The Multiple Pallet Type Mean Value Analysis Model (2)*	LUT-FMS Research Group

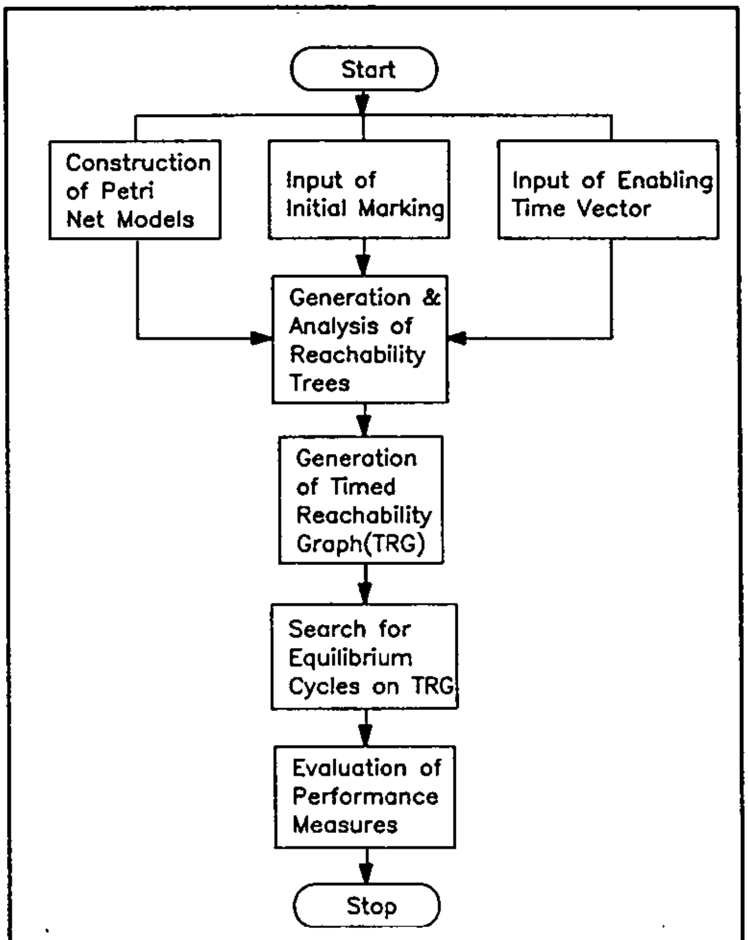


Figure 5.10	*A Perti Net Approach to Manufacturing Modelling*	LUT-FMS **Research Group**

Timed Petri Net (TPN) = (P, T, I, O, M, D) where:

- $P = \{p_1, p_2, \ldots, p_m\}$ set of places,
- $T = \{t_1, t_2, \ldots, t_n\}$ set of transitions,
- $I: (P * T) \rightarrow N$ input function that defines directed arcs from places to transitions (N is a set of all non-negative integers),
- $O: (P * T) \rightarrow N$ output function that defines directed arcs from transitions to places,
- $M: P \rightarrow N$ marking, i.e. labelling of the elements of P by non-negative integers,
- $D = \{d_1, d_2, \ldots, d_n\}$ enabling time vector.

State of TPN = S(M, R) where:

- M a marking,
- $R = \{r_1, r_2, \ldots, r_n\}$ remaining firing time vector.

Figure 5.11	*Concepts of Timed Petri Net Models*	LUT - FMS **Research Group**

(1) Transitions represent the set of activities to be performed in the system.

(2) Durations of the activities are represented by the enabling time vector.

(3) Input places of a particular transition indicate the conditions or resources or buffers associated with the firing of the activity.

(4) Output places specify the activities that are required next and the release of of certain resources.

(5) Tokens of various types represent available resources and parts which flow through activities according to the system control rules.

Figure 5.12	*Principles of Manufacturing Modelling by Petri Nets*	LUT - FMS **Research Group**

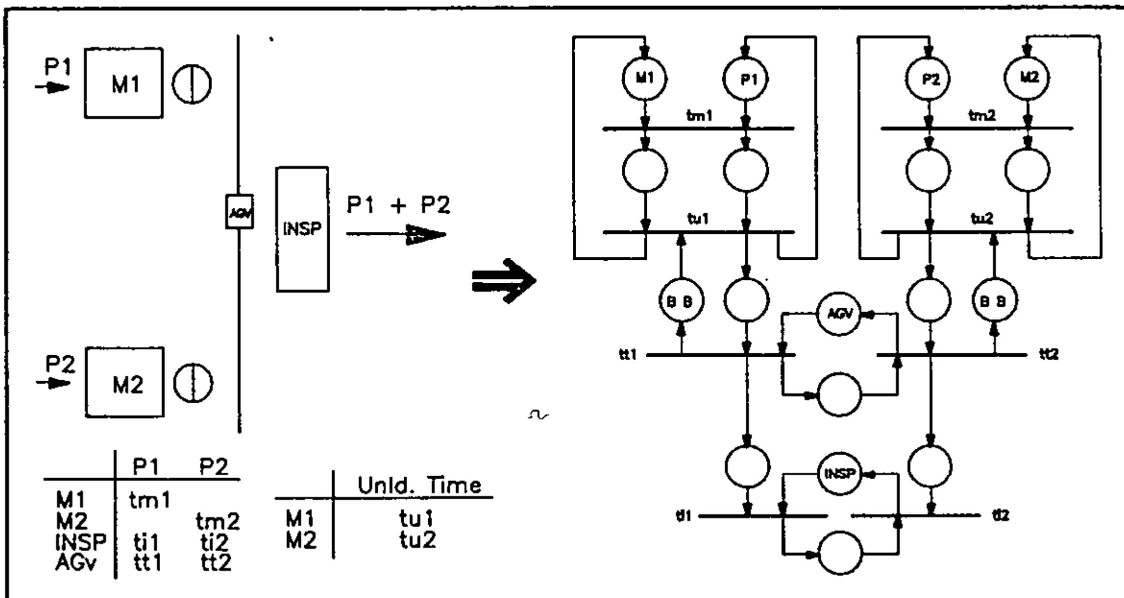


Figure 5.13	*Petri Net Modelling of a Sample Manufacturing Cell*	LUT - FMS **Research Group**

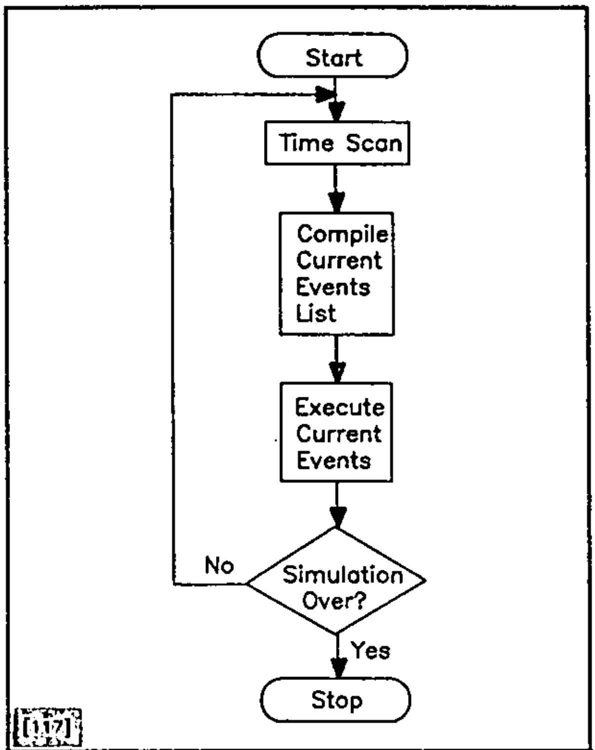


Figure 5.14	*The Event Based Approach*	LUT-FMS **Research Group**

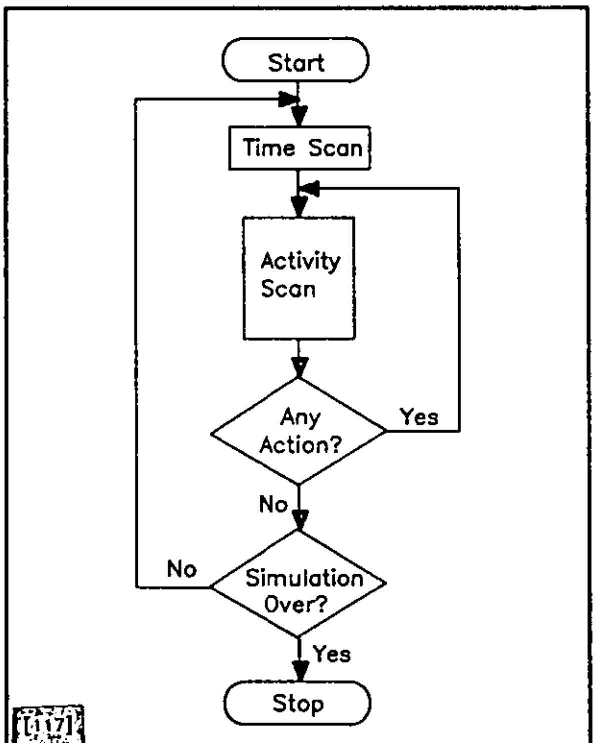


Figure 5.15	*The Activity Based Approach*	LUT-FMS **Research Group**

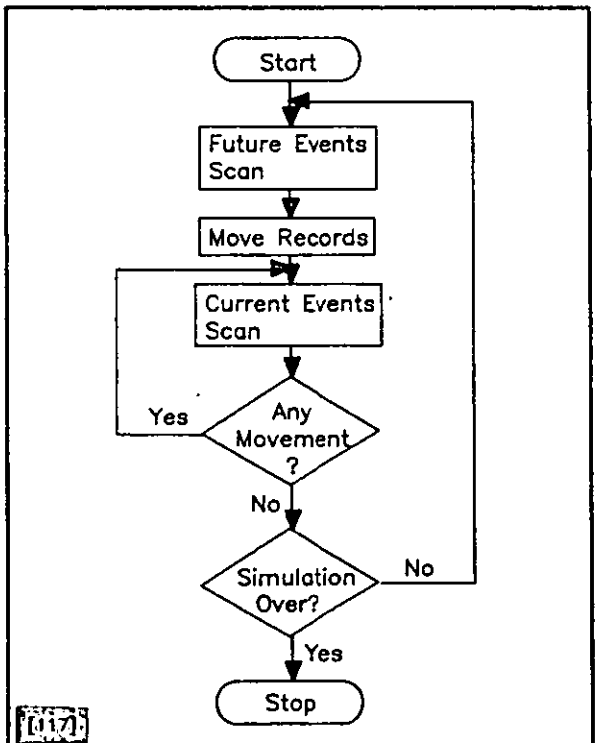


Figure 5.16	*The Process Based Approach*	LUT-FMS **Research Group**

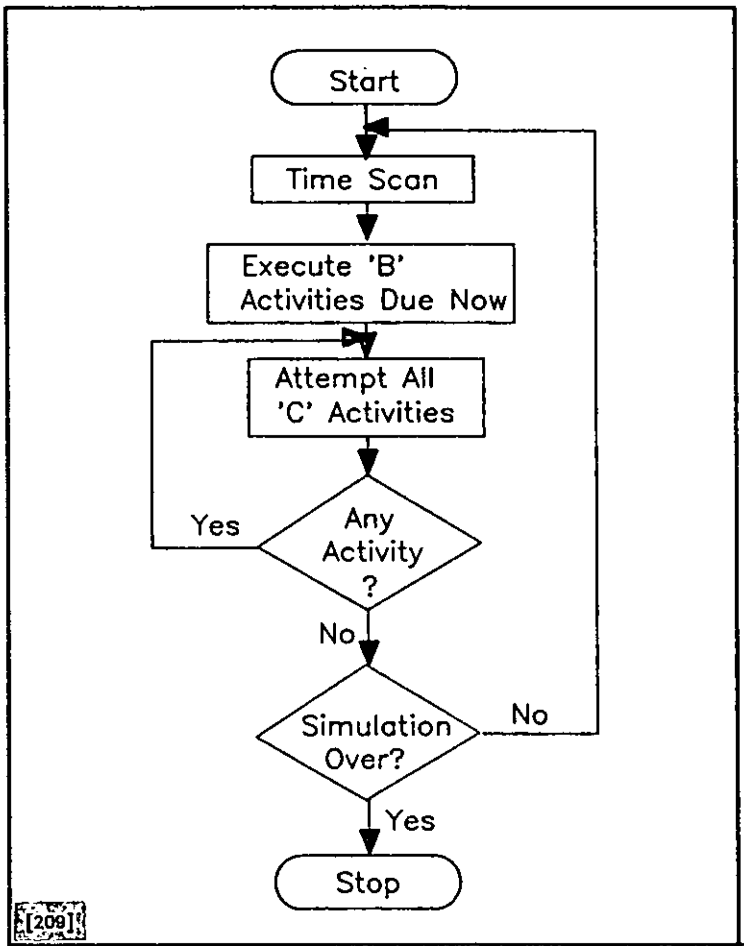


Figure 5.17	*The Three Phase Approach*	LUT-FMS **Research Group**

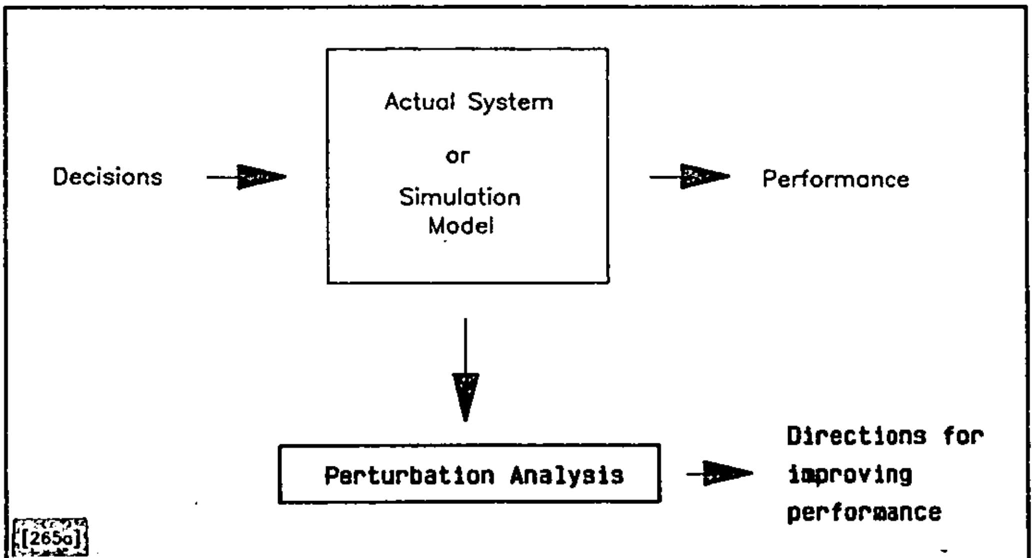


Figure 5.18	*Overview of the Perturbation Analysis Method*	LUT - FMS **Research Group**

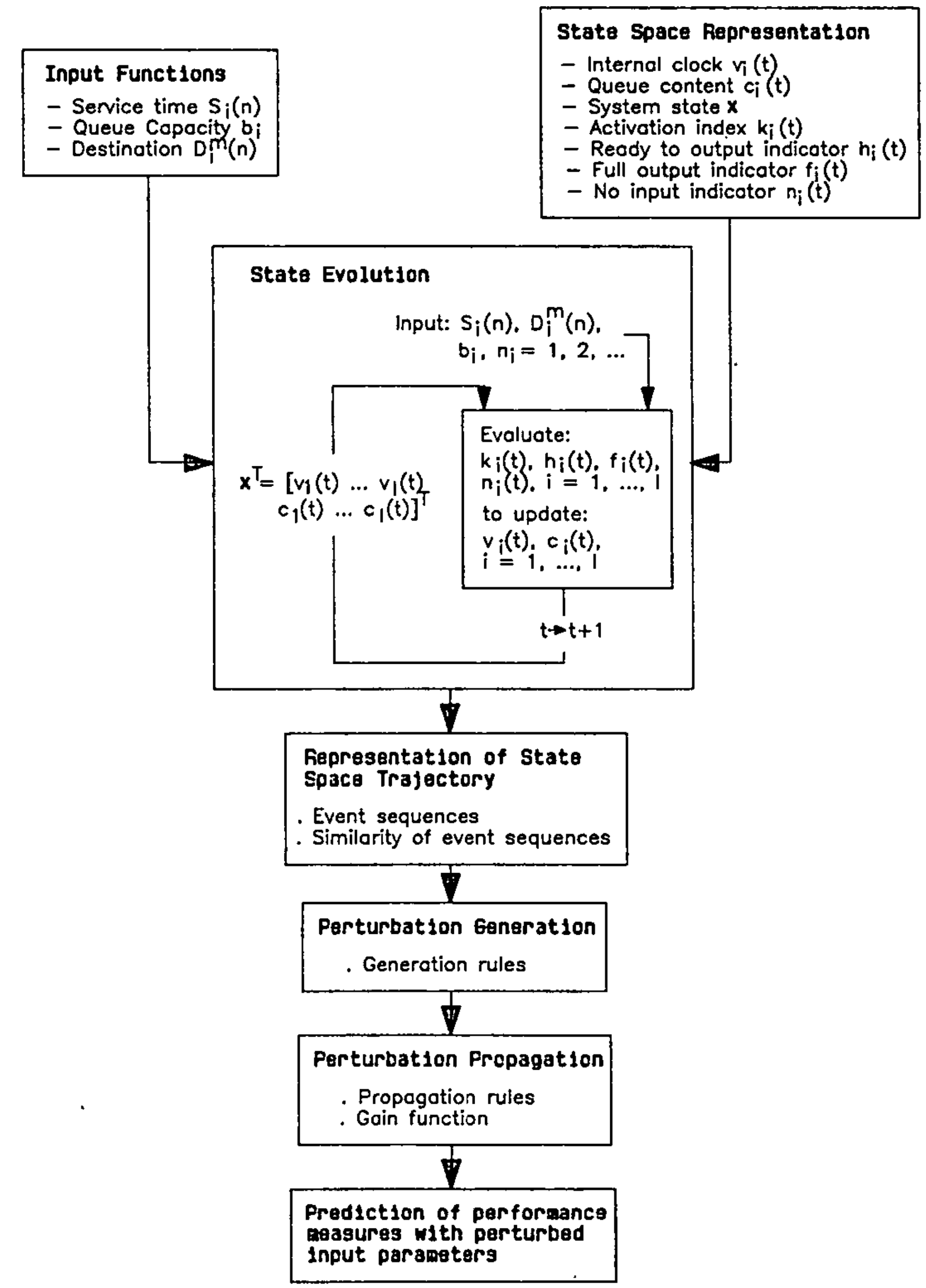


Figure 5.19

Macro Process of Perturbation Analysis

LUT-FMS Research Group

Method	Characteristics	Capabilities
Static Analysis	Static and simple; Easy to implement	Feasibility/Sizing
Queueing Networks	Aggregate and dynamic; Steady state	Design/Operation 'Ballpark' decisions
Petri Nets	Deterministic and dynamic	Design/Operation
Simulation	Detailed and dynamic	Design/Operation Make detailed decisions
Perturbation Analysis	Actual dynamics plus analysis; efficient	Design/Operation Fine–tuning

Figure 5.20	*Characteristics and Capabilities of Existing Modelling Methods*	LUT - FMS Research Group

Method	Limitations
Static Analysis	Ignore dynamics, interactions and uncertainties; Overestimation of system performance
Queueing Networks	Unrealistic assumptions, unsuitable for detailed decisions
Petri Nets	Inefficient for large–scale complex systems; Not fully developed, no uncertainties
Simulation	Time consuming to develop, debug and run; Difficult to modify
Perturbation Analysis	Unsuitable to analyze large changes in system parameters

Figure 5.21	*Limitations of Currently Available Modelling Methods*	LUT - FMS Research Group

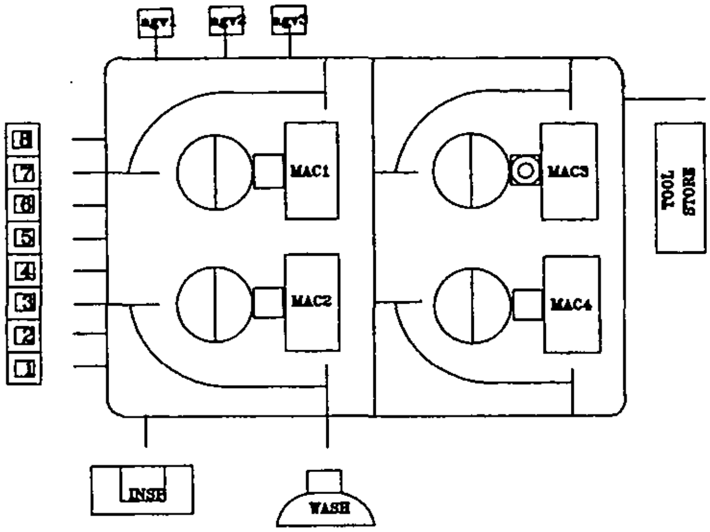


Figure 5.22	*Benchmark Cell*	LUT - FMS Research Group

```
****** MODELLING OF THE BENCHMARK CELL ******
          BASED ON THEORY OF MVA

**********************************************
        ***** System Size ****

No. of Part Types          No. of Stn. Groups
       8                          8

     ***** Calculation Requirements ****

Calculation Precision     Calculation Coefficient
      0.001                       1.38

   ***** Machine Group Information *******

Machine Group No.          No. of Machines
       1                          8
       2                          1
       3                          1
       4                          1
       5                          1
       6                          1
       7                          1

      **** Transport Information ******

Machine Group No.             No. of Carts
       8                          3

**** Total No. of Pallets/Fixtures for Each Part Type ****

Part Type No.            No. of Pallets/Fixtures
       1                          1
       2                          1
       3                          1
       4                          1
       5                          1
       6                          1
       7                          1
       8                          1
```

Figure 5.23	*Input to the Multi-Pallet Queueing Model - System Elements*	LUT-FMS Research Group

***** Part Routing Information *****

Part Type No.	No. Routes	Route No.	No. Ops.	Op. No.	Mach. Group No.	Proc. Time	Rout. Propn.
1	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	2	12.6	1
				4	8	1.0	1
				5	6	2.0	1
				6	8	1.0	1
				7	1	2.0	1
2	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	2	9.2	1
				4	8	1.0	1
				5	7	1.0	1
				6	8	1.0	1
				7	1	2.0	1
3	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	3	7.3	1
				4	8	1.0	1
				5	6	3.0	1
				6	8	1.0	1
				7	1	2.0	1
4	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	3	10.0	1
				4	8	1.0	1
				5	7	0.6	1
				6	8	1.0	1
				7	1	2.0	1
5	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	4	13.2	1
				4	8	1.0	1
				5	6	2.0	1
				6	8	1.0	1
				7	1	2.0	1
6	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	4	6.7	1
				4	8	1.0	1
				5	7	1.0	1
				6	8	1.0	1
				7	1	2.0	1
7	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	5	21.7	1
				4	8	1.0	1
				5	6	2.0	1
				6	8	1.0	1
				7	1	2.0	1
8	1	1	7	1	1	2.0	1
				2	8	1.0	1
				3	5	13.8	1
				4	8	1.0	1
				5	7	1.0	1
				6	8	1.0	1
				7	1	2.0	1

Figure 5.24	*Input to the Multi-Pallet Queueing Model - Part Routing*	LUT-FMS Research Group

SYSTEM PERFORMANCE MEASURES

*** PRODUCTION RATE ***

Part No	Parts per shift(8 Hours)
1	18.03
2	18.01
3	21.78
4	18.94
5	17.18
6	21.39
7	12.59
8	12.91

Total Parts/Shift = 140.83

AVERAGE TIME IN SYSTEM

Part No	Time(Minutes)
1	26.62
2	26.65
3	22.04
4	25.34
5	27.94
6	22.44
7	38.12
8	37.19

******UTILIZATION******

Machine Group	Machine Util.
1	0.15
2	0.82
3	0.73
4	0.77
5	0.94
6	0.34
7	0.13
8	0.29

AVERAGE QUEUE LENGTH (AT MACHINE GROUP)

Mchine Group No	No of Parts in Process	No of Parts Waiting
1	1.17	0.22
2	0.82	0.35
3	0.73	0.27
4	0.77	0.31
5	0.94	0.44
6	0.34	0.18
7	0.13	0.05
8	0.88	0.41

Figure 5.25 *Output from the Multi-Pallet Queueing Model*

LUT-FMS Research Group

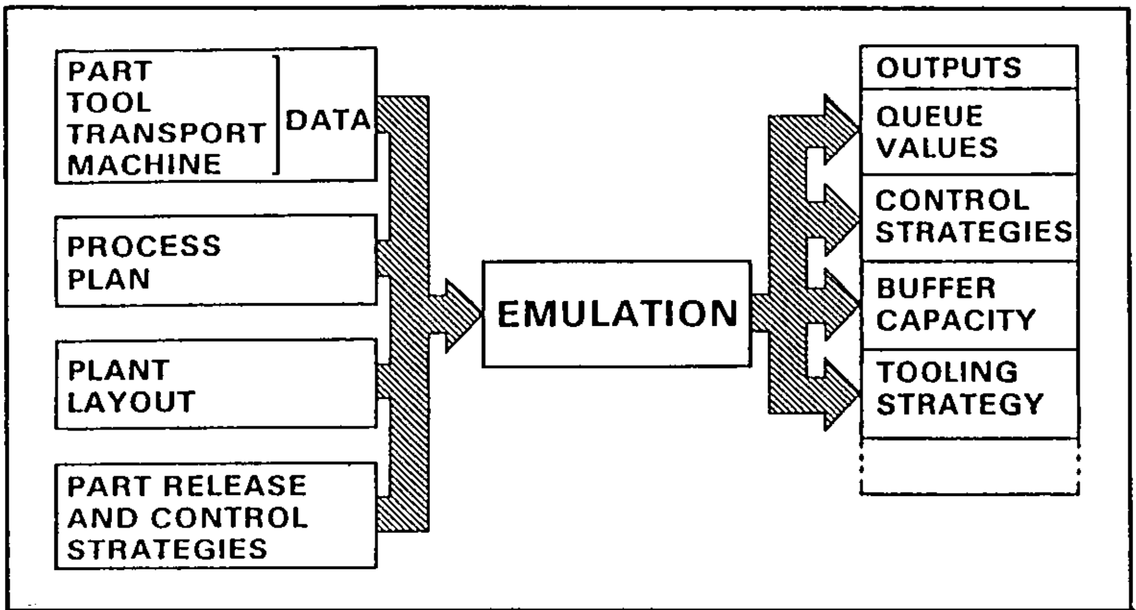


Figure 5.26	*Input and Output Requirements of the Emulator*	LUT - FMS Research Group

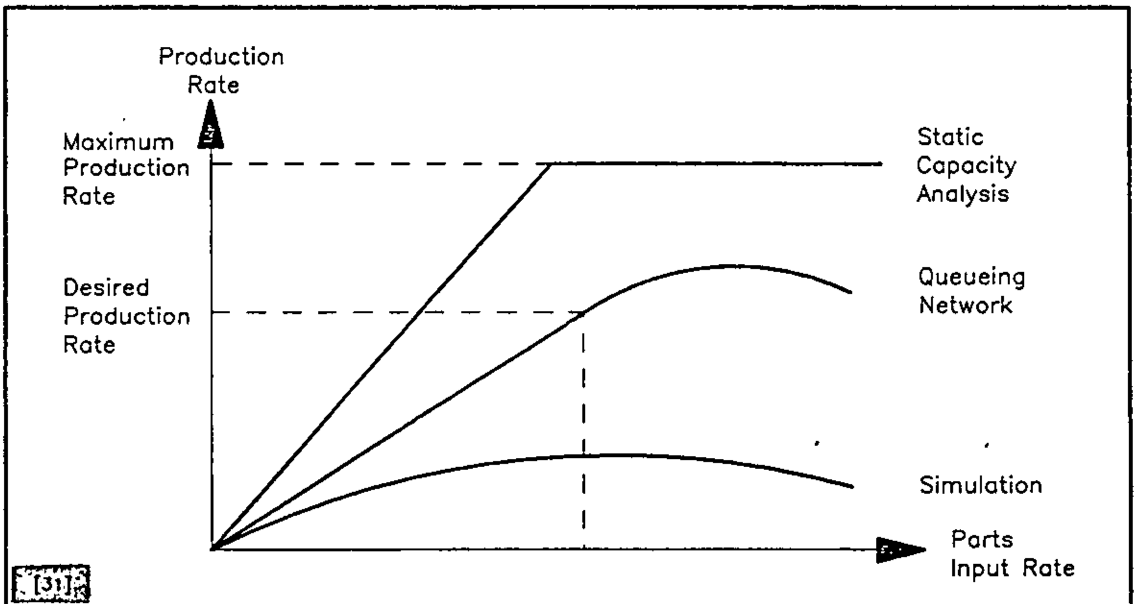


Figure 5.27	*Comparison of Modelling Methods w.r.t. Production Rate*	LUT - FMS Research Group

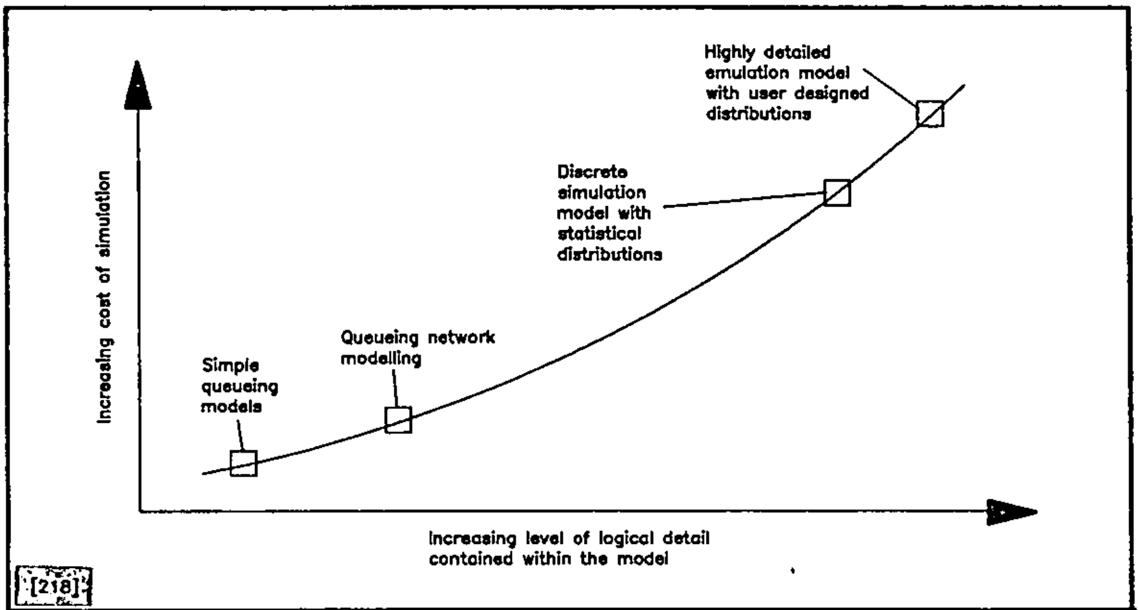


Figure 5.28	*Comparison of Modelling Tools w.r.t Level of Contained Logic Detail*	LUT - FMS Research Group

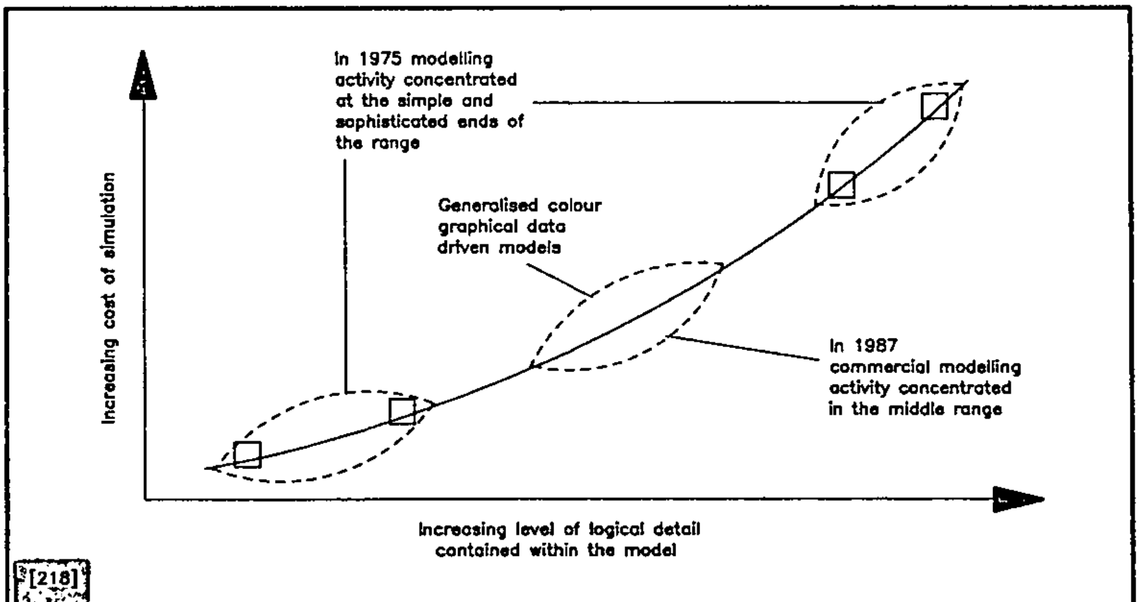


Figure 5.29	*Trends of Model Sophistication*	LUT - FMS Research Group

Chapter 6
SCOPE OF THE RESEARCH

6.1 Introduction

This chapter presents the scope of the research on the application AI techniques in the modelling of flexible manufacturing facilities. The research task is indicated first. After this, the operational structure of the proposed modelling system is highlighted.

6.2 The Research Task

The objective of this thesis is to explore the use of AI techniques for modelling in flexible machining as a competition for conventional discrete event simulation systems [43] [165] [187].

With this end in view, it was decided to develop a knowledge based modelling system with the following features (Figure 6.1):

(1) Although there are many system variants in flexible manufacturing, it was choosen to develop the modelling system which is domain specific to the design of flexible machining cells, but the modelling method should be able to be easily extended to other fields. The flexible machining cells usually consist of machining stations, load/unload stations, pallets, part buffer storages, tools, tool stores, work and tool transporters, and control functions for the management of both work flow and tool flow in the cell [15] [40].

(2) With the advanced knowledge representation facilities provided by the LOOPS environment, the modelling system is to be built within a typical knowledge system structure, where the general control structure should be separate from the modelling knowledge specific to flexible machining and be separate from the application specific information [234a] [45]. As a result, the modelling system should be able to be conveniently extended to the modelling of other discrete event systems, such as assembly systems, by employing the same general control strategies.

The general control structure can be implemented by designing an *inference* *engine* which uses metarules to control the application of domain dependent modelling knowledge which makes decisions by applying the application specific information. The domain dependent modelling knowledge should be expressed in terms of rules. Among these rules, the transformational rules should be developed and stored in a *knowledge base,* but the decision rules which handles the conflicts arising from the application of transformational rules can be specially developed in a *decision centre.* The application specific information can be stored in two connected data bases, the *data base browser* which contains the static data and the *working memory* which stores the dynamic data.

(3) Since the object-oriented programming paradigm provides a close correspondence between modelled objects and real world objects, it is natural to represent the elements of a flexible machining cell using objects [178] [223] [123] [46] [263]. The interactions among these elements are therefore modelled as message passing. As rules can be defined around objects, behaviours of each element are best defined as rules which are more comprehensible than procedures [178] [46]. The connection between the static data base and the dynamic data base can be represented using the access-oriented programming method so that any modification in the static data base can automatically be sent to the dynamic data base. In addition some of the graphic facilities may be developed using this approach. For most of the auxiliary functions, in particular those that do not need to be explicitly represented but are iterative or recursive in nature, they are more easily represented as procedures.

(4) Most manufacturing modelling systems using conventional methods are data driven, but it was decided that this AI based modelling system should be knowledge driven, i.e., both data and rules are used in the formulation of a model [289]. The advantage of this method is that it allows the user to design alternative models easily by entering the data which define the physical structure of a manufacturing cell, and the operational rules which govern the behaviour of the cell elements [178].

The data driven requirement is similar to that of generalised manufacturing simulators [43] [164], but the rule driven requirement is to be defined at two levels. The first level is to be concerned with the expression of decision rules which are normally defined in manufacturing terms [164]. A decision rule can be defined as a rule which is used to handle the conflict between certain actions of objects, such as the part release rule, the station loading rule, etc. [165] [143] [196] [41]. To enable entry of these rules, decision points must be designed so that the related data structures can be accessed. In addition, a rule language is required, which has been provided by LOOPS [53].

The second level is to require more experience with the modelling system, where the existing behavioural rules can be accessed by the user. A behavioural rule may be defined as a rule which changes the system state (or rather the state of the working memory) as the modelling proceeds. There can be two types of behavioural rules, i.e. the transformational rules and the descriptive rules. The former are the rules which describe the actions that make up the model and are concerned with the interactions of the objects. The latter are the rules which are used to model the details of the objects, such as those for accessing and changing particular attributes of an object or for collecting statistics. The user should be able to modify these rules or to specialize them in order to incorporate user-desired logic details. All the transformational rules should be built aournd the corresponding objects [178].

(5) As a result of the complexity of the design process and the need for an structured design approach, this modelling system is intended to be capable of modelling a cell over multiple levels of detail (Figure 6.2). This can be made possible by applying the AI hierarchical abstraction concepts [226].

Since each transformational rule can be defined to model one or a chain of actions associated with the system elements depending on the level of abstraction and transformational rules can be hierarchically structured such that each higher level rule triggers hidden lower level rules, the lower the level the more detailed the modelling, and the less hidden the assumptions [200] [44]. Therefore, each level can be defined to have a set of self- contained primitive rules for the modelling system to apply, and this creates the various levels of detail that are determined by the user's needs [45]. For a flexible machining cell, the actions to be considered are palletisation, unloading, loading and depalletisation at the load/unload stations; loading, empty running, load running and unloading of the work transporter; part loading, tool change, cutting and part unloading at machining stations; empty running, tool loading at STS*, tool transfer to PTS**, tool exchange at PTS, tool transfer to STS and tool unloading at STS for the tool transporters [43] [165] [238].

(6) Since the design and analysis of flexible manufacturing systems has to take an integrated systematic approach, the levels of abstraction have to be consistent with the decisions made at the various stages during the design process [43] [165]. At the early

* Secondary Tool Store
** Primary Tool Store

stages, modelling is required at an aggregate level, while for the subsequent implementation, detailed decisions have to be made by the model.

There can be many possibilities for abstracting a flexible machining facility with varying modelling methods. This topic has been discussed in Chapter 5. According to the decisions a manufacturing system designer has to make during the design process, it was intended to define three levels of modelling detail in this modelling system (Figure 6.3). At the first level, the primary objective is to provide a quick estimation of the performance of the designed system. This estimation should help identify the sufficient numbers of machines, transporters, load/unload stations and pallets. In addition, assessment is to be provided with regard to the work in progress at each station, and this helps to determine the size of the local buffer for each station.

Once these numbers have been accepted, the user is able to enter the next level. The major objective of level 2 is to study the flexible integration effects resulting from integrating the above system elements with buffers and temporary storages of specified capacities [167]. Additionaly tool requirements planning can be conducted in order to give preliminary indication on the strategies of tool management.

Since tool availability can have considerable influence on the performance of a system [63], the third modelling level is intended to assess this effect. This helps to determine the appropriate tool management strategies, the actual tool requirements, the size of major tool stores, and the number of tool transporters.

(7) In order for the modelling system to be able to be used by industrial engineers, a user-friendly interface is intended to be developed. Menu driven software should be developed which enables the selection of desired level of modelling detail from defined options, with explanations being provided with regard to the selection.

The machining cell data should be able to be interactively input and manipulated using a menu driven data base management system. Unlike the conventional data file approach [43], instance objects are to be created to organize and store pieces of information relating to a single concept into a single location. The pieces of information include attributes about the object and with whom the object interacts. Since the number of objects created to describe a machining cell can be considerable. For example, the number of tools involved in a cell tend to be hundreds [84]. Thus there must be an effective method for organizing and managing these objects. The simplest approach is to store all the objects in a list [32], but this suffers from the inefficiency on object searching and the difficulty on identifying subject to the user's access.

A more effective way is to group objects according to the class they belong to [161]. With the LOOPS facilities, data base browsers are to be developed which organize objects by grouping them hierarchically so that a class of objects can be stored in an item collector which is specially designed for that particular class. This can be achieved by specializing the LOOPS *LatticeBrowser* class [53].

The logic of the model is defined by entering decision rules for the defined decision points occuring within the model. This can be realised through two ways. One is to select the built-in decision rules from the rule libraries via an interactive menu driven editor. To help the selection of these rules, each of the decision rules should be paraphased using English. Another is to express the rules, using the LOOPS rule editor [53], by calling to the primitive rules which have been built around each of the objects or by building new primitive rules which access the data structures directly. For the modification of behavioural rules, menu driven software should be developed which can guide the user through to any part of the model and can provide the option for the user to define new rules.

(7) Another feature of the interface that the modelling system should take into account is that the user be able to configure the physical structure of a model by manipulating user-friendly icons and other powerful graphics capabilities. To allow structural choices, a library of icons representing standard and generic classes of cell elements and transport routes can be provided. The user should be capable of interactively selecting icons from the library, placing them on the screen and connecting them as needed [178] [123] [32]. In addition, options should be provided to allow the user to modify the layout structure of a model by deleting and moving the icons placed on the screen and to edit or define new icon images.

Since not all the objects should be placed on the screen [123], such as the components and the tools, the defined physical structure should be automatically linked with the data base browsers so that a fluent specification of the objects can be facilitated via these data base browsers.

In addition to the graphic objects representing the images of the cell elements and transport routes, facilities should be developed which allow the status of each cell element to be dynamically displayed by using the 'split- screen' technique [223]. These facilties should help the user to understand the operation of a designed model and to debug and modify the model as desired. Besides, menu driven software should be developed which allow the LOOPS gauges [53] to be attached to specific objects so as to show the change of a particular parameter of a specific object during model running as desired by the user.

The animation of the operation of the designed model will not be considered within the scope of this thesis, but this aspect will be fully covered in the further work (see Chapter 21).

(8) Although most conventional modelling systems do not have explanation facilities with regard to the modelling process, it was decided that this knowledge based system should be capable of providing explanations about its inference process [106] [178] [223] (Figure 2.19). This can be achieved by tracing the specific rules which are applied during the running of the model. Facilities should be developed which enable trace of the specific object as desired by the user. As a result, the relationship between the computer code and the immediate behaviour of the model can be made explicit, and this enables convenient and straight-forward debugging and modifications of the model.

(9) It should be able to provide with confidence rapid feedback of system performance parameters as desired by the user. As this modelling system is domain specific to the modelling of flexible machining cells, default statistics should be collected and provided automatically with regard to the major aspects of the performance of the cell [165] [43]. At this stage, presentation of the results is to be in a text form, but further research will cover more powerful graphic facilities, such as bar charts, histograms, etc. [165].

Since the user can be guided through to all the behavioural rules of a model, he can then specify performance parameters he desires to collect by inserting new descriptive rules [45].

(10) Since LOOPS has only reached the stage as a research system [289], the computational time for a run in certain cases may be considerable, but in this research priority is given to the support environment of the software and the Xerox Workstation rather than the run time.

6.3 Operational Structure of the Modelling System

With the available Xerox Workstation [302] and LOOPS software [53], and AI techniques, decisions are made to develop an integrated multi-level modelling system [294] [223]. Figure 6.4 shows the user's requirements in flexible manufacturing modelling and the expertise to be embedded in the knowledge based modelling system. Four major areas have been identified by the author as being pertinent to the application of AI/expert systems in solving the manufacturing modelling problem: model configuration and data specification, control rules formulation, model running and analysis of results [168].

For a modelling system specialised in flexible manufacturing, there is great demand for software to be constructed which provides a user-friendly method for the physical configuration of a model and a logical method for the definition and collection of the data needed to run the model [263]. This can be achieved by developing the icon facilities and the data base browser, and applying the AI hierarchical abstraction concepts which enable models of different levels of details.

Once the data has been provided for a model, the next step is to describe how the manufacturing system is to operate. These include decisions of part scheduling, station selection, queue priorities, transporter selection, operation sequencing and traffic control [165]. The facilities to be embedded in the knowledge based modelling system should allow the users to review the default rule which has been selected, to change to another rule from a library of existing rules and to express their own rules using the LOOPS rule language.

After an operational model is established, the user requires facilities to be provided to help understand the behaviour of the model and the computer code behind this behaviour during the running of the model [178]. This can be met by developing graphics and textual output facilities using the Xerox graphics techniques and the LOOPS rule oriented programming method. Another feature which can be fruitful with regard to the running of a model is the application of the concept called rule composition [89]. It uses an automatic learning mechanism to combine the rules which are executed sequentially during a run in order to speed up the run. Although this feature is not to be implemented within the scope of this research, it will be considered in the further research (see Chapter 16).

In order for a modelling system to become an effective tool for solving manufacturing problems, it must provide expertise not only in the above mentioned three areas, but also in the evaluation and understanding of the results [168] [224]. This can be achieved by representing the knowledge of manufacturing evaluations as a set of rules. In addition, the perturbation analysis (see Chapter 5) technique can be applied to develop a separate module to assist in the analysis of the relationship between the input parameters and the modelling results. Again this area is only to be considered in the future work (see Chapter 16).

Figure 6.5 shows the proposed operational structure of the integrated modelling system. On entering the modelling system, the user is required to choose the appropriate level of modelling detail first. As mentioned before, three levels are to be defined.

At the first level, each machine station is assumed to have an infinite local buffer so that no blockage could occure. Temporary storages and tool availability are ignored. These requirements are similar to those of CAN-Q [248], MVAQ [268], PMVA [239] or SIM-Q [85].

Level 2 allows more system details to be modelled. Each machine station or load/unload station has a specified buffer type and size. Part temporary storages, and tool requirements with regard to the whole cell or particular machine stations are considered [50].Decision rules with respect to the availability of buffers and allocation of parts or resources. Models containing similar system features to this level can be found in [43] [165].

At level 3, in addition to the features considered at level 2, influence of tool availability on the performance of the cell is modelled [63] [142]. Primary tool stores, cell secondary tool stores, tool transporters and tool flow strategies [42], are all considered in the modelling. In particular, control rules for work flow subject to tooling availability should be modelled. As for similar studies using different methods, see references [142] [238].

Once the appropriate modelling level is chosen, the user is required to configure the cell layout by manipulating the library-provided icons representing each of the major cell elements. Then the user can enter the cell data within the data base browser, which define the physical structure of the modelled system. After this, decision rules have to be entered to define the behaviour of the system, and if desired, the user can modify the existing behavioural rules or express new ones. Next the user can determine the special output results he desires to collect by entering new descriptive rules, and this will result in an operational model which is ready to run.

During the running of the model, two options are to be provided for understanding the behaviour of the model. One is to provide graphic outputs on the screen showing the dynamic updates of the status of each of the major cell elements. Another is to invoke the trace option by displaying the applying rules in order to follow the line of inference within the model. Gauges can also be attached to specific objects at any time point to dynamically display the value of particular variables of an object.

When the running of the model is completed, the model can give the default results in the desired forms. The user specified outputs can only be displayed or retained by attaching gauges at this stage. If the results from the model are unsatisfactory, the user may want to initiate further runs, and this can be realised by re-configuring the cell layout, re-entering/editing the cell data, or re-selecting/entering the cell operational rules to produce a new specification, and re-run the established model. Once the outputs from the model are satisfactory, the user can exit from the selected modelling level [43].

If a detailed analysis is required, the user can choose a lower modelling level, enter the required data and operation rules in order to design a more detailed model. Once the new model is established, the user can run it and analyse the results obtained from it, and can repeat the analysis process as the previous level until the results are satisfactory [165].

The results from a run can be displayed on the screen by invoking the various options on the output results menus. Hardcopies can be made on the linked printer. Similar ways can be used for the presentation of the input information.

As can be seen from the figure, the rule composition and the analysis of results are not to be considered in this thesis, but will be covered in the further work as two integral parts of the modelling system.

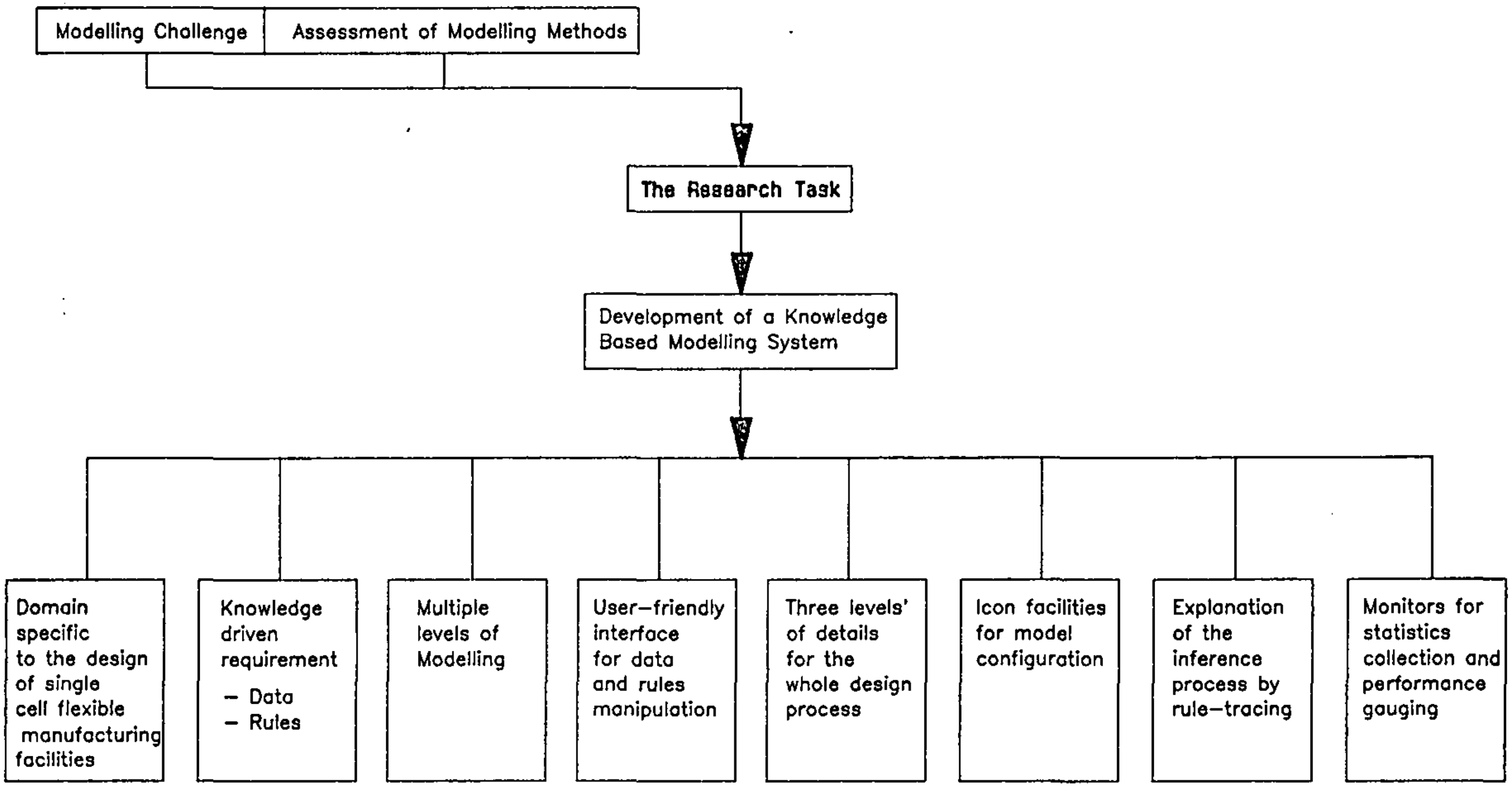


Figure 6.1

The Research Task of the Work

LUT - FMS
Research Group

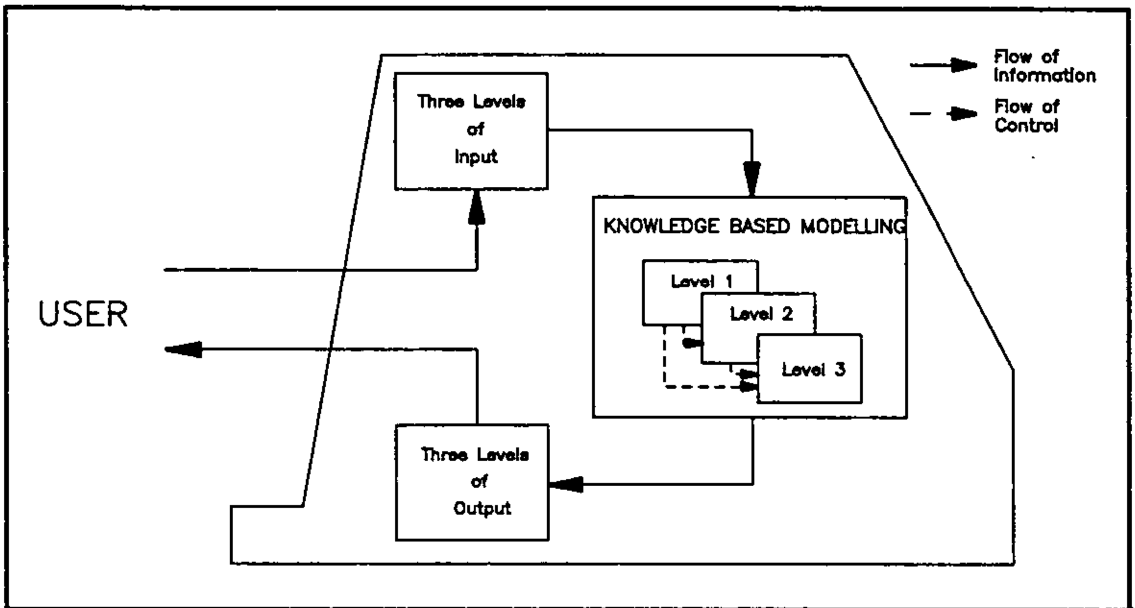


Figure 6.2 — *Hierarchical Modelling Architecture* — LUT - FMS Research Group

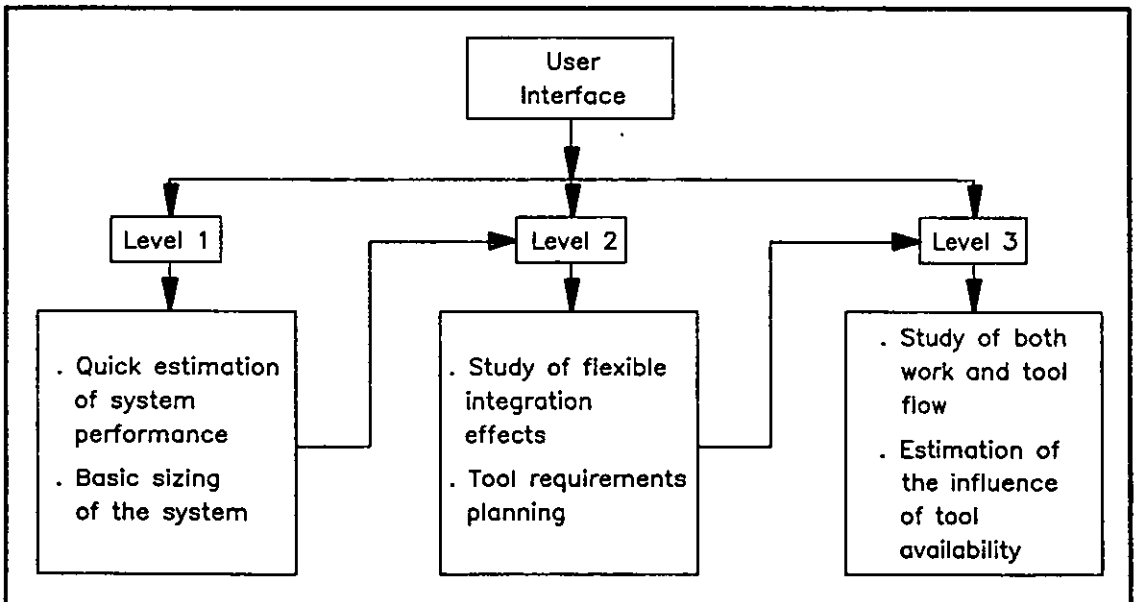


Figure 6.3 — *Objectives of Three Levels of Modelling* — LUT - FMS Research Group

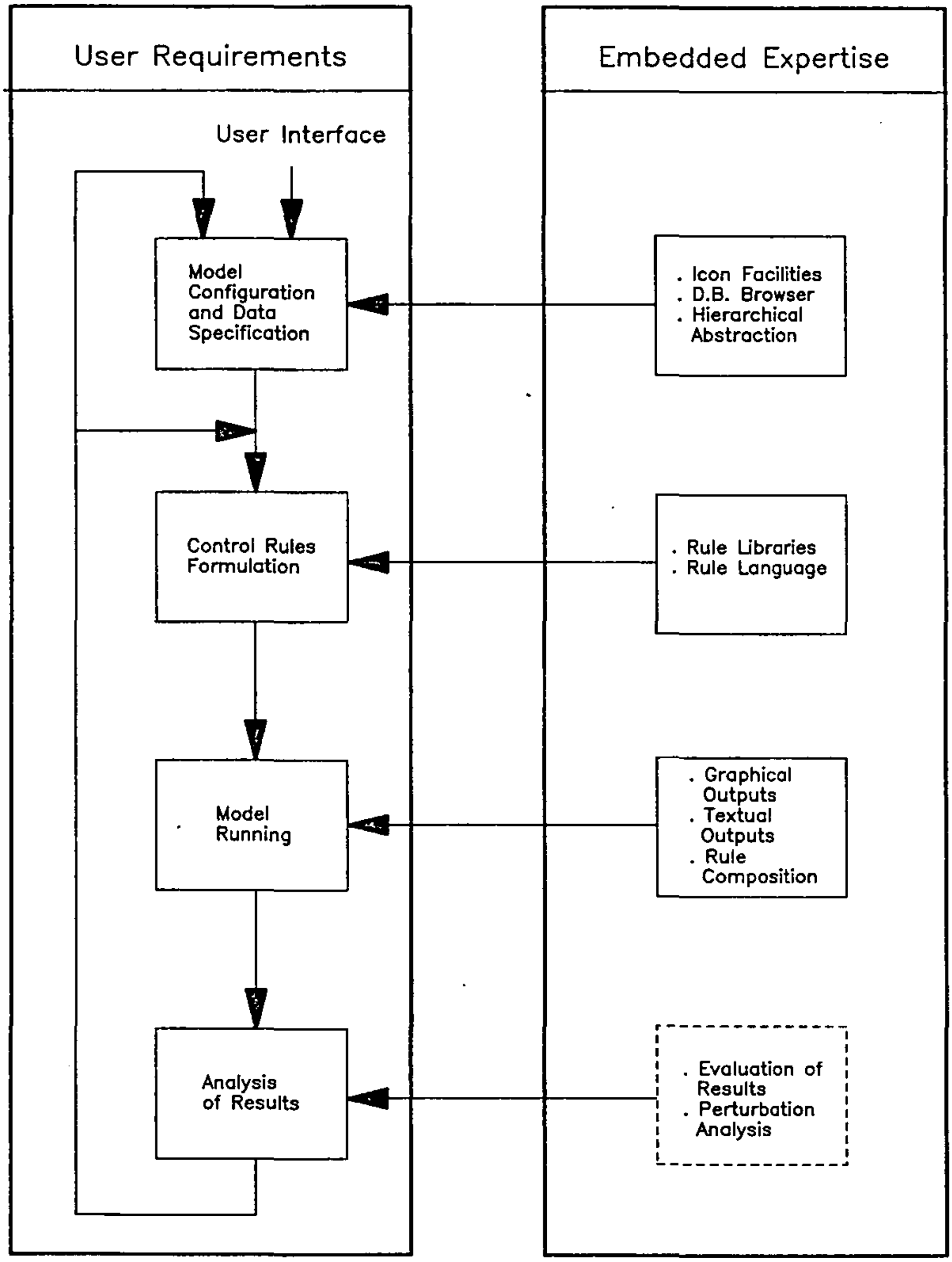


Figure 6.4

Requirments versus Expertise in Manufacturing Modelling

LUT-FMS Research Group

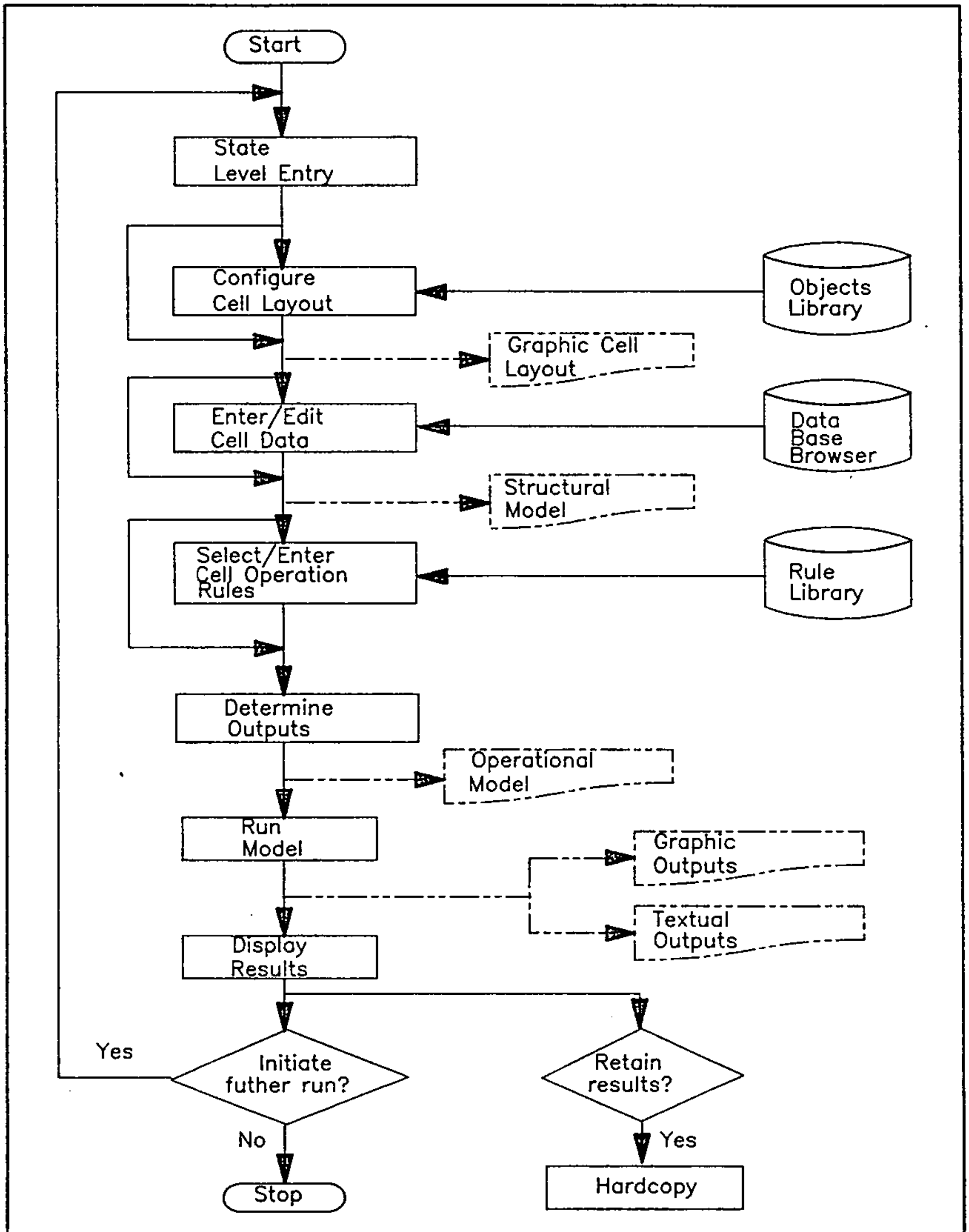


Figure 6.5	*Operational Structure of the Knowledge Based Modelling System*	LUT-FMS **Research Group**

Chapter 7
FUNCTIONAL STRUCTURE OF THE MODELLING SYSTEM

7.1 Introduction

This chapter presents the functional structure of the knowledge based modelling system. First the proposed system structure is discussed. Then each of the major system elements, i.e., the data base, the working memory, the knowledge base and the inference engine, is examined. The knowledge for the three levels of modelling and the user interface are presented more closely in the following chapters.

7.2 System Structure

As shown in Figure 7.1, the overall modelling system has been functionally designed to consist of two basic parts like many other modelling systems [293] [43] [263]:

- the user interafce, and
- the logic elements part.

The user interface is constructed to provide all the facilities required in the user's interactions with the system. These interactions include model configuration, data input, rule entry, control of simulation runs and presentation of simulation results, which are involved in the modelling of a manufacturing system. The links between the components of the interface are of a control nature. For example, the model configurator, the rule entry editor, the graphics and textual output facilities, and the statistics output facilities are all under the control of the global menus, and the data browsing and editing facilities can only be accessed through the model configurator. The structure of the interface and each of its components are more closely described in Chapter 9.

The logic elements part is the main functional component of the modelling system in the sense that all the modelling knowledge is formulated there. It includes the management and maintenance of manufacturing data and modelling procedures, and the logic data structures and functions for the dynamic execution of the designed model.

Four components have been logically constructed and integrated to constitute this part, i.e. *the data base, the working memory, the knowledge base and the inference engine* [234a] [44] [89] [245] [243]. Through the interface, the user can access the data base, the knowledge base and the working memory. The data base is accessed to input and manage the machining cell data, the knowledge base to enter rules and define classes, and the working memory to attach graphic and textual devices for model verification.

The data base is designed to manage the flexible machining cell data associated with a designed model, including the production requirement data, the cell elements data and the part process data. By interactively inputing, editing and managing these data in the data base, the structure of different cell configurations can be specified and modified.

The working memory is constructed to represent the dynamic state of the modelled system. It contains all the objects which are created to represent the corrensponding cell elements as defined in the data base. In addition, it acts as a blackboard [226] for storing all the temporary information which is generated and shared by the other components of the system during the simulation process. As part of the working memory, *the decision centre* is specially designed to maintain and resolve the conflicts arising from the simulation process (Figure 7.2).

As shown in Figure 7.1, *the working memory* is created from *the data base,* and there is information flow from the data base to the working memory. This is effectively facilitated through the access-oriented programming paradigm [53].

The knowledge base contains the knowledge which is either domain dependent or application specific [242], except the model specific data as stored in the data base. The former, although used in a particular application, is general to the domain of flexible machining. The latter has to be defined and entered by the user subject to the requirements of specific applications.

The inference engine is a general structure for controlling the overall functioning of the logic elements part. This is characterized by the information communications between the inference engine, the knowledge base and the working memory. The inference engine determines what rules in the knowledge base should be applied when and how. These rules in the knowledge base are then used to update the state of the working memory. The termination of the inference process is dependent upon the state of the working memory [234a] [226].

It is evident that this structure provides the modelling system with the following advantages:

(1) The four modules of the logic elements part are distinct, and except the inference engine, the other components are accessable by the user through the designed interface. The user can modify any of them without affecting the others. This increases the modularity of the system, its comprehensibility and uniform structure. This is in significant contrast to conventional simulation techniques where the information and control functions are integrated [43] [165] [187].

(2) Since the working memory elements inherit the information from the corresponding objects in the data base browser, the user can flexibly modify or edit the data associated with the objects in the data base browser without affecting the structure of the working memory, and the inference engine can conduct inferences by directly using the information contained in the working memory without communicating with the data base browser. Thus data management for alternative models can be achieved, and this enables convenient modelling of different system configurations.

(3) Since the modelling knowledge with regard to the interactions of cell ellements can be represneted as rules in the knowledge base and rules be organized in various layers of increasing detail, it is possible to represent the state transformations in a hierarchical way. This then brings the hierarchical modelling capability to the system. In addition, the inference process of a model can be based on pattern-directed searches, and no explicit steps are required, which have to be defined in conventional simulations [43].

(4) Due to the explicit representation of conflicts in the decision centre, complicated decision rules can be represented and modelled. This brings the capability of allowing the user to enter his own desired rules.

(5) The state of the modelled system can be dynamically displayed by associating gauges, status windows or other graphic facilities to the objects contained in the

working memory. This is useful for model verification and debugging.

(6) The generality of the inference engine can extend its use to the modelling of other typé of discrete event systems, such as assembly systems.

(7) As the major components of the user interface are separately designed, each of them can be conveniently enhanced and be brought under the total organization by the global menus.

(8) The user can specify the output required for decision making or understanding of the designed system by entering rules to the knowledge base, which collect these statistics.

7.3 The Data Base

As mentioned in Chapter 6, instance objects are created to represent the part spectrum, the cell elements and the part processes. The classes which have been defined as templates for these objects are shown in Figure 7.3. Among these classes, ***DEntities, DStation*** and ***DTransporter*** are *abstract classes*, ***DResourceMixin*** is a *mixin class*, and therefore they can not be used to create instance objects. All these class definitions start with 'D' to indicate that they are used to create the data base elements. The construction of this hierarchy is based on the consideration that more general classes should be defined first, and then the special ones. The following is the definition of a typical abstract class ***DStation:***

```
[DEFCLASS DStation
    (MetaClass AbstractClass Edited:            (* edited:                    )
                                         doc    (* Station mixin for data entry))
    (Supers DResourceMixin DEntities)
    (ClassVariables   (PossibleBufferbms (TwoPosRPBuff-BM FourPosRPBuff-BM
                                         DualPExchange-BM LinePBuff-BM ,
                                         SquarePBuff-BM)
                                         doc    (* Possible buffer images for a station))
                      (Gate-Descriptors ((work 1 24 0)
                                        (tool 1 18 85))
                                         doc    (* Graphic parameters for the work an
```

```
                                        tool paths ))
                (inits ((image SetBitMap (@ ::DefaultImagebm))
                        (buffer SetBuffBm (- self ChooseBuffbm)))
                                  doc      (* The initialisation commands))
                (partList ((image GraphicImage -12 26)
                           (buffer GraphicImage 0 0))
                                  doc      (* The graphic component parts of a
                                           station)))
[InstanceVariables (DefaultBufferbm NIL doc   (* Default buffer image for a station))
                (BufferType NIL doc        (* Part buffer or exhange store))
                (BufferSize NIl doc        (* Rotational buffer size of
                                           station))
                (InputBufferSize NIL doc   (* Size of input buffer))
                (OutputBufferSize NIL doc  (* Size of output buffer))
                (BIndexTime NIL doc        (* Buffer index time of station))
                (BExchangeTime NIL doc     (* Buffer exchange time of station))]
```

In this definition, the class variables specify the parameters and functions for the graphical display of the object in the model configurator (Figure 7.1). Among the instance variables defined, *DefaultBufferbm* is also a graphic parameter used to represent the buffer image specified when placing the object in the model configurator on creation. The other instance variables have to be specified in the data base. An example which inherit these variables is the class ***DMachineStation:***

```
(DEFCLASS DMachineStation
    (MetaClass Class Edited:                 (* edited:          ))
    (Supers DStation)
    (ClassVariables  (super NIL doc          (* Super stack of the machine))
                     (Description "Machine Station"
                                  doc        (* Logical description of machine station)))
    [nstanceVariable (PTSCapacity NIL doc    (* Primary tool store capacity of
                                             machine station.)
                     (TExchangeTime NIl doc  (* Tool exchange time of machine station))
                     (TIndexTime NIL doc     (* Tool index time of machine station))
                     (MExchangeTime NIL doc  (* Machine exchange time)]
```

As can be seen, more instance variables have been defined. These variables together with those defined in ***DStation, DResourceMixin*** and ***DEntities*** constitute the complete definition of a machine station. Depending on the selected modelling level, not all these variables have to be specified. For instance, the *PTSCapacity* of a machine does not need to be defined at level 1.

Since the detailed definitions of all these classes represent the structural knowledge of a machining cell and each modelling level requires the specification of different set of variables defined in a class, the parameter definitions for each object for each level are to be more closely presented in the next chapter.

To organize the created instance objects, the LOOPS class - *LatticeBrowser* has been specialised with a class called *DBaseBrowser* being defined. major definitions include the functions associated with the title bar menu, the left and middle button menus [53], which are the interface facilities. See Chapter 9 and Appendix V for details.

The organization of the instance objects within the browser is realized in a three level hierarchy. The first level acts as a root list which points to the next level which consists of a list of stacks for the different types of objects as shown in Figure 7.3. These different types of objects constitute the third level in the hierarchy. The class which has been defined for the first level is called ***DBase:***

```
(DEFCLASS DBase
    (MetaClass Class Edited:                                (* edited:          ))
    (Supers Object)
    (InstanceVariables  (subs NIL doc                       (* Successors of node))
                        (Identity NIL doc                   (* Identity of the root stack))
                        (ModellingLevel NIL doc             (* Level at which the model is constructed
                                                            and run)
                        (PlanningHorizon NIL doc            (* Planning horizon for the modelle
                                                            system)
                        (SpecifiedSequence NIL doc          (* Specified sequence in which
                                                            components are released into the
                                                            system))
                        (FixedPalletsFlg T doc              (* Flg shwoing whether the number
```

of pallets in the system is fixed or not))

(TemporaryStorageFlg NIL

doc *(* Flg showing whether to use the temporary work storage specified in the data base)))]*

The variable whose value can point to a list of objects is *subs*. It is this variable that links all the instance objects together to form the hierarchy in the browser. The other variables are defined for storing special machining cell parameters which are not represented using objects.

The class defined for the second level is *ItemStack* which is specialised from *DBase* with a number of additional variables being defined to distinguish the different stacks at this level.

Three data base browsers have been defined corresponding to the three modelling levels. For each modelling level, a different set of item stacks are created from *ItemStack* to store a particular type of instance objects. Figure 7.4 shows the item stacks defined for the level-1 data base browser. These include part, pallet, transporter, load/unload station, machine station and process. As shown in Figure 7.5, the level-2 data base browser contains additional categories of objects: temporary storages and tools. In the data base browser for the level-3 modelling (Figure 7.6), more types of objects are contained which are tool transporters and the secondary tool store.

In each of the data base browsers, the data base icon is the root node which points to the different item stacks representd by the corresponding elements icons. For the instance objects defined with the browsers, their identities are shown as labels. It is apparent that this structured organisation of instance objects is obviously more effective than a single list collector which is used in [260]. A similar method has been used in KEE [88], where the class definition in the class browser is used as the instance stack.

7.4 The Working Memory

The working memory is a dynamic data base which stores the current knowledge about the tasks being performed. Figure 7.7 shows the structure of the working memory. It consists of four different types of elements: facts, goals [203], conflicts [206] and statistics collectors. Refer to Appendix II for the complete definition of the class *WorkingMemory.*

The first are represented using dynamic instance objects of classes defined within the class browser (Figure 7.8). Variables and variable properties adhered to the objects are used to describe the state of system elements.

As shown in Figure 7.8, all the classes for creating dynamic instance objects are specializations of the classes defined for the data base. Three abstract classes, i.e. ***Entities, Station*** and ***Transporter***, and a mixin class- ***ResourceMixin***, are added. All the other classes can be instantiated to create the dynamic instance objects for the working memory. These include ***Operation, SubOperation, PartType, ToolType, Part, Tool, Pallet, Storage, SecondaryToolStore, InspectionStation, LoadUnloadStation, MachineStation,WashingStation, ToolTransporter, WorkTransporter, HMachiningCentre, VMachining -Centre,*** and ***Lathe.*** See Appendix II for the definition of these classes. The following is the definition of the abstract class ***Station:***

```
(DEFCLASS Station
    (MetaClass Class Edited                              (* edited:          )
    (Supers ResourceMixin Entities)
    (InstanceVariables  (PalletOnStation NIL doc         (*Pallet that is currently residing on
                                                          station))
                        (PalletsInBuffer NIL doc         (* Pallets that are in the buffer of the
                                                          station))
                        (PalletsInInputBuffer NIL doc    (* Pallets that are in the input buffer of the
                                                          station))
                        (PalletsInOutputBuffer NIL doc(* Pllets that are in the output buffer of the
                                                          station))
                        (QueueLength NIL doc             (* Current queue length in terms of
                                                          pallets00
                        (MaxQueueLength NIL doc          (* Maximum queue length of station))
                        (StationaryTime NIL doc          (* Time that the station is stationary with a
                                                          pallet))
                        (LoadingUnloadingTime NIL
                                             doc         (* Cummulative time that the station is
```

loading/unloading)))]

In this definition, all the instance variables change dynamically during the running of the model. Among these variables, *PalletOnStation, PalletsInBuffer, PalletsInInputBuffer andPalletsInOutputBuffer* are used to describe the state of a station, whereas *QueueLength, MaxQueueLength, StationaryTime and LoadingUnloadingTime* are statistics indicators or collectors for a station. A sample class which is specialized from *Station* is ***MachineStation:***

```
(DEFCLASS MachineStation
    (MetaClass Class Edited:                          (* edited:          ))
    (Supers DMachineStation Station)
    (ClassVariables    (ResourceType MachineStation
                                        doc          (* Type description og the resource))
    (InstanceVariables (ToolTimes NIL doc             (* Tool and cutting time list))
                       (ToolRequirement NIL doc       (* Tool types and times for this machine))
                       (TotalToolTypes NIL doc        (* Total tool types required by the
                                                      machine))
                       (TotalToolsNumber NIL doc      (* Total number of tools required by the
                                                      machine))
                       (ToolList NIL doc              (* Tools that are required by the machine)
                       (ToolsInPTS NIL doc            (* Tools that are in the PTS))
                       (UnusableTools NIL doc         (* Tools that are not required by the next
                                                      job))
                       (ChangesOfWornTools NIL
                                        doc          (* Number of changes of worn tools))
                       (ChangesOfUnusableTools NIL
                                        doc          (* Number of changes of tools that are not
                                                      usable))
                       (ChangesOfPositionTools NIL
                                        doc          (* Number of changes of tools that are not
                                                      the required types))
                       (MachiningTime NIL doc         (* Time that the machine is in machining)
                                        Percentage NIL)
                       (CuttingTime NIl doc           (* Time that the machine is in cutting)
                                        Percentage NIL)
                       (ToolChangingTime NIl doc      (* Time that the machine is changing tools)
```

Percentage NIL)

(TimeWaitingForTools NIL

doc (Time that the machine is waiting for tools)*

Percentage NIL)

(ToolLoadUnloadTime NIL

doc (Time that the PTS exchanges tools with tool transporters))))]*

Notably, in this definition, except the instance variable *ToolsInPTS* which is used to represent the state of the PTS of a machine, all the others are various statistics indicators or collectors. The class variable *ResourceType* is defined to distinguish a machine station from other stations, such as load/unload stations.

Goals are the scheduled bound events which provide a direction to the system's processing by sketching the situations that must be achieved [89] [200]. Two instance variables (*FinishTime* and *GoalStatus*) of a working memory element are used to represent the goals. They help ensure that the most appropriate aspects of a task are searched and processed first.

In the level-1 and level-2 modelling, since only work flow is considered, these two variables are only defined for the class *Pallet*. However, they are defined for both the class *Pallet* and the class *ToolTransporter* at level 3 as both work flow and tool flow are modelled at this level. Refer to the next chapter for more detailed discussions on this issue.

The third type of elements in the working memory, which are also represented by objects, are the various conflicts among the actions of system elements with regard to the selection of resources for a component or the sharing of a resource by several components [243]. Figure 7.9 shows the classes of conflicts defined in the class browser. Here, the classes *ConflictSet, PalletConflictSet* and *ResourceConflictSet* are abstract classes, while the seven specialised classes, *OperationConflictSet, ToolingConflictSet, ReleaseConflictSet, LUStationConflictSet, NextStationConflict-Set, ToolTransporterConflictSet* and *WorkTransporterConflictSet,* are used to create instance objects which act to record the different types of conflicts generated in the modelling process. Refer to Appendix II for the detailed definition of these classes. The following lists the variables defined for ***ReleaseConflictSet:***

```
(DEFCLASS  ReleaseConflictSet
    (MetaClass Class Edited:                        (* edited:          )
    (Supers ConflictSet)
    (ClassVariables   (Type #$ReleaseConflictSet
                                      doc     (* Logical type of the conflict set))
                      (CandidateType "Part" doc    (* Type of candidates))
                      (Rule Library (("Specified Sequence" SpecifiedSequence
                                              "Will select specified sequence rule."
                            "This rule selects the part type according to a pre-specified
                            list.")
                               ("Earliest Due Date" EarliestDueDate
                                              "Will select earliest due date rule."
                            "This rule selects the part type that has the earliest due date.
                            It intends to minimise the lateness of the order."))
                                      doc     (* Release rule library))
                      (SpecifiedRule SpecifiedSequence
                                      doc     (* Rule that has been specified or selected
                                              by the user for particular application.)))
    (InstanceVariables (PartTypeSelected NIL doc   (* Part type that is selected by the
                                              application of the release rule))
                      (PartTypes NIL doc       (* Part types that compete for the same
                                              pallet.))
                      (FilteredPartTypes NIL doc  (*Part types that are left after  initial
                                              filtering))
                      (Pallet NIL doc          (*Pallet that parts are competing for))))
```

Here, the class variable *Type* is used to distinguish this type conflict set from the others in the decision centre. The class variable *CandidateType* is a logical description of the candidates in the conflict set. The class variable *RuleLibrary* is the key parameter for a conflict set, which contains all the information and functions for setting up the rule library menu and activating the rule entry on invocation. For conciseness, only two rules are shown in the above definition. The strings *"Specified Sequence"* and *"Earliest Due Date"* are used as item labels on the rule library menu. The next parameter, *SpecifiedSequence* or *EarliestDueDate*, is the actual decision or conflict-resolution rule which is defined around this conflict set. The next string is the help information to appear in the prompt window on selection of the rule. The last string is the English paraphase for the rule, which is displayed in the summary window (see Chapter 9 for details). It is apparent that this structured definition provides a very flexible framework for addition of other rules.

As mentioned before, the conflicts are contained in the *decision centre* which is defined as follows:

```
(DEFCLASS DecisionCentre
    (MetaClass Class Edited:                        (*edited:          )
                                        doc         (* Decision centre defined for decision
                                                    making))
    (ReleaseConflictSets NIL doc                    (* List of release conflict sets))
    (OperationConflictSets NIL doc                  (* List of operation conflict sets))
    (LUStationConflictSets NIL doc                  (* List of load/unload station conflict sets))
    (WorkTransporterConflictSets NIL doc            (* List of work transporter conflict sets))
    (NextStationConflictSets NIL doc                (* List of next station conflict sets))
    (ToolingConflictSets NIL doc                    (* List of tooling conflict sets))
    (ToolTransporterConflictSets NIL doc            (* List of tool transporter conflict
                                                    sets)))]
```

Since the number of conflicts of each type generated in the suimulation process is problem-dependent, a machanism is required which can create new conflict sets if there are not enough in the decision centre (Figure 7.2). These newly created conflict sets should be automatically put into the decision centre and be used afterwards. To make the most efficient use of these data structures, the conflict sets are cleared after conflict-resolution so that they can all be used again.

In addition, the working memory contains information about sampling and recording, which will provide the statistics of the system performance [203]. This is facilitated by th instance variables and the variable properties defined for the system entities (refer to the definitions of *Station* and *MachineStation*) and the working memory (Appendix II). See Chapter 10 for a more detailed discussion on the collection and computation of the system performance statistics.

7.5 The Knowledge Base

The knowledge base is the main component of the modelling system in the sense that all the knowledge about the system being modelled is formatted there. It is built mainly around the LOOPS class browser [53]. According to the way that knowledge is represented, there are three basic types of modelling knowledge, i.e. classes, rules and procedures (Figure 7.10). Rules are defined in the class browser around the classes, which are used to create the instance objects for the data base browser or the working memory. Procedures are either LOOPS methods which are defined for classes, or LISP functions which are of general use.

The classes and methods defined for the data base browser elements can be accessed through the model configurator, and new classes be defined. The classes for the working memory elements are accessable through the rule entry editor, because they provide the work spaces for all the rules defined.

The rules can be divided into three main groups (Figure 7.10) according to their functions in the modelling system:

- ***Inference rules***: These are the metarules contained in the inference engine, which are to be more closely described in the next section.

- ***Decision rules:*** These are the conflict-resolution rules which select a resource from alternative available ones for a component, or a componet from a list for allocation to a resource [165] [41] [89a] [257]. Examples are the *LWL* (least work load) rule which selects a station with the least work load for the subsequent operation on a component, and the *FTU* (fewest tools unavailable) rule which selects a part with the fewest required tools being unavailable in the magzine for the next loading, etc.

- ***Behavioural rules:*** These rules are defined around the classes for the working memory elements to describe the behaviour of particular objects, such as a machine station, a transporter, etc.[178].

According to whether these rules interact with the inference rules or not, behavioural rules can be further divided into ***transformational rules*** and ***descriptive rules***. The transformational rules are the set of statements which describe the actions that make up the model, and in particular, the interactions between the obejcts in the working memory. The application of these rules triggers the change of the state of the working memory.

Depending on the functionality of the transformational rules in the simulation process, these rules can further be divided into the following (see next chapter for more details):

- *Start-action recognition rules*: In these rules, various actions that can be applied are recognised by matching the rules with the individual or combinations of elements in the working memory, and the conflicts concerning alternative available resources for an action are detected and automatically put into the working memory.

- *Start-action rules*: In these rules, the states of the system elements in the working memory are changed by starting the vaious actions whose conflicts have been resolved and setting the goals to be achieved by the actions. For example, Start-Processing corresponds to loading the part from the buffer onto the machine; setting the status of the machine to busy; setting the time of the goal to the addition of the current time and the expected processing time, and the status of the goal to active; etc.

- *End-action rules*: These rules change the state of the relevant elements of the working memory by ending the actions in order to achieve the goals. For instance, End-Processing will load the part into the local buffer, set the status of the machine to idel, set the status of the goal to dead, and update the cummulative busy time of the machine, etc.

The ***descriptive rules*** are the rules which are used or called by the transformational rules to directly change the data structure or the details of the objects, or to collect the statistics of the objects. The following is an example of descriptive rules, which is defined around the class *MachineStation* to describe the state change of a machine station upon starting part loading and processing at the station:

WorkSpace Class: Machine Station;
Compiler Option: A;
Temporary Vars: ;
Control Structure: DOALL;
Rule Class: MyRule;

****

```
IF      :BufferType='PartBuffer
THEN    .CollectIdleTime
        :Status-'LoadingAndProcessing
        .SchedulePalletOperation
        .CollectLoadingUnloadingTime
        :PalletsInBuffer-(REMOVE :PalletOnStation :PalletsInBuffer);

IF      :BufferType='ExchangeStore
THEN    CollectIdleTime
        :Status-'LoadingAndProcessing
        .SchedulePalletOperation
        .CollectLoadingUnloadingTime
        :PalletsInInputBuffer-(REMOVE :PalletOnStation :PalletsInInputBuffer);
```

In this ruleset, two rules are actually defined. The first rule handles the case where the machine station has a *PartBuffer* -type buffer, and therefore the state changes include the updating of the status of the machine and the transfer of the pallet selected from its local buffer. The other three rules, i.e. *.CollectIdleTime*, *.SchedulePalletOperation* and *.CollectLoadingUnloadingTime* are themselves descriptive rules for collecting statistics. The second rule does the same function, except that it handles the case where the machine has a *ExchangeStore*-type buffer.

For efficiency, some of the methods of objects are developed using the procedure-oriented programming paradigm of LOOPS. Although they are less comprehensible, these methods are functionally self-contained and can be used as general utilities. An example is the LISP function definition called *PickLowObj* . It takes an object, a method and a list of candidate objects as arguments, and selects the one from the object list which returns the lowest value on applying the method (see the use of this function for new rule definitions in Chapters 14 and 16). Although no facilities have been built for the access or application of these procedures, they can be called directly when they are needed.

7.6 The Inference Engine

The inference engine of the modelling system is responsible for the execution process by controlling the termination, goal searching, end-action calls, identification of start of actions, conflicts resolution and start of actions. It specifies control explicitly by using metarules (rules which determine how to apply other rules) [200]. The following is the class definition of ***InferenceEngine:***

```
(DEFCLASS InferenceEngine
    (MetaClass Class Edited:                              (* edited:        ))
    (Supers Object                           doc          (* Definition of inference engine))
    (InstanceVariables (ModellingLevel NIL doc            (* Level at which the model is to be run))
                       (MasterClock NIL doc               (* Master clock of the engine))
                       (BlockTime 0 doc                   (* block time of the simulation process))
                       (DisplayFlg NIL doc                (* Flg showing whether simulation is
                                                          displayed or not))
                       (StartTime NIL doc                 (* Start time of the run))
                       (FinishTime NIL doc                (* Finish time of the run))
                       (RunTime NIL doc                   (* Run time of the model)))]
```

As can be seen above, the class definition of the inference engine is fairly simple. The main variables are the *ModellingLevel* and *MasterClock*. The former is required to be specified by the user, and the latter functions as a clock for keeping the current time as simulation proceeds.

This inference engine has been designed to have eight metarules for controlling the running of a model, which are all contained in the top metarule ***RunModel:***

```
WorkSpace Class: InferenceEngine;
Compiler Option: A;
Temporary Vars: ;
Control Structure: WHILEALL;
While Condition: T;
Rule Class: MyRule;

*********

IF      .TerminationCondition
THEN    (Stop T 'Done);

THEN    .NextGoal;
```

THEN .EndActions
.UpdateStatesDisplay;

IF .FindFreePallets
THEN .TestReleaseOfParts;

IF .ResolveReleaseConflicts
THEN .ReleaseParts;

THEN .TestStartOfActions;

IF .ResolveStartActionConflicts
THEN .StartActions
.UpdateDisplay;

THEN .BlockModelling;

As shown in Figure 7.11, only the first seven metarules are used as the control executive for the whole simulation process, which are executed in the following order. The termination metarule first checks if the execution can terminate. The goal identification metarule then searches the most recent goal which can be achieved. Following this, the end-action metarule accomplishes the goals identified above by ending relevant actions [89].

Next, the parts-release recognition metarule checks the release of parts into the system and detects the conflicts with regard to the release of different part kits. After this, the parts-release metarule resolves the conflicts detected above and allocate part kits to available pallets.

Following this, the start-action recognition metarule checks each of the actions which are waiting to be started and detects the conflicts with respect to the competition of resources for the same work. Finally, the start-action metarule resolves the conflicts identified above, detects the conflicts with regard to the sharing of resources by different components, and immediately executes the conflicts-resolved actions.

Refer to Appendix III for the detailed definition of these metarules.

The eighth metarule is used in the above definition to block the simulation process so that a simulation can be suspended, resumed, or stopped under the control of the user. The *.UpdateStatesDisplay* and *.UpdateDisplay* are two rules for updating the status windows of the objects during the simulation. They are not used for state transformations but the dynamic display of the state of the modelled system.

There are two basic types of termination conditions for a simulation. One is to simulate a system for a specific time period [165], and another is to terminate the simulation when a specified production requirement is completed [43] [142] [89a]. The latter has been considered in the modelling system, and the former can easily be implemented. The *.NextGoal* metarule is actually implemented in a LISP procedure, the logic flow of which is shown in Figure 7.12.

Since in modelling there are many equally acceptable goal states but only one single initial state, the author chose a forward-chaining rather than a backward-chaining method [3] [226] [234a] for the inference engine of the modelling system. Another point that supports the forward-chaining strategy is that even in very simple systems, there is no predetermined final state, and the purpose of modelling is really to discover what the future will look like [279]. Furthermore, an irrevocable search strategy should be used by the inference engine for the purpose of evaluting rather than optimising the performance of a designed system. Therefore, many tentative search strategies [3] [226] are not appropriate.

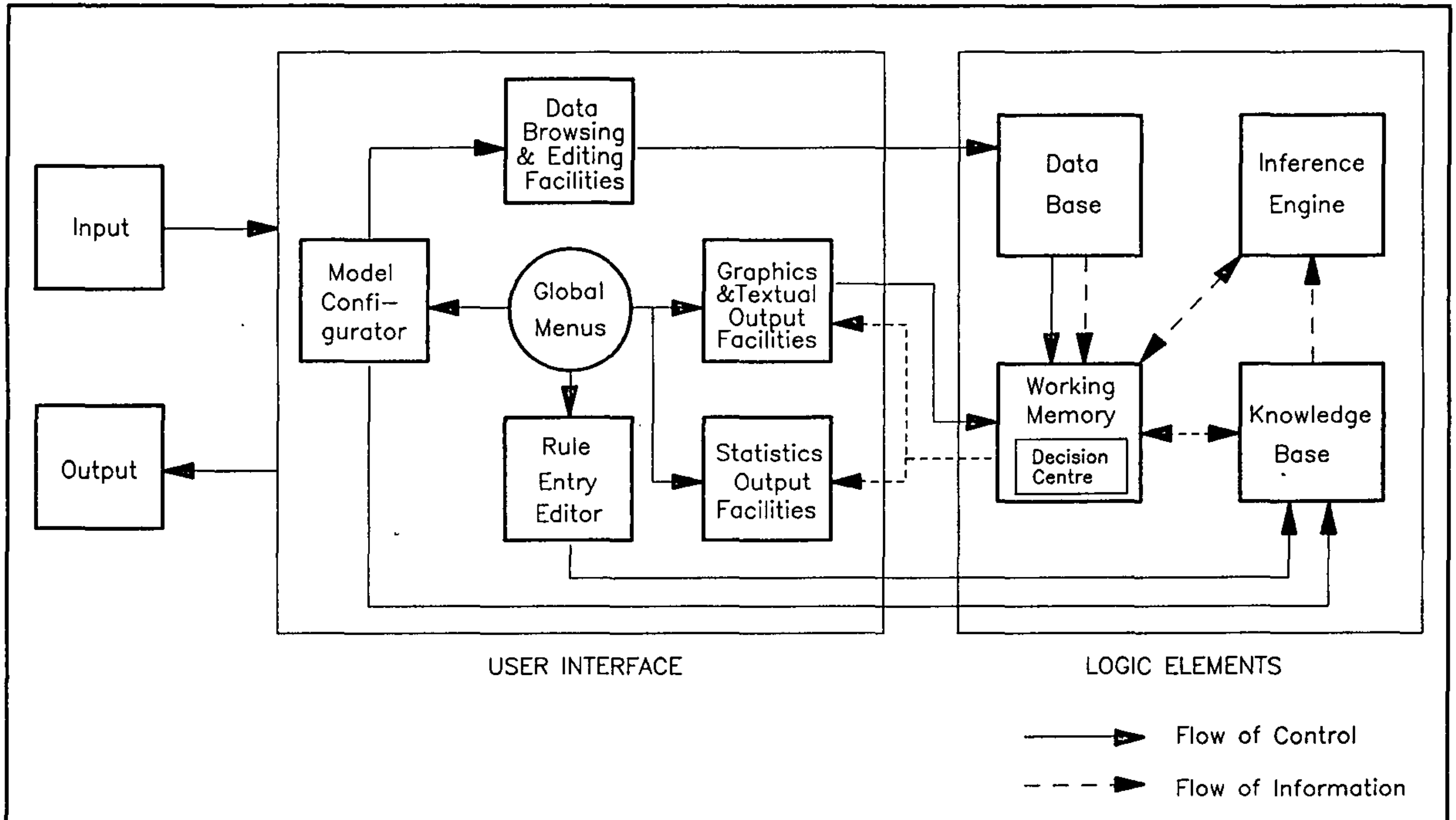


Figure 7.1	*Functional Structure of the Knowledge Based Modelling System*	LUT - FMS **Research Group**

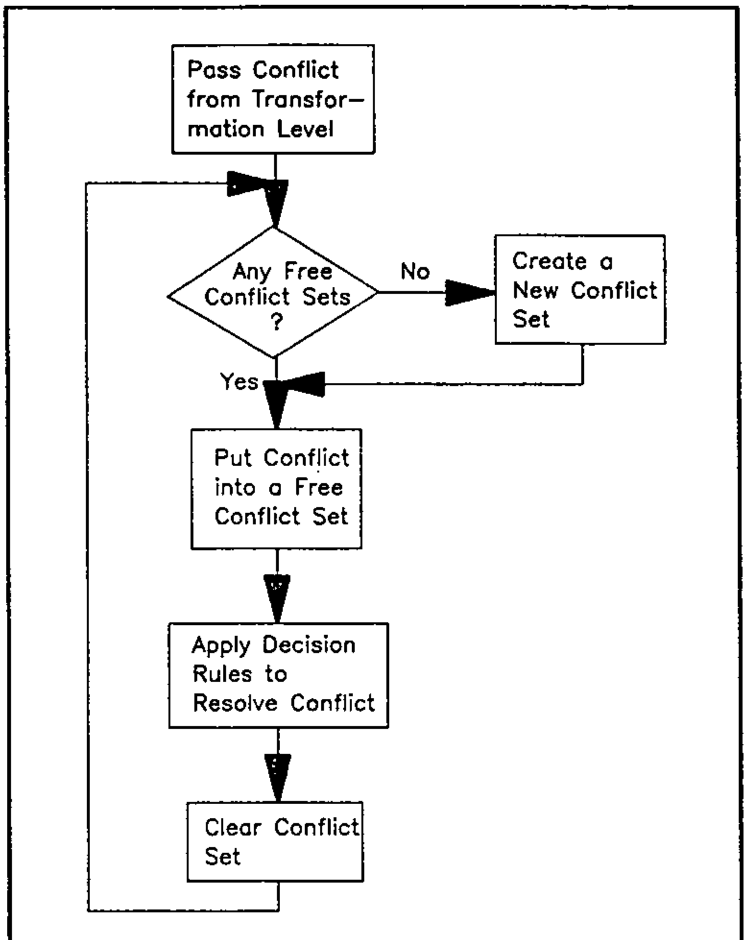


Figure 7.2	*Operational Structure of the Decision Centre*	LUT-FMS Research Group

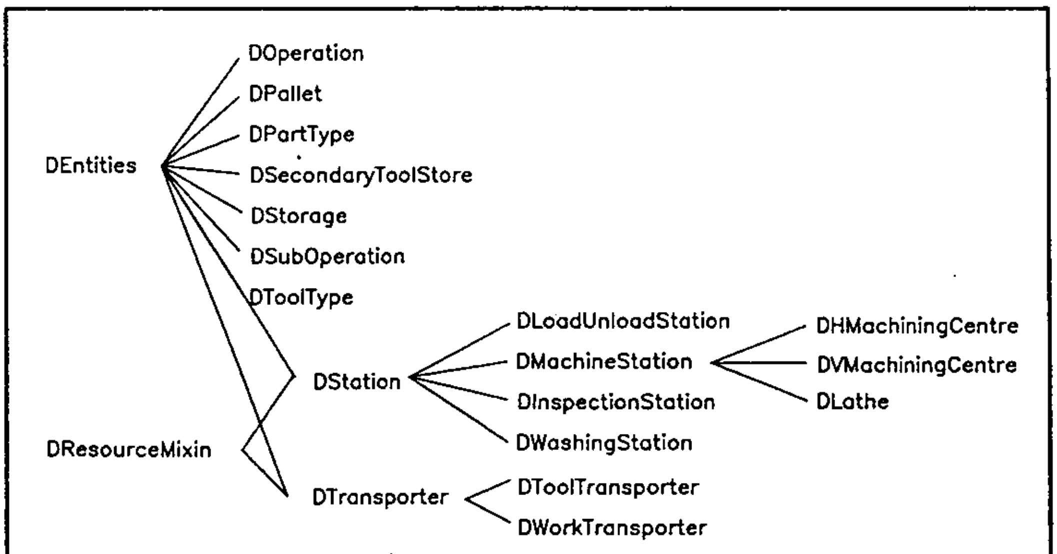


Figure 7.3	*Class Inheritance Lattice - Data Base Browser Elements*	LUT - FMS Research Group

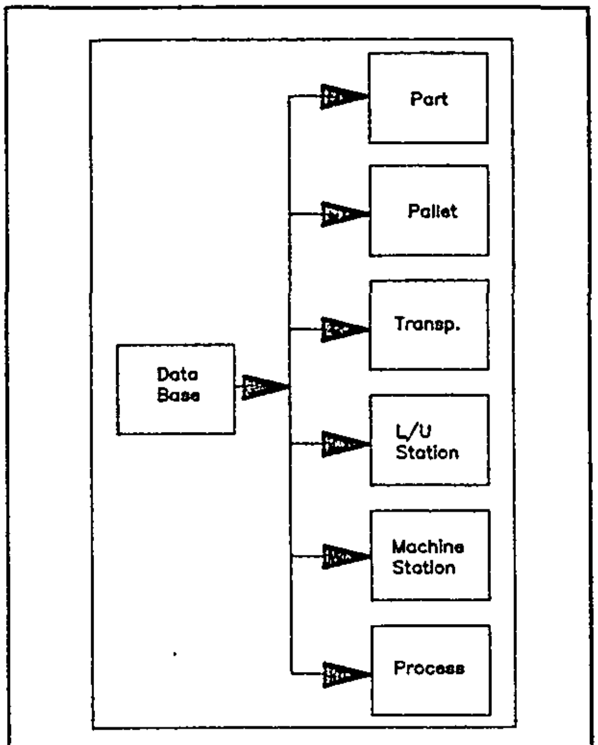


Figure 7.4	*Data Base Browser - Level 1*	LUT-FMS **Research Group**

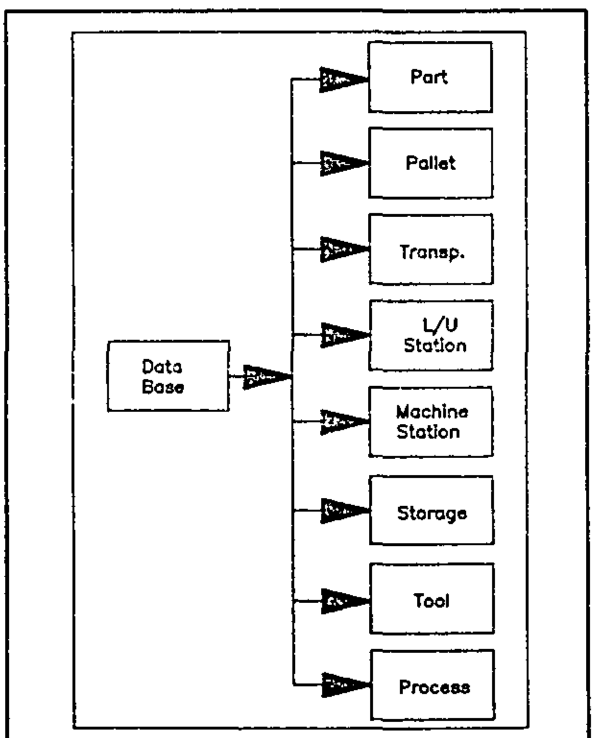


Figure 7.5	*Data Base Browser - Level 2*	LUT-FMS **Research Group**

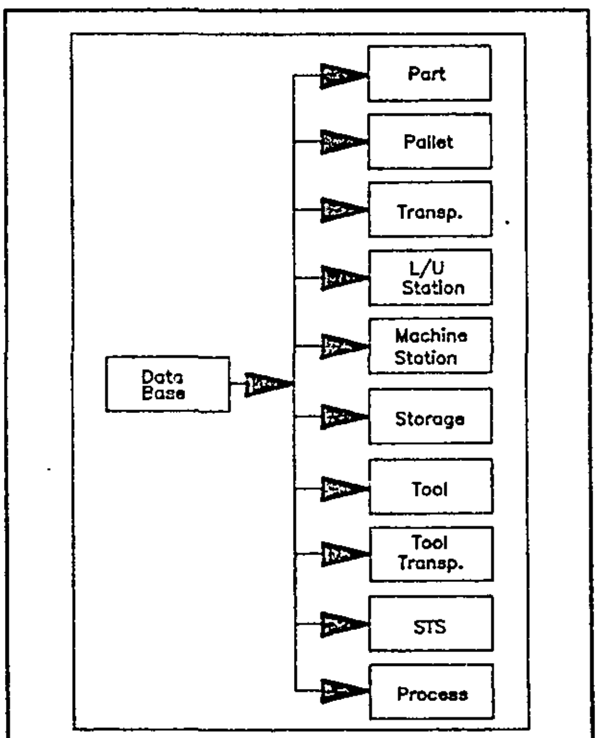


Figure 7.6	*Data Base Browser - Level 3*	LUT-FMS **Research Group**

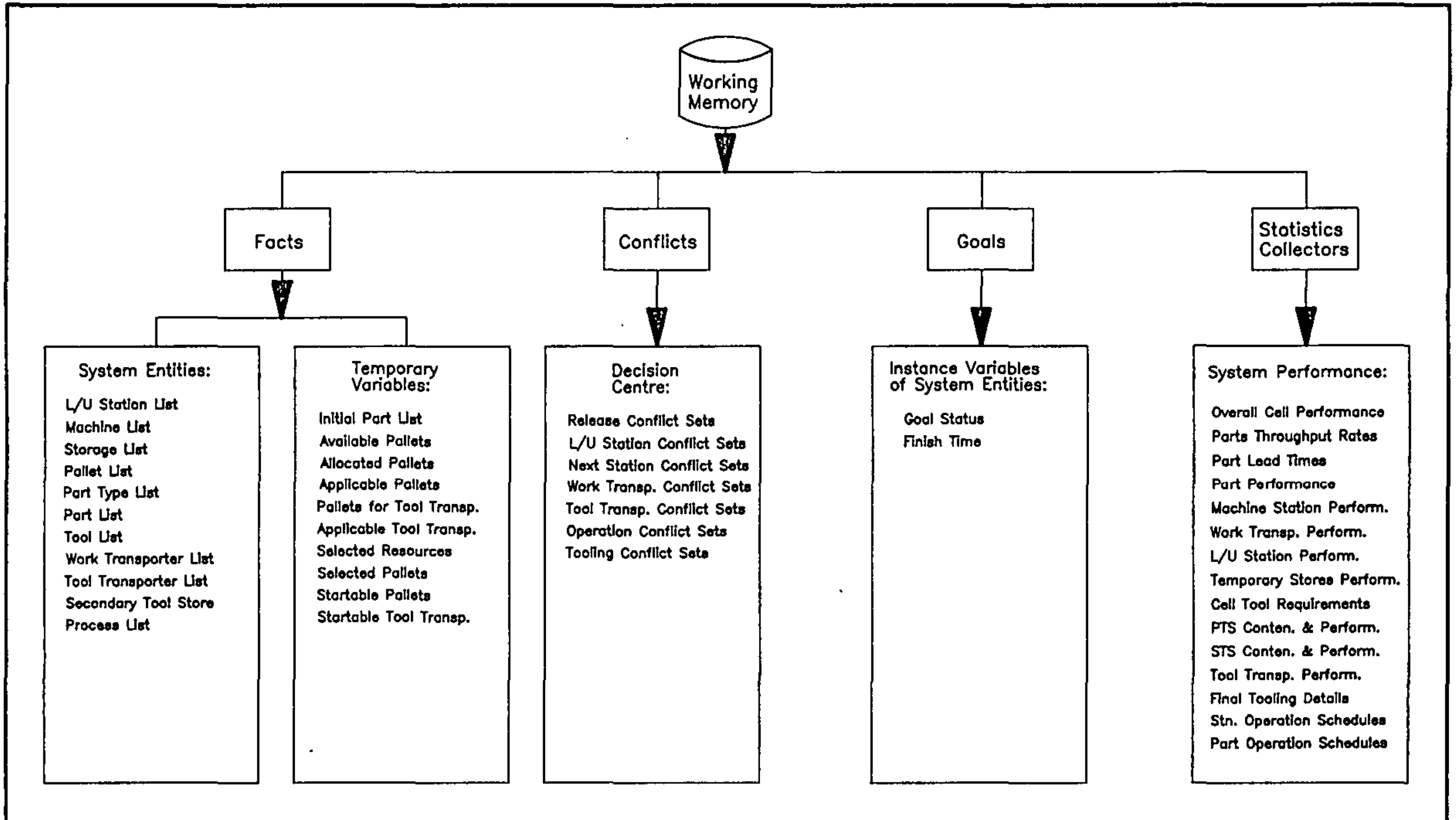


Figure 7.7	*Structure of the Working Memory*	LUT - FMS Research Group

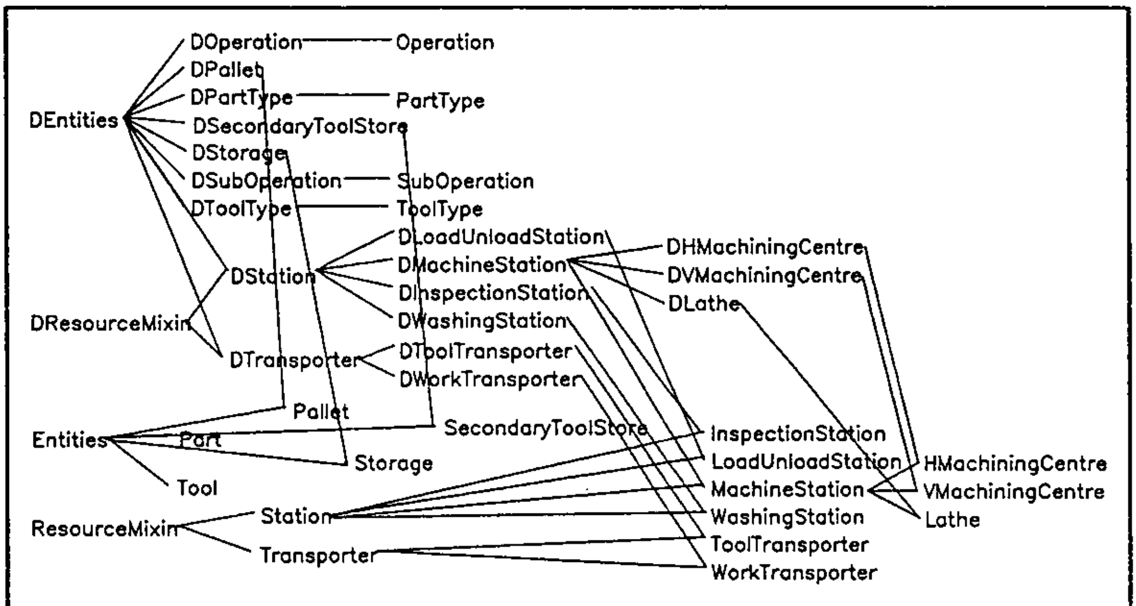


Figure 7.8	*Working Memory Elements - System Entities*	LUT - FMS **Research Group**

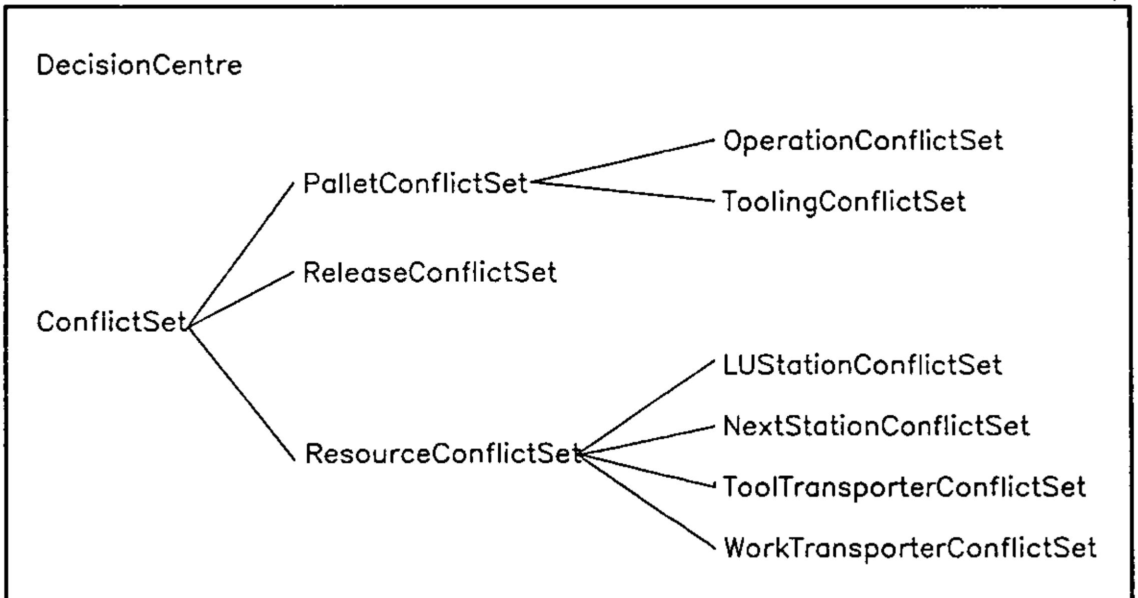


Figure 7.9	*Working Memory Elements - Conflict Sets*	LUT - FMS **Research Group**

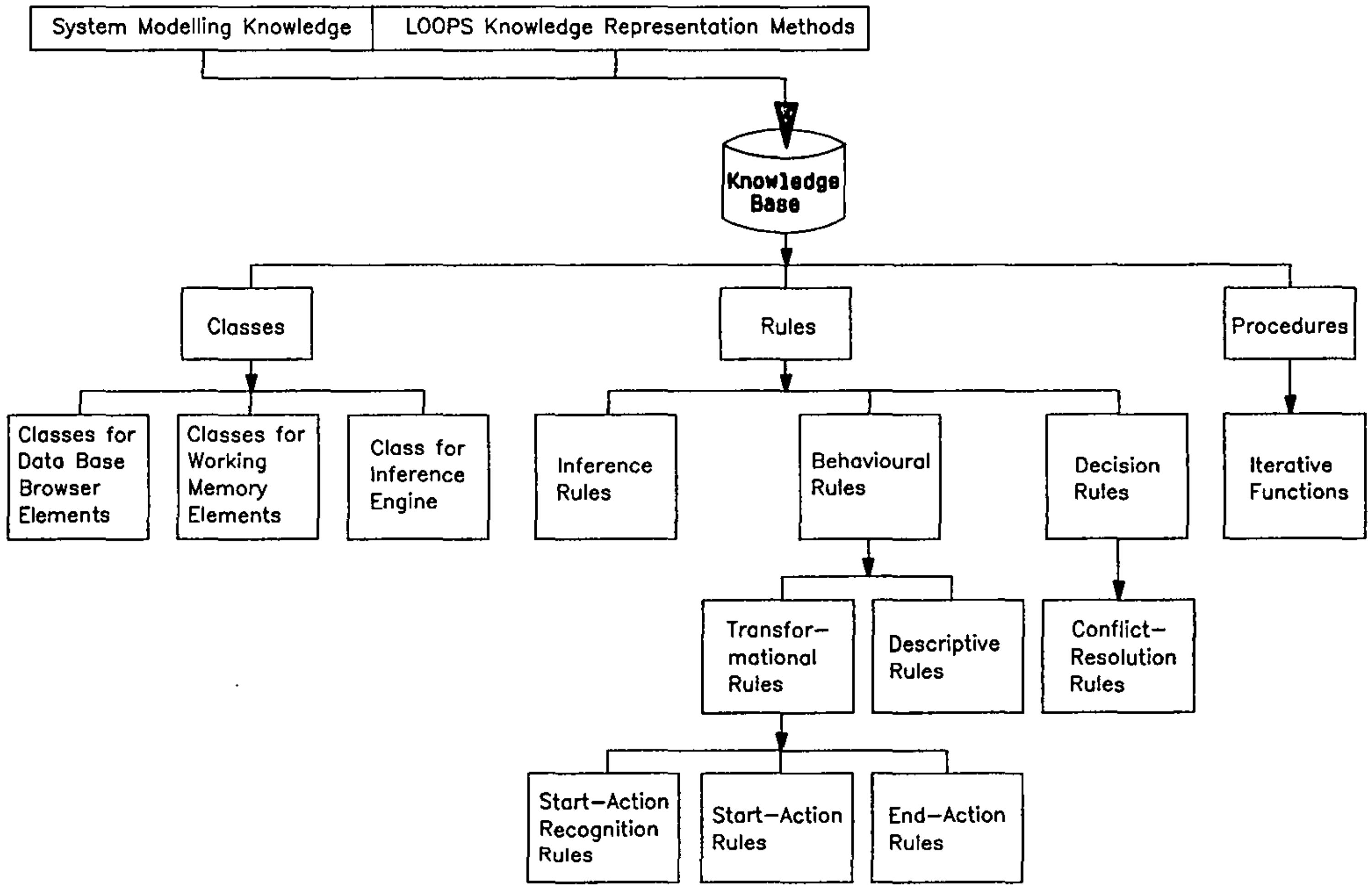


Figure 7.10

Contents of the Knowledge Base

LUT - FMS
Research Group

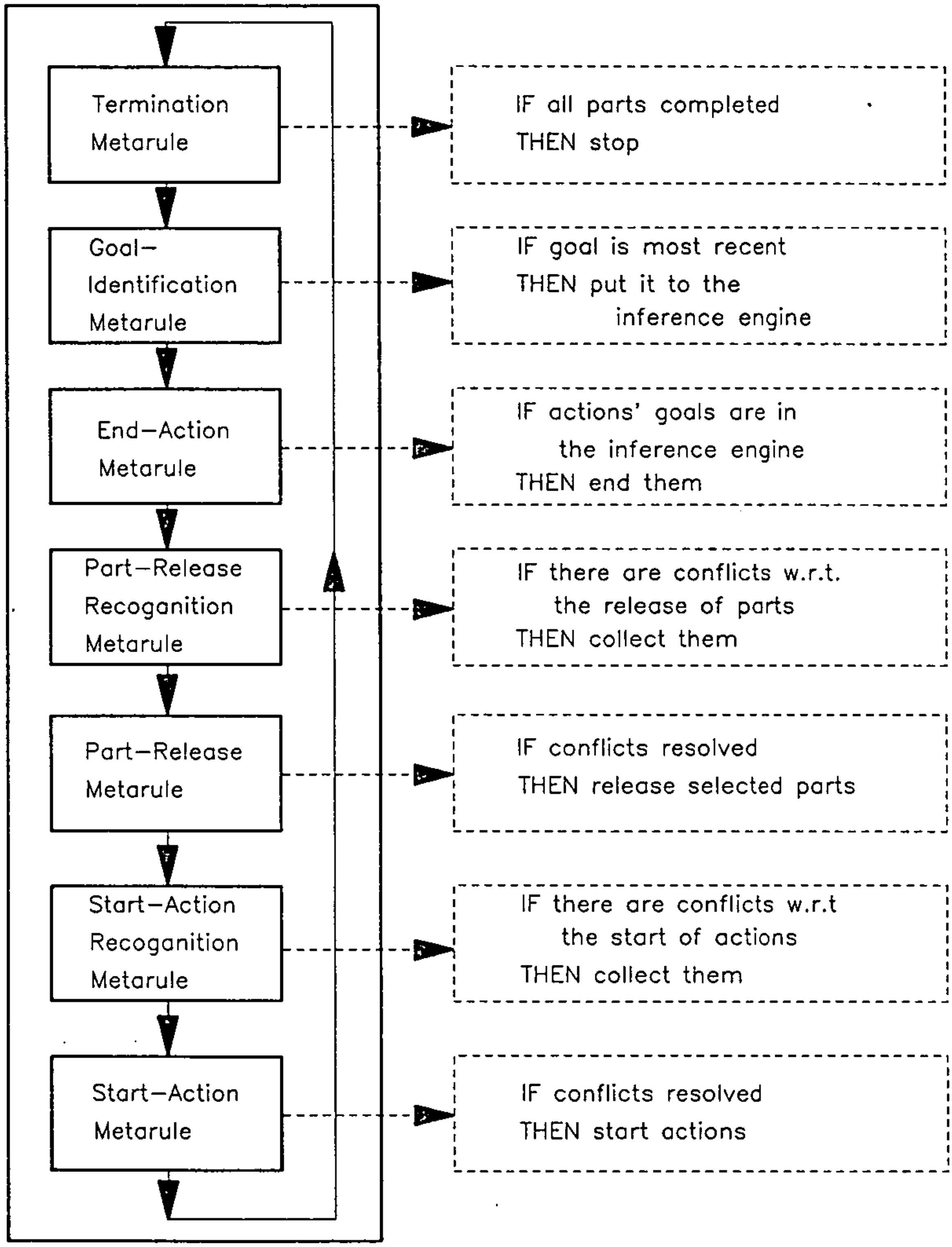


Figure 7.11	*Functional Structure of the Inference Engine*	LUT-FMS Research Group

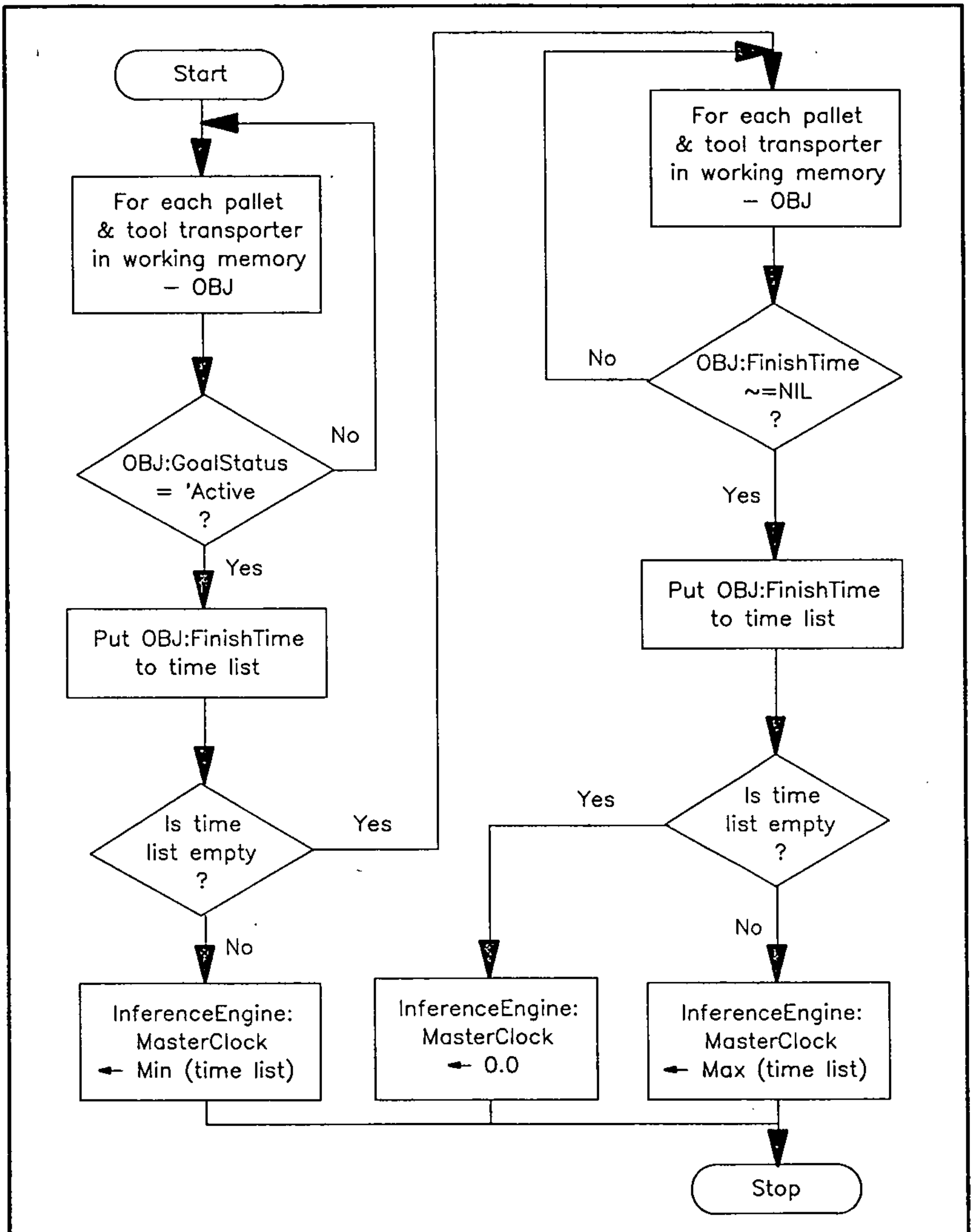


Figure 7.12	*Logic Flow for InferenceEngine.NextGoal*	LUT-FMS Research Group

Chapter 8
THE MODELLING KNOWLEDGE

8.1 Introduction

The main objective of this chapter is to introduce the modelling knowledge embedded in the modelling system. First, the work and tool flow in a single-cell flexible manufacturing facility and the corresponding system states are discussed. Then the information flow hierarchy is examined. After this, each of the three modelling levels is described in detail, with the emphasis being placed on the operational strategies considered at each modelling level.

8.2 Work/Tool Flow and System States

In a single-cell flexible manufacturing installation, the entities that flow through the system can be considered as two types. One is work, i.e. parts or components, and the other is tool (Figure 8.1). Although work and tool transporters, such as AGVs or robots, are also moving entities, they can be considered as being attached to work and tools since it is the work and tools that demand for transporters in order to be transported to their next destinations. The workstations like load/unload stations, machine stations, washing stations, inspection stations and the work/tool stores are all stationary elements, and they are requested by workpieces and tools.

Thus it is natural to view the system elements as being connected by workpieces and tools while they are flowing through the system. However, workpieces are usually carried by pallets, and tools are transported by tool transporters according to the parts' tool requirements at particular machine stations. As a result, a manufacturing system can be seen as having pallet and tool transporter flows. The state transformation of the system can then be effectively described through work pallets and tool transporters which link the other system elements.

In a normal event based simulation approach [209], an *event list* is required, where each event contains the time at which this event is due to occur, the identity of this event and the extra record to identify which entities are involved in this event. The disadvantage of this approach is that the event routines can get rather complex and this can make enhancement and debugging somewhat tortuous. In addition, decision-making is usually embedded in the event routines [243]. On the other hand, the activity based approach suffers from the fact that the activity scan is always complete and this is clearly inefficient and a waste of time. The process approach, however, usually requires a rather complex executive, and the processes are considerably complex to program [209].

Depending on the rules to be used for handling the interaction between work and tools, there are three basic work and tool integration strategies, i.e. work-oriented, tool-oriented and hybrid work and tool oriented (Figure 8.2) [84]. The work-oriented strategy can be defined as that tools are provided to meet the work's tool requirements. The tool-oriented strategy is at the other extreme, that is , work is scheduled through the system subject to the tool availability constraint at each of the machine stations. The hybrid strategy is a combination of the above two extremes, i.e. workpieces and tools can be influenced by each other.

8.3 The Four-Level Control Strategy

To deal with all these cases in the modelling system, it was decided that the control within the knowledge based modelling system should be facilitated in a hierarchical way, with four basic levels being defined, i.e. the inference level, the decision level, the state transformation level and the state description level (Figure 8.3). The inference level is the highest level, which is responsible for sequencing the actions and conflict resolutions which occur as the modelling proceeds. It controls not only the decision level but also the state transformation level. The decision level receives the conflicts arising at the transformation level, resolves these conflicts, detects and resolves further conflicts which can occur between the pre-conflict-resolved actions, and finally sends the non-conflicting decisions to the working memory which are then used at the transformation level.

The transformation level is the set of statements which describe the actions that make up the model. These are the explicit rules about the interactions of the cell elements. The state description level is the set of rules used by the transformation level to model the details of the cell elemments. It consists of rules for accessing and changing those particular attributes of the cell elements, and for collecting statistics. It has been designed to consist of a number of sub-description levels, i.e. rules have been written which can be called by higher level rules.

Within this hierarchical framework, work pallets and tool transporters are to be modelled independently at the state transformation level, with their interactions being handlled at the decision level. At the transformation level, it is adequate to use indexing variables [226] to represent the state of work pallets and tool transporters, and hence the state of the associated system elements. For example, if a pallet has a status value *Processing*, then this indicates that the workpieces on the pallet are currently being processed at the station pointed from the pallet, and the involved station is no doubt in the state *Busy* which means that the pallets waiting for this station can not be loaded. The advantage of this index method is that the status of system elements is explicitly represented, and this reduces the rule matching time significantly [226]. Another advantage of this method is that it is very easy to extend the modelling system to include the modelling of more complex behaviours. This can be achieved by adding new state indexes and defining the associated decision rules, transformational rules and descriptive rules.

The following is the list of variables defined in class ***Pallet*** (see Table II.27 of Appendix II for a complete definition of this class):

```
[DEFCLASS Pallet
    (MetaClass Class Edited:                          (*edited:          ))
    (Supers DPallet Entities)
    (InstanceVariables (AvailablePartTypes NIL doc    (* Part types that are available for pallet)
                       (SelectedPartType NIL doc      (*Part type that has been selected for
                                                      release)
                       (AllocatedParts NIL doc        (*Parts of the selected type that are
available                                                             for pallet))
                       (PartsOnPallet NIL doc         (* Parts that are on pallet))
                       (TransporterArrivalTime NIL
                                               doc    (* Time that the transporter runs from a
                                                      station to its next destination))
                       (SubOpsDuration NIL doc        (* Time to finish the sub-operations))
                       (FinishTime NIL doc            (* Finish time of pallet))
                       (StartTimeForTools NIL doc     (* Time when pallet starts waiting for
                                                      tools))
                       (GoalStatus NIL doc            (* Status of the gosl showing whether it is
                                                      active or not))
                       (CurrentLocation NIL doc       (* Location of pallet))
                       (NextOperation NIL doc         (* Operation that is onsidered fnext or the
                                                      allocated parts))
                       (OperationList NIL doc         (* Operaion list for allocated parts))
                       (CurrentOperation NIL doc      (* Current operation for parts on pallet))
                       (AvailableNextStations NIL
                                               doc    (* Next stations that are available))
                       (SelectedNextStation NIL doc   (* Selected next station for pallet when it
```

is

```
                                            transferred))
        (AvailableResources NIL doc   (* Available resources for considered
                                            operation))
        (SelectedResource NIL doc     (* Selected resource for considered
                                            operation))
        (AvailableToolTransporters NIL
                          doc         (* Available tool transporters for pallet))
        (SelectedToolTransporter NIL
                          doc         (* Selected tool transporter for the pallet))
        (TemporaryStorages NIL doc    (* Temporary storages for this pallet))
        (UnavailableTools NIL doc     (* Tools that are unavailable in the
PTS)))]
```

In the above definition, *FinishTime* in conjunction with *GoalStatus* are used to represent the goal of the simulation. *PartsOnPallet, CurrentLocation, NextOperation, OperationList, CurrentOperation* and *TemporaryStorages* are used to both represent the state of a pallet and link to other objects. The other variables can be classified as three groups. The first group of variables are *AvailablePartTypes, AvailableNextStations, AvailableResources,* and *AvailableToolTransporters,* whose values are set at the transformation level and are used by the decision level for making decisions (i.e., resolving conflicts). *SelectedPartType, AllocatedParts, SelectedNextStation, SelectedResource* and *SelectedToolTransporter* belong to the second group, which are set at either the decision level or the transformation level and are used at the transformation level for identifying the relevant objects whose states should be updated. All the others are the third group of variables which are both set and used at the transformation level, i.e., they act as temporary variables for passing information between the objects involved at this level.

Similarly, the class ***ToolTransporter*** has been defined to have the following variables for acting at the transformation level:

```
[DEFCLASS ToolTransporter
    (MetaClass Class Edited:                        (* edited:          ))
    (Supers DToolTransporter Transporter)
    (ClassVariables     (ResourceType ToolTransporter
                                      doc           (* Type of resource)))
    (InstanceVariables  (ToolsOnIt NIL doc          (* Tools that are currently carried by the
                                                    transporter))
                        (FinishTime NIL doc         (* Finish time of transporter))
                        (GoalStatus NIL doc         (* Status of the goal showing whether it is
                                                    active or not))
                        (ArrivalTime NIL doc        (* Time it takes for the transporter to
arrive                                              from its previous location)))]
```

Compared with the definition of *Pallet,* the above definition is much simpler. This is because of that the work and tool integration strategy under the present consideration is mainly work-oriented, though the influence of tooling on work can also be modelled. As a result, much information has to be associated with work flow, or rather, *Pallet*. Again, *FinishTime* and *GoalStatus* are used to represent the simulation goal. *ToolsOnIt* is used to represent the partial state of the transporter, and *ArrivalTime* is a temporary variable used at the transformation level.

In the previous chapter, the decisions made at the inference level have been closely examined. The following sections will concentrate on the logic of the other three levels, with emphasis beling placed on the decision level and the transformation level.

8.4 The Level-1 Modelling

8.4.1 Operational Assumptions of Level 1

As discussed in chapter 6, at this level, each work station is assumed to have a infinite local buffer so that no blockage could occur. Therefore, when a job is finished with its current processing, it can be unloaded immediately into its buffer. This means that the loading action, the actual processing action and the unloading action can be combined to form a single action. Equally, when a free transporter is requested to transport a job to its next destination, the transporter's empty run action, the transporter's loading action, the load run action and the unloading action can be considered as one action. Besides, temporary part storages, tooling availability and tool flow are not to be modelled at this level.

8.4.2 State Transformation at Level 1

As a result of the above assumptions, the actions of a work pallet at this level can be defined as the follows (Figure 8.4):

- ***Palletisation,***
- ***Processing,***
- ***Transporting,*** and
- ***Depalletisation.***

Here, *Palletisation* means the action of fixturing a batch of parts onto a free pallet and unloading the pallet from the station off to the local buffer at the load/unload station. *Processing* indicates the whole action involved at a pallet's visit to a machine station, inclusive of work loading, cuttings, tool changes and work unloading. The action *Transporting* involves the arrival of a free transporter, loading of a pallet onto the transporter, load run of the transporter and finally the unloading of the pallet off the transporter. *Depalletisation* includes the loading of a pallet from the local buffer of a load/unload station onto the station and the actual depalletisation of the workpieces.

The transformational rules which are built around the class *Pallet* for this level are contained in three rulesets called *TestStartofAction1* (Figures III.15), *StartAction1* (Figures III.21 to III.24) and *EndAction1* (Figures III.27 to III.30) (see Appendix III).

As shown in *TestStartofAction1*, before starting the four actions, the pallet can be in any of the four states, i.e., *AwaitingForPalletisation, AwaitingForTransfer, AwaitingForProcessing* and *AwaitingForDepalletisation*. In each of these states, the availability is checked of the relevant resources. Notably, the order of the four rules in this ruleset can be deliberate because the status values have been uniquely defined. On checking the availability of load/unload stations (i.e., when the status of the pallet is *AwaitingForPalletisation)* (Figure III.16), four rules have been defined corresponding to the four operational constraints:

Rule 1: *IF the station has an empty pallet (i.e., the system has a fixed number of pallets), THEN the station is set as the selected resource and the pallet is put to the working memory as an applicable pallet.*

Rule 2: *IF there is only one station assigned to the considered operation and it is available, THEN the assigned station is set as the selected resource and the pallet is put to the working memory as an applicable pallet.*

Rule 3: *IF there are alternative stations assigned to the considered operation but there is only one station available, THEN the available station is set as the selected resource and the pallet is put to the working memory as an applicable pallet.*

Rule 4: *IF there are alternative stations assigned to the considered operation and more than one stations are available, THEN the stations are set as available resources and the pallet is put to the load/unload station conflict set of the decision centre.*

When the pallet is in status *AwaitingForTransfer,* two relevant resources have to be checked. One is the next station (Figure III.18) and the other is the transporter (Figure III.19). When the status of the pallet is either *AwaitingForProcessing* or *AwaitingForDepalletisation,* the status of its residing station is checked (Figure III.20). The working logic of the rules in each of these three rulesets is similar to the above description for *.CheckLoadUnloadStation.*

After all the above identified conflicts are resolved in the decision centre, the pallet can start its transformation actions. This is defined in the ruleset *StartAction1* (Figures III.21 to III.24). On starting the palletisation action, the following changes have to be made with regard to the state of relevant objects:

Rule 1: *IF the pallet is awaiting for palletisation,*
THEN remove the pallet from the startable pallet list of the working memory,
change the state of the pallet,
collect part performance statistics,
set the new goal for the pallet,
change the state of the load/unload station,
collect the load/unload station performance statistics;

As shown in Figure III.21, the state transformation involves the updating of many variables and parameters. Therefore, metarules can be defined to group the relevant statements in a rule so that a more formal and comprehensible rule definition can be achieved. This has been falititated for the two later designed modelling levels.

The ruleset for setting the new goal for the pallet is shown in Figures III.25 and III.26.. When the pallet is in status *Palletisation,* the following time handling procedure is performed:

Rule 1: *IF the pallet is in palletisation,*
THEN set goal status to active,
compute the duration of the palletisation action,
set the finish time of the pallet to the addition of the current time and this duration;

The computation of the palletisation duration is done by ***Pallet.FindPalletisationDuration:***

THEN duration-.FindOperationTime + .FindStationExchangeTime;

That is, it adds the time of the considered operation with the time of the station exchange time. This coincides with the assumption of combining the palletisation and unloading actions together to form a single action.

When an action is due to end, the *EndAction1* ruleset is applied to change the state of relevant objects (Figure III.27 to 30). On ending the palletisation action, the following changes have to be made:

Rule 1: *IF the pallet is in palletisation,*
the current time as shown in the inference engine is greater than the finish time of the pallet,
THEN update the state of the pallet by placing parts on pallet and unloading the pallet to the local buffer of the station,
collect part performance statistics,
update the operation list of the pallet,
set the status of the pallet to AwaitingForTransfer,
set goal status to NIL,
change the state of the load/unload station,
update the statistics of the station;

Although not strictly a transformation action, the release of parts involves the identification of free pallets and the allocation of the parts to these pallets. Figure III.13 shows the ruleset for testing the release of parts. The actual release of parts is done in the ruleset *Pallet.ReleaseParts* as shown in Figure III.14.

8.4.3 State Description at Level 1

Many descriptive rules have been defined for application within the transformational rules. The objective of this is to simplify and formalise the statements used in the definition of a transformational rule. The following is the definition of a descriptive rule *Pallet.LoadOntoTransporter1* in rule 2 (Figure III.22) of *Pallet.StartAction1:*

THEN .Remove FromBuffer
.CollectPartsTimeAtBuffer
.CollectPartsLoadingUnloadingTime
:CurrentLocation-:SelectedResource
:CurrentLocation:PalletOnIt-self
:CurrentLocation.CollectLoadingUnloadingTime;

Here, direct data accesses are also used to set the pallet on a transporter. A sub-descriptive rule *.RemoveFromBuffer* is used to update the state of the buffer of the station. The other rules are for collecting statistics.

Another function of decriptive rules is to access the input parameters. Typical examples are the two rules *FindOperationTime* and *FindStationExchangeTime* used in *Pallet.FindPalletisationDuration* as shown above. Figure 8.5 shows the data input requirement for parts, pallets, load/unload stations, machine stations, work transporters and processes at this level. Notably, the links between objects in the data base are represented indirectly using litatoms rather than directly using objects. This resolves for the user the problem of complicated links among objects when designing a model.

8.4.4 Decision-Making at Level 1

There are five categories of conflicts that have been defined for this level of modelling. These are the *Release Conflict, Load/Unload Station Conflict, Next Station Conflict, Transporter Conflict,* and *Operation Conflict.* A release conflict is defined as the competition of several part types to be allocated to a free pallet (Figure III.13). The load/unload station conflict occurs when there are more than one stations for the initial fixturing of parts onto a pallet (Figure III.16). The next station conflict is defined as the availability of more than one stations capable of performing the next operation on a job (Figure III.18). The transporter conflict can be generated when there are more than one transporters available for a transfer action (Figure III.19). An operation conflict can be defined as the competition of several pallets for a resource (Figure III.9). This is the only conflict at this level which is detected at the decision-making level rather than at the transformation level. Refer to Appendix II for the class definitions of these conflicts.

Five categories of conflict-resolution rules or decision rules can be defined to resolve these conflicts respectively. These rules make decisions by accessing the information of the objects contained in the conflict sets. The release conflict-resolution rules select one of the available part types according to the orders information or the process characteristics of the part types (Table II.8). The load/unload station conflict-resolution rule selects a load/unload station according to the performance of the stations or the operation details on the stations (Table II.4). The application of the next station conflict-resolution rules chooses a station from several available ones for the job's immediate operation based on the performance of these stations or the job's status and process details at the stations (Table II.5). The transporter conflict-resolution rules are used to manage the movement of the transporters and they select a free transporter from several candidates for a job waiting to be transported to its next destination (Table II.12). The operation conflict-resolution rule selects a pallet from several ones for allocation to a resource, which can be a load/unload station, a transporter or a machine station (Table II.6).

Generally speaking, a decision rule can be developed based on two sources of information. One is the *local* information, i.e. the attributes of the candidates, and the other is the *global* information, i.e. the attributes of not only the candidates but also the objects linked to the candidates.

The standard form for a release conflict-resolution rule is as follows:

```
WorkSpace Class: ReleaseConflictSet;
Compiler Option: A;
Temporary Vars: partselected;
Control Structure: DOALL;
Rule Class: MyRule;
********

IF      partselected-(PickLowObj self 'GetAttribute :FilteredPartTypes)
THEN    :PartTypeSelected-partselected;
```

Here, the utility function *PickLowObj* is used to select a part type which returns the lowest value on applying the method *GetAttribute*. The method *GetAttribute* has to be defined by the user to retrieve the value of certain attribute of the part type. This attribute can either be a single parameter or a weighted combination of several parameters of the candidate, or its linked objects. An example of the single parameter is the batch size of a part type or its due date. The parameters of the linked objects can be the list of operations. Another utility function is *PickHiObj* which selects an item which has the highest value on applying the get-attribute function.

Notably, any other form of release conflict-resolution rules can be defined so long as the variable *:FilteredPartTypes* is used as the argument and the variable *:PartTypeSelected* is set to a candidate at the end.

The decision rules or conflict-resolution rules for the other types of conflicts take a similar form but have a different set of variables. Refer to the definition of these classes in Appendix II for a detailed description. In the case studies in later chapters, these rules will be developed and applied to control the operation of the cells.

Figure 8.6 summarizes the decision rules which have been built into the rule library so that the user can select for particular applications.

8.5 The Level-2 Modelling

8.5.1 Operational Assumptions of Level 2

At this level, each station has a specified buffer type of limited size(s). Therefore work can be blocked at a station, resulting in the necessity of considering the station unloading and transporter loading actions separately. The change of the state of a buffer or a station influences the jobs that are waiting for a position at the station. These jobs are either completed with their previous operations and waiting to be loaded off the table of the stations, or are at a station and waiting to be transported to its next station which has all its positions being ocuppied.

Similarly, the transporitng action at this level can not be modelled in the same way as level 1. Although the transporter's empty run sub-action can be combined with the pallet's transporter-loading sub-action, this combined action can not be further integrated with the transporter's load run sub-action and its unloading sub-action. For the loading of the transporter will free a position for the pallet's current station, and this can cause other jobs to start their actions. In addition, the use of temorary storages is to be modelled and tool requirements planning considered.

8.5.2 State Transformation at Level 2

As a result of the above assumptions, a work pallet can be defined to have the following actions at this level (Figure 8.7):

-***Palletisation,***
- ***WaitingforTransporterandLoading,***
- ***TransportingandUnloading,***
- ***LoadingandProcessing,***
- ***Processing,***
- ***Unloading, and***
- ***Depalletisation .***

At this level, *Palletisation* only indicates the action of fixturing a batch of components onto a free pallet at a load/unload station. *WaitingforTransporterand-*

Loading includes two sub-actions, i.e. the actual arrival of a selected free transporter from its current loacation to the pallet's current location and the loading of the pallet from either the station or the stations's buffer onto the transporter.

TransportingandUnloading involves the load run sub-action of the transporter and the unloading of the transported pallet off to its selected destination which can be either a station without any buffers or a station's buffer. *LoadingandProcessing* is the process of work loading, and cuttings and tool changes involved at a job's visit to the station. *Unloading* is the action involved for a job which is waiting to be unloaded off the station to its local buffer. *Processing* is applied when a job is transported to a station which has no buffers. In other words, after the job is unloaded to the station from the transporter during the *TransportingandUnloading* action, the loading sub-action in the *LoadingandProcessing* action is no longer necessary.

Depalletisation can be applied in two cases. One is that the load/unload station has a local buffer. In this case the execution of this action indicates the loading of the pallet from the buffer and the actual depalletisation of the work on the pallet. Another is that the load/unload station does not have any local buffers and the loading of the pallet to the station has already been done during the *TransportingandUnloading* action, and therefore *Depalletisation* only involves the depalletisation of the components off the pallet.

The transformational rules that have been built around the class *Pallet* for this level are defined in three rulesets known as *TestStartofAction2*, *StartAction2* and *EndAction2* respectively (Appendix III).

In modelling the use of temporary storages in the cell, rule 3 has been incorporated in the ruleset *Pallet.TestStartOfAction2* (Figure III.31). Here, the order of rules 2 and 3 is important. For the order shown in the figure implies the following operational constraints:

Rule 2: *IF the pallet is awaiting for transfer and the next station(s) is available, THEN check the availability of the transporter;*

Rule 3: *IF the next station is not available but the temporary storage(s) is available, THEN check the availability of the transporter;*

If these two rules are ordered another round, the pallet will be first checked against the temporary storages and then the requested stations. This will significantly influence the operation of the system.

8.5.3 State Description at Level 2

At this level, the parts and transporters information required for the modelling is identical to what is needed for level 1. The pallet description is similar to that of level 1 , but has an optional information requirement on the temporary part stores that the pallet can go to (Figure 8.8). The data input requirements for load/unload stations and machine stations are also similar to those for the level-1 modelling, except that the user is required to specify the type and size(s) for their local buffers. Additional information is required for part temporary stores and tools respectively. The process information at this level is much more detailed than that of level 1. A machining operation is defined as consisting of a list of sub-operations which are the tooling actions involved in the pallet's visit to the station. For each of the operations or sub-operations, the user can specify a tooling requirement in terms of a tool type identity.

The tooling input information at this level is used to carry out the tool requirements planning [50]. The tool requirements planning is conducted as an integral part of the level-2 modelling at the state description level. However, if the user does not want to do this experiment, he can simply ignore the tooling input requirements for the process information. The *descriptive rules* for carrying out this function are *ScheduleSubOps2* and *CellToolRequirement* in rule 4 (Figure III.39) and rule 5 (Figure III.40) of *Pallet.StartAction2*. *ScheduleSubOps2* is actually implemented as a Lisp procedure, which schedules each sub-operation, counts the tools required at the machine and returns the total operation time. *CellToolRequirement* is also a Lisp procedure which carries out the minimum tool requirement planning by updating the tool time list in the working memory.

From Figure 8.9, it can be seen that the tool requirements planning simply counts the tools required for the work scheduled to a particular machine station or to the cell. For each of the tools required for a job, a check is made of all the tools at the station or in the cell. If a tool of the required type is not available or is available but does not have enough tool life left for this operation or sub-operation, then a new tool of the required type is created for the station; if a tool of the required type does exist and remains enough life for the considered operation or sub-operation within the limit of the permissible life value, the data of that tool is updated.

For clarity, two basic types of buffers have been defined within the system. One is the part buffer which can take both the in- comming and out-going parts. Another is the pallet exchange store which consists of an input buffer for the in-comming pallets and an output buffer for the out-going pallets. For the pallet exchange store, the states of the input buffer and the output buffer are modelled separately. Notably, the above definitions can allow most of the real buffer types to be modelled. See later chapters for graphics representaions.

8.5.4 Decision-Making at Level 2

The decision points that are available at this level for entering control rules are identical to those of level 1 modelling, but those rules can be entered which take into account the position availabilities caused by the limited local buffers at the stations. For example, the next station conflict-resolution rules can be developed based on the spare positions available at each of the candidate stations (see the case studies).

8.6 The Level-3 Modelling

8.6.1 Operational Assumptions of Level 3

As mentioned before, this level takes into account the influence of tool flow on the performance of a manufacturing system. Although the additional consideration of tool flow also influences the work flow pattern of the system, the actions involved in the work flow are identical to those in the level-2 modelling.

For the tool flow, the tool transporter's empty run to the secondary tool store and the loading of tools onto the transporter at the STS can be combined into a single action. The actual transporting of tools from the STS to a primary tool store and the exchange of tools on the transporter with those at the PTS can be considered as one action. Equally, since the tools transporting action from the PTS to the STS and the unloading of tools from the transporter to the STS are executed sequentially, these can also be modelled as one action without affecting the performance of the system.

8.6.2 State Transformation at Level 3

As a result of the above considerations, the actions that a tool transporter takes can be defined as follows (Figure 8.10):

-***ArrivalandLoading,***
- ***TransportingandExchange,*** and
-***TransportingandUnloading.***

The *ArrivalandLoading* action means that a free tool transporter is requested by a pallet waiting for tools at a machine station to arrive at the STS and to load the tools according to the tool issue strategies [84].

TransportingandExchange is the process of tools' transporitng from the STS to a particular PTS and the exchange of these tools with the tools in the PTS subject to the constraint of the PTS's size. The following shows the tool change strategies at a PTS (rule set *ToolTransporter.ExchangeTools)*:

```
THEN .ExchangeWithWornTools
     .FillSparePositions;

IF   ThereAreToolsLeft
THEN .ExchangeWithUnusableTools;
```

As shown above, the tool exchange with PTS takes three possible steps. First worn tools in the PTS have to be taken away. Then if there are more tools, spare positions in the magazine should be filled. Finally, if there are still more tools to be exchanged, the tools which are not required by the next operation should be exchanged with those on the transporter. A tool is considered as worn if the cummulative cutting time of the tool has reached the maximum permissible limit [84]. The *TransportingandUnloading* action is very straightforward in a sense that all the old tools that have been loaded from the PTS onto the transporter in the exchange sub-action are just transported back to the STS, with the transporter being freed.

The transformational rules that are developed around *Pallet* for this modelling level are known as the following: *TestStartofAction3* (Figures III.53 and 54), *StartAction3* (Figures III.55 to 62) and *EndAction3* (Figures III.63 to 67).

Transformational rules that have been built around the class *ToolTransporter* are the following: *TestStartofAction* (Figures III.68 and 69), *StartAction* (Figures III.70 and 71) and *EndAction* (Figures 74 to 76). See Appendix III for a detailed description of these rules.

8.6.3 State Description at Level 3

The input information requirements for parts, pallets, work transporters, load/unload stations, part stores and tools are the same as those of the level-2 modelling, whereas the machine stations have an additional input requirment on the sizes of the PTSs (Figure 8.11). In addition, specifications have to be made of the tool transporters and the STS. The process information is also the same as that for level 2.

An example of the descriptive rules for a tool transporteris *LeavingForSTS* as used in rule 1 of *ToolTransporter.StartAction:*

Rule 1:	*IF*	*:CurrentLocation~=$InstWorkingMemory:SecondaryToolStore*
	THEN	*.CollectIdleTime*
		.CollectEmptyRunningTime
		:ArrivalTime-.FindTransferTime;
Rule 2:	*THEN*	*.CollectIdleTime*
		:ArrivalTime-0;

In the above ruleset, the second rule corresponding to the case where the tool transporter is already at the STS, and therefore the arrival time of the transporter is set to 0.

8.6.4 Decision-Making at Level 3

As shown in Figure 8.12, the interactions between work and tool can take two forms. The first is the influence of tool on work, where workpieces are usually scheduled to machines subject to tooling availability [63] [238]. In level 3 modelling, this is handled by the next station conflict-resolution rules. The second is the influence of work on tool, where tools are issued from the STS to the machine PTSs according to the the workpieces scheduled to the machines [84] [304] [146]. To model this aspect, tool issue strategies have to be carefully defined. De Souza has studied a cross section of tool issue strategies, such as *total tool changeover, tool kit, differential kit, single tool(s), resident kit, functional tool number, and tool cluster set* [84]. In level 3 modelling, the *differential kit* strategy has been incorporated at the present time.

With *differential kit* strategy, tools are issued in kit for each pallet of parts, but common tools can be shared between tool kits at the same machine. Tool transporter will transfer the tools that are not available in the PTS of the machine for the next pallet. The use of this strategy can reduce the tool requirement [238].

As mentioned before, the interactions between work flow and tool flow has been modelled at the decision level (Figure III.11). The *DecisionCentre.DetectTooling Conflicts* is defined as follows:

Rule 1:	*THEN*	*(- self ExecuteObjects 'CheckToolsAvailability $InstWorkingMemory: StartablePallets);*
Rule 2:	*THEN*	*(- self ExecuteObjects 'TestStartOfAction $InstWorkingMemory: ToolTransporters;*

The first rule checks the tools present in the PTS for the pallet waiting to be loaded onto a machine and generates the tool requirements by identifying *worn tools* . A *worn tool* is defined as a tool whose used life has reached the maximum permissible life. The second rule is used to identify whether there are pallets waiting for tool transporters, and whether there are any conflicts with regard to the requesting of a transporter by several pallets.

There are two additional conflicts available for the user to define rules at this level (Figure 8.13). One is the tooling conflict and the other is the tool transporter conflict. The former occurs when there are more than one pallets waiting for tools fir their immediate operations. The latter is the case when there is a choice on the selection of a tool transporter form several available ones. Definition of the decision rules or conflict-resolution rules is similar to that of the other rules.

Besides, the decision rules for work flow can be developed which take into account the tool availability constraint at the machine stations. For example, the next station conflict-resolution rules can be built according to the number of tools available at the PTS in comparison with the tools required for the next operation.

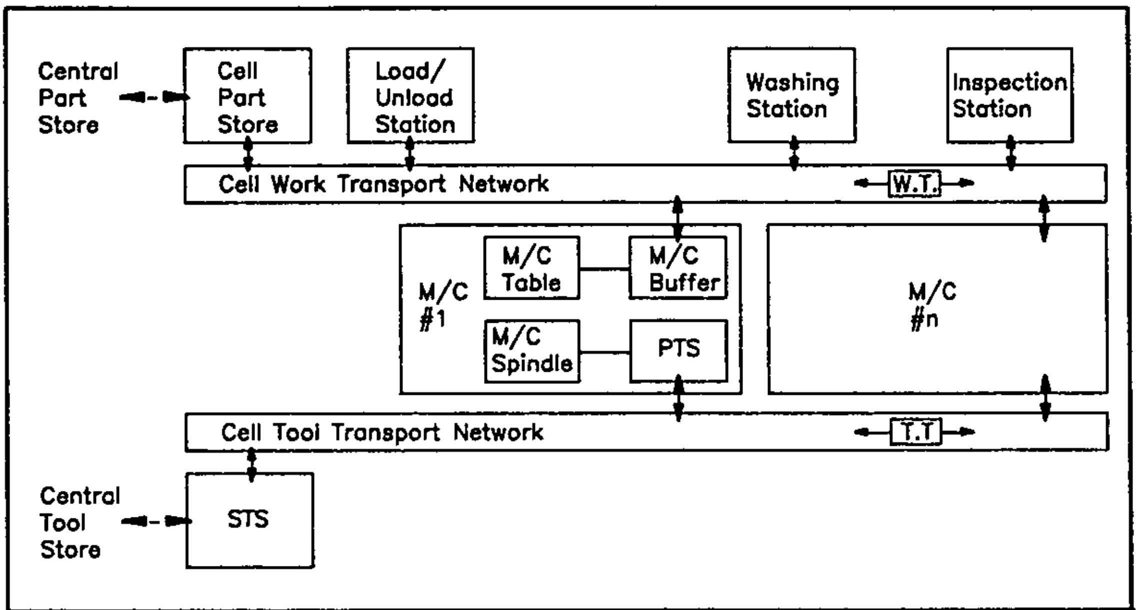


Figure 8.1 — *Work and Tool Flow in an FMC or FMS* — LUT - FMS **Research Group**

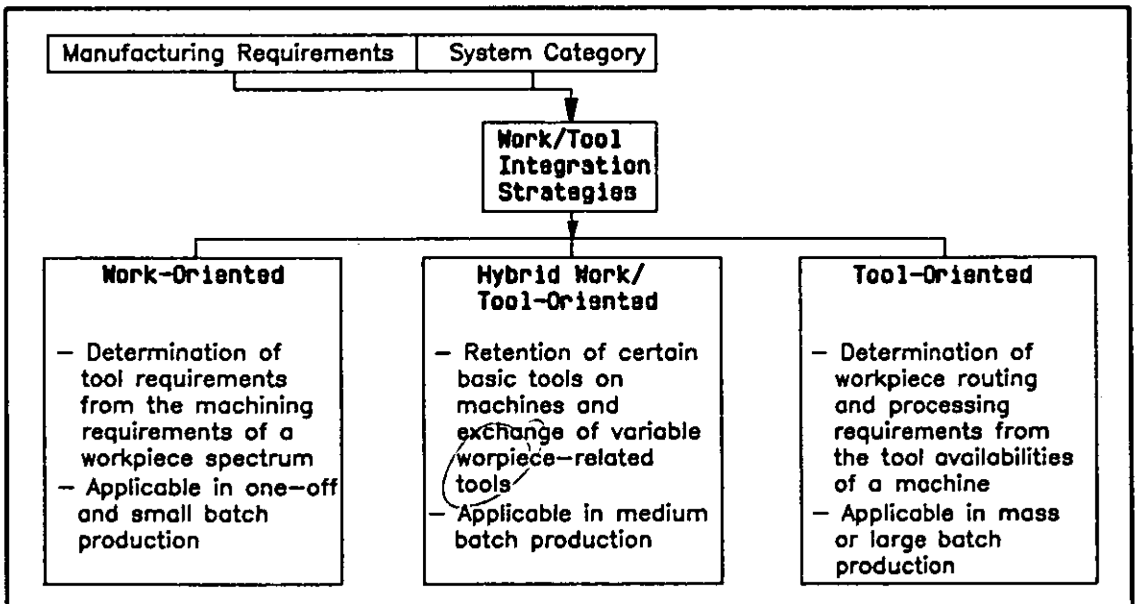


Figure 8.2 — *Work and Tool Integration Strategies* — LUT - FMS **Research Group**

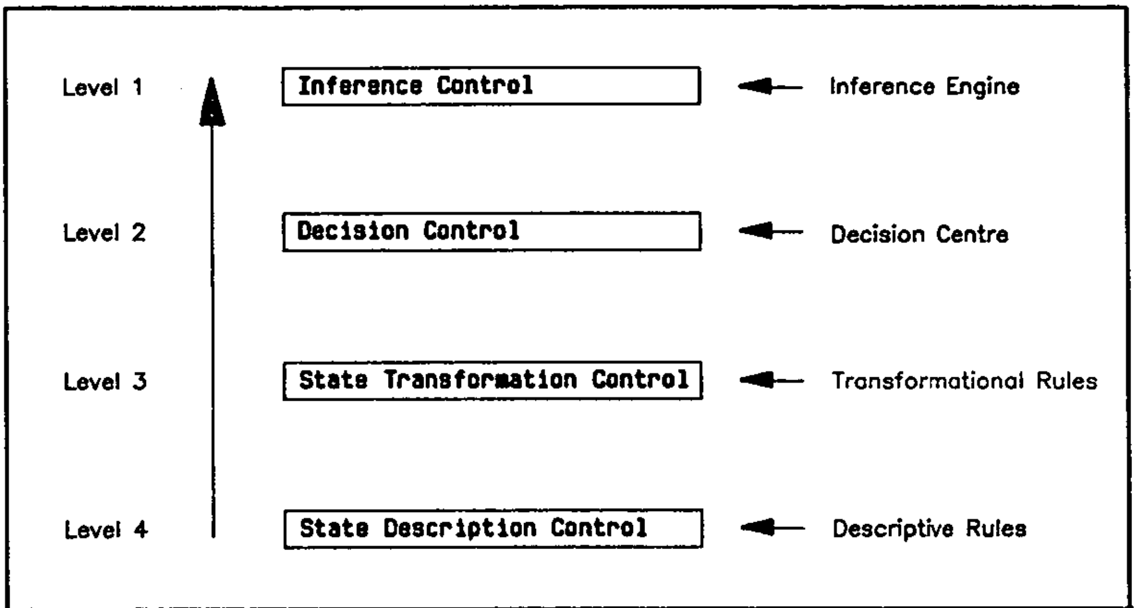


Figure 8.3	*Control within the Knowledge Based Modelling System*	LUT - FMS **Research Group**

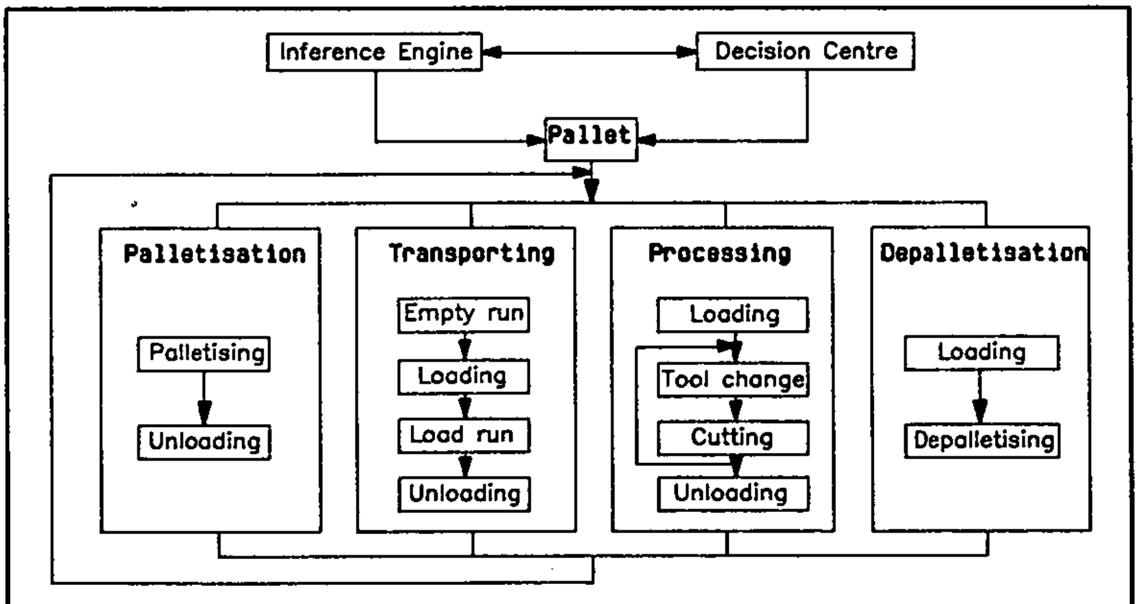


Figure 8.4	*Actions of Work Pallet in Level-1 Modelling*	LUT - FMS **Research Group**

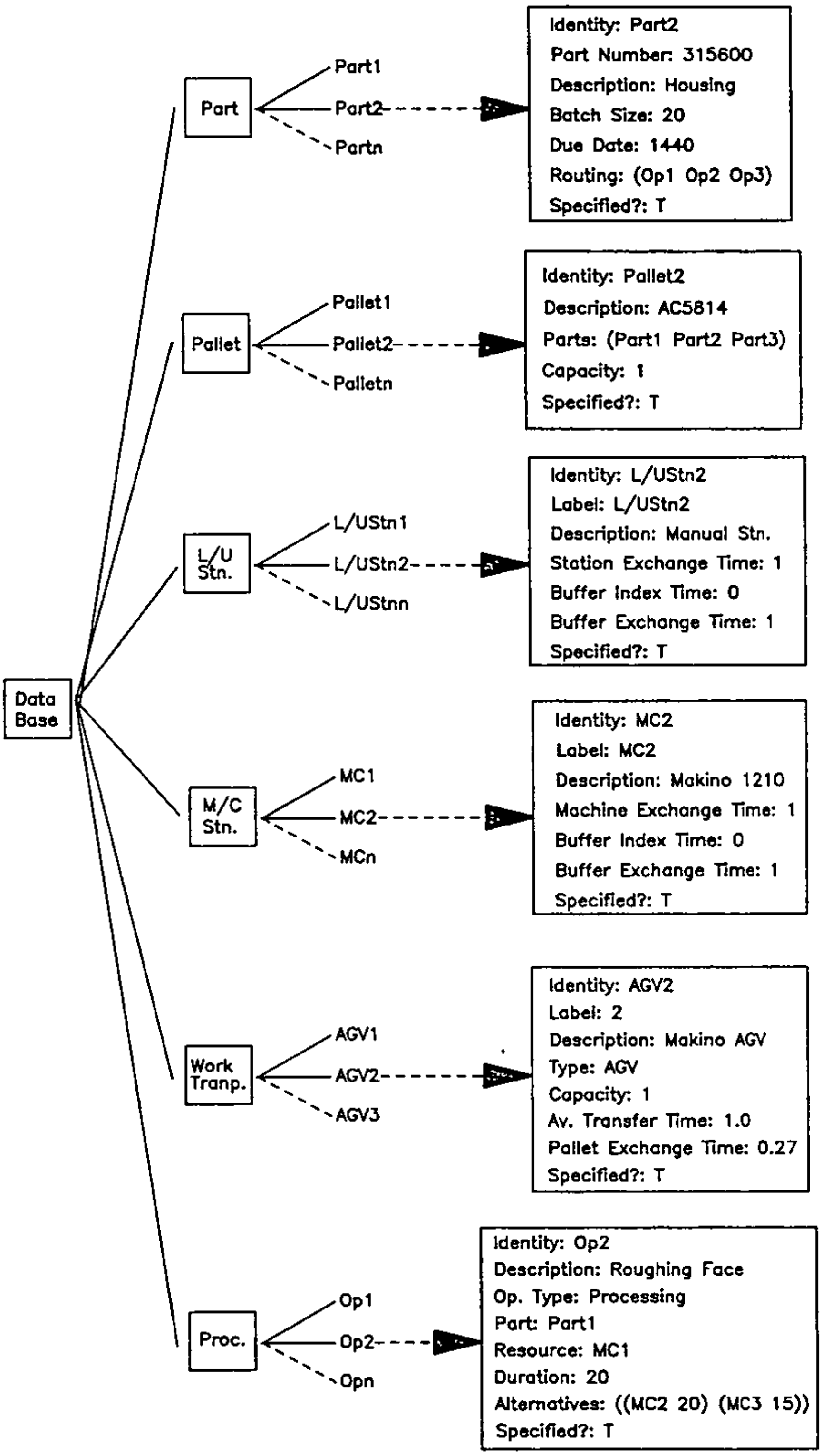


Figure 8.5	*Data Input Requirement of Level 1 Modelling*	LUT-FMS Research Group

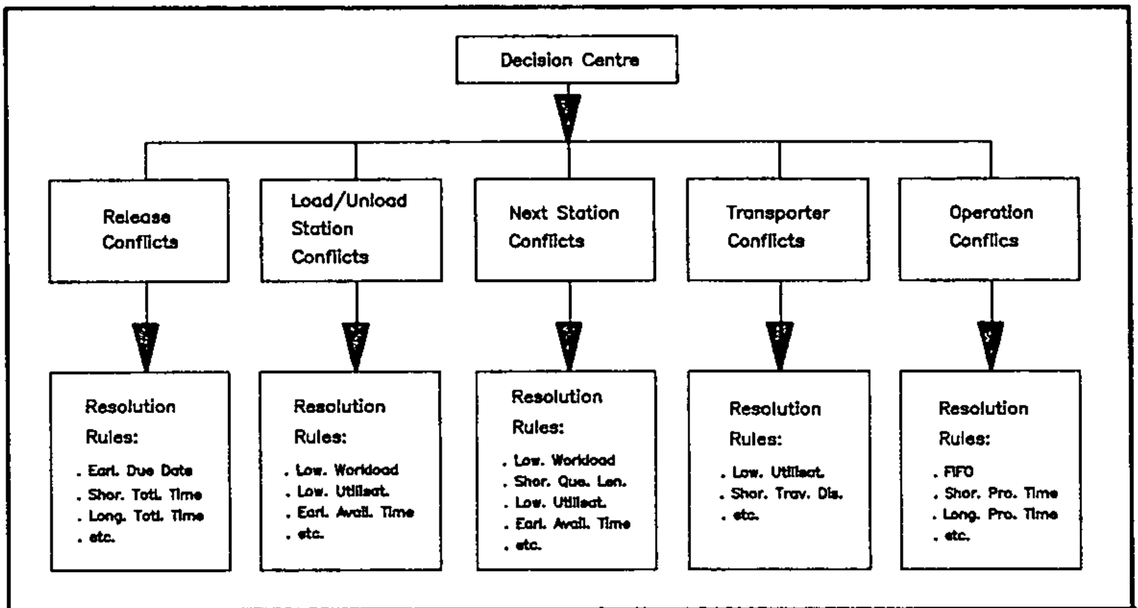


Figure 8.6	*Conflicts and Resolution Rules in Level-1 Modelling*	LUT - FMS Research Group

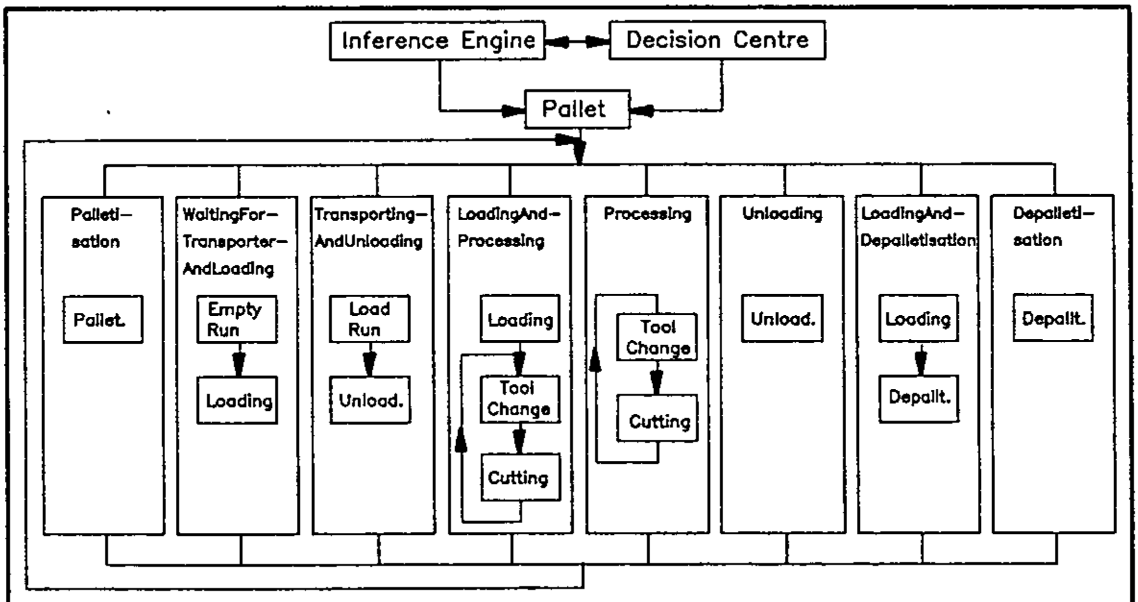


Figure 8.7	*Actions of Work Pallet in Level-2 Modelling*	LUT - FMS Research Group

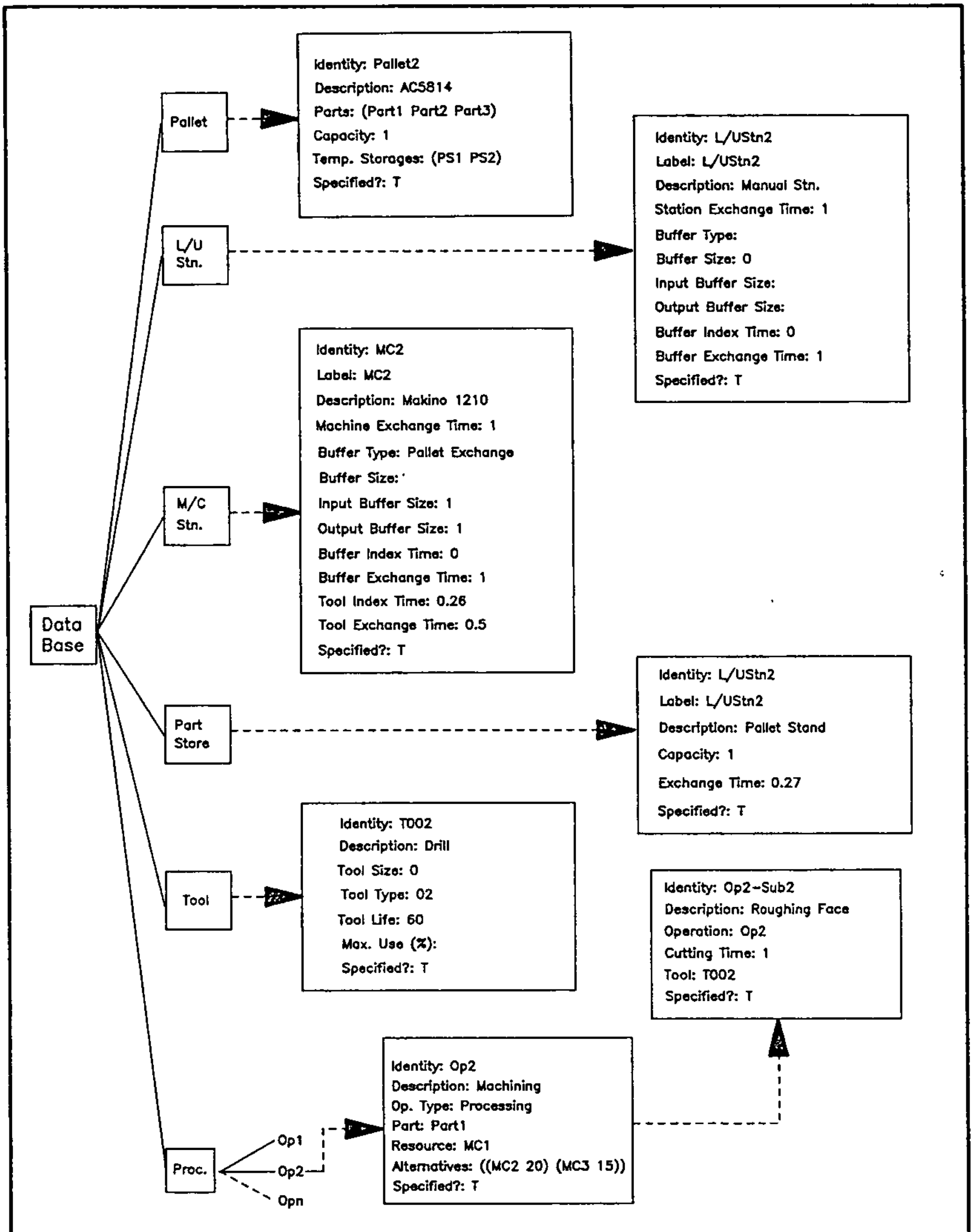


Figure 8.8	*Additional Data Input of Level 2 Modelling*	LUT-FMS Research Group

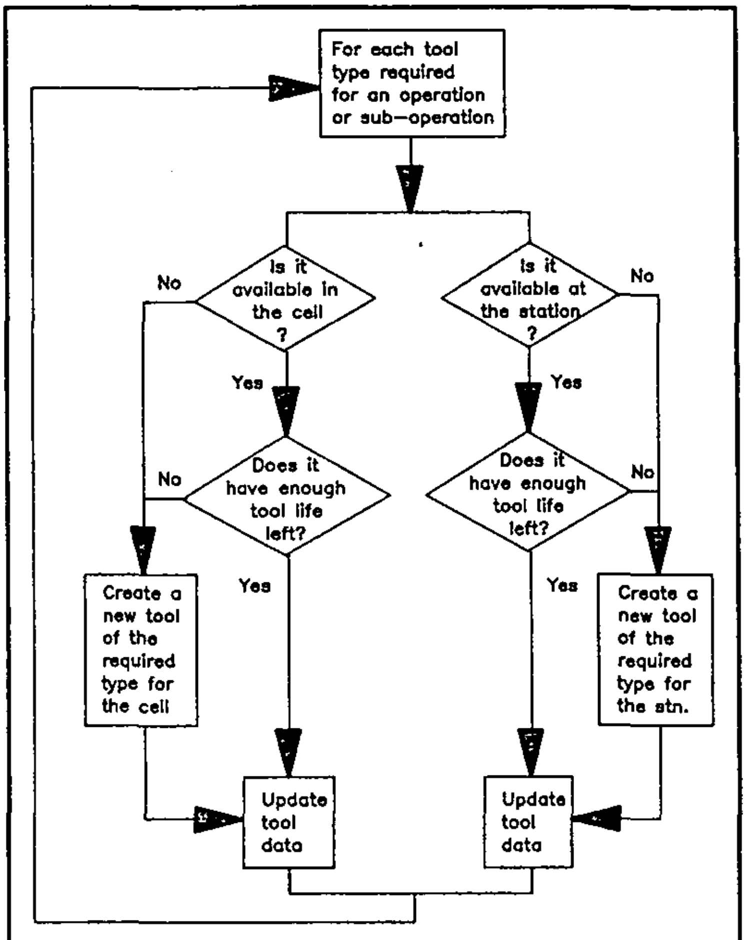


Figure 8.9 | *Logic Flow of Tool Requirements Planning* | LUT-FMS Research Group

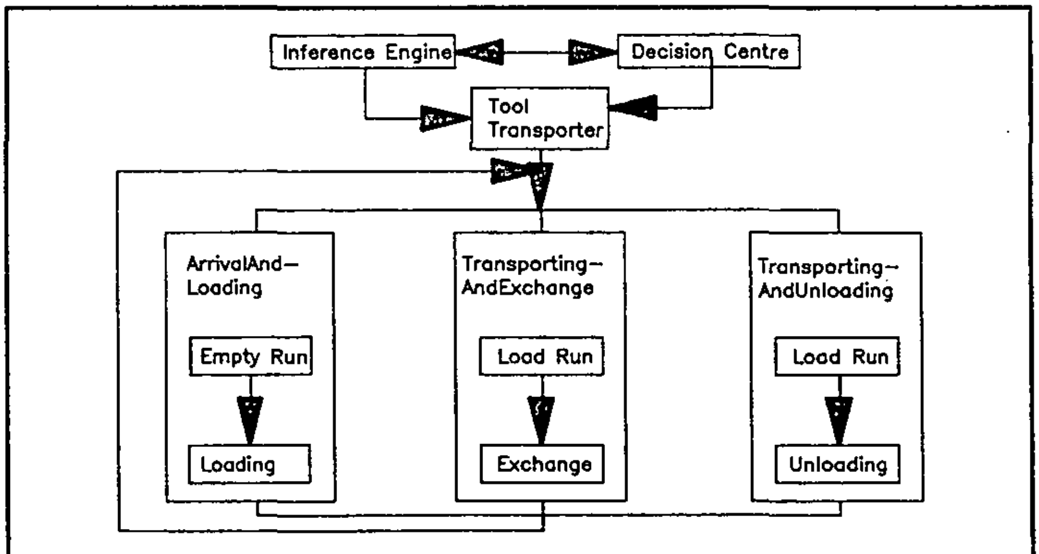


Figure 8.10 | *Actions of Tool Transporter in Level-3 Modelling* | LUT - FMS Research Group

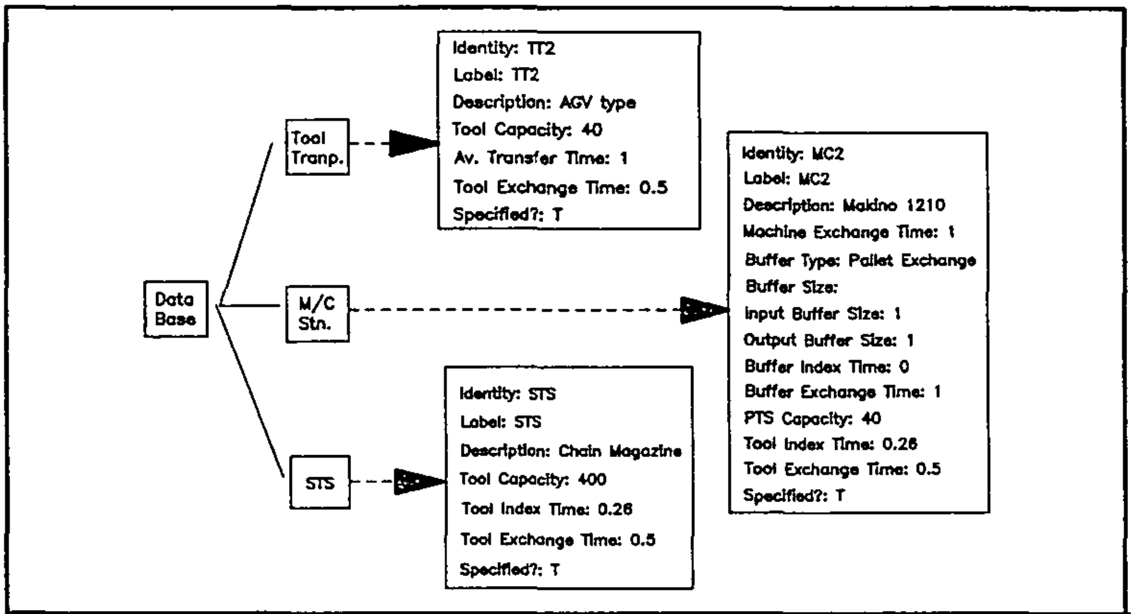


Figure 8.11 | *Additional Data Input for Level 3* | **LUT - FMS Research Group**

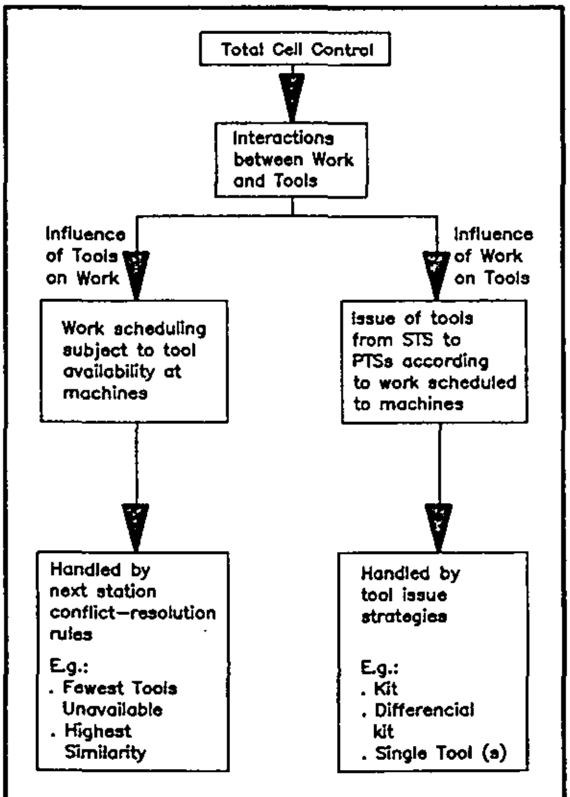


Figure 8.12 | *Work/Tool Interactions in Level-3 Modelling* | **LUT-FMS Research Group**

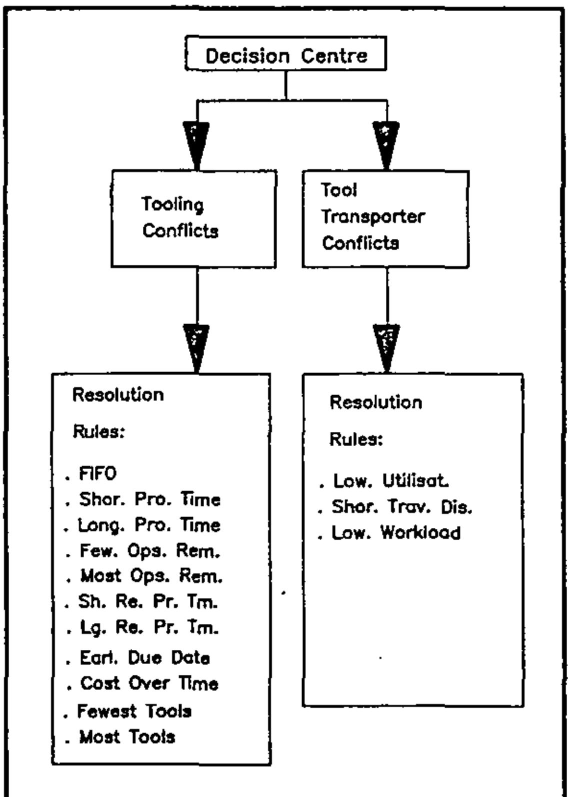


Figure 8.13 | *Additional Conflicts in Level-3 Modelling* | **LUT-FMS Research Group**

Chapter 9
AN OVERVIEW OF THE USER INTERFACE

9.1 Introduction

This chapter presents an overview of the user interface of the modelling system. First the user requirements of the interface is discussed. Then the overall structure of the user interface is described. After that, the facilities which have been developed according to these requirements are briefly illustrated. These facilities include the global menus, the model configurator, the data base browser, the interactive rule entry editor, and the graphics and textual output facilities.

9.2 User Requirements of the Interface

It is well recognised that the user interface can often be a critical feature of a knowledge based system. A system is not likely to be of much use if the user can not use it because of an ill-conceived interface.

As shown in Figure 9.1, the first requirement of an interface is that the user involvement with the system should not be either too little or too much at all three stages. This is especially true in the modelling of manufacturing systems, where both manufacturing cell data and operational rules need to be entered, but the collection of data and definition of rules are non-trivial tasks [168]. Thus appropriate options should be provided which are adequate for the design of a model for a manufacturing system but are not cognitive overload to the user. As for the existing modelling tools, generalised simulators enable quick configuration of a model with the user-specific data but operational rules can not be conveniently entered by the user [164] [43]. On the other hand, general purpose simulation languages like SEE-WHY [10] and SIMULA [131] usually require extensive programming in order to develop a model for a manufacturing system. This may be overloaded to the user in some cases, because he has to know some special programming constructs which do not have any manufacturing meaning.

Thus the interface of this knowlegde based modelling system for manufacturing system design is constructed with special considerations being given to the above factors. Depending on the programming experience of the user, several choices are available for the building of models, i.e. purely data driven, data driven plus rule selection, data driven plus rule modification, and data and rule driven. In addition, two levels of rules have been defined for user's manipulation, i.e. decision rules in manufacturing terms and behavioural rules in modelling terms. By providing all these flexibilities, the user can model a system according to both the modelling objectives and his modelling experience.

The most important requirement may be the functionality of the interface, i.e. the tasks of the user as reflected in the interface [263]. So far as manufacturing system modelling is concerned, a user needs to interact with a modelling system at three basic stages, i.e. design of a model, running of the model and presentation of the results. When designing a model for a flexible machining facility, the interface should provide options for the cell physical layout, specification of the cell data and entry of operational rules. In the running of the designed model, graphics and textual facilities showing the performance and behaviour of the model are required in order for the user to understand the operation of the model. Presentation of the results need also be performed in a form which is effective for the subsequent evaluation and analysis.

In addition, the interface needs to be flexible in use so that the user can easily access any part of the data base and knowledge base. Friendliness [164] [43] is another requirement of a user interface, i.e. explanations should be provided against the selection of an option from a menu of available ones, and any prompts for the user's actions must be technologically meaningful. As addressed by Schanehchi [240], although user-friendliness reduces development time and increases the general acceptance of a technical package, it may well be conflicting with the generality of application and the flexibility that the package should offer. Therefore, both of these contrasting requirements need to be considered. Besides the interface should be adaptable, to some extent, to new options so that they can be conveniently included.

It is obvious that any user's requirements are restricted by the hardware and software to be used. Since this work is based on the Xerox Workstation [302] and LOOPS software [53], the capabilities which can be provided by the user interface have to be confined to the facilities like windows, menus, browsers and bitmaps, etc. [302] [53].

9.3 Overall Structure of the Interface

With the facilities [302] provided by the Xerox Workstation, the interactions between the user and the modelling system can be of three types: menu driven, natural language based and graphics based [242]. Among these, the menu driven interaction is the easiest and the user is most confident when he is guided through the system. However, if all menus are displayed on the screen without logical organization, this friendliness will be lost. Therefore, an organized suit of menus should be provided which can guide the user throughout the whole modelling process. The facility which has been developed to play this role in the modelling system is called *the global menus*. As shown in Figure 7.1, the global menus should act as a control centre for accessing all the other interface facilities.

To allow physical configuration of a model, *the model configurator* has been developed which is based on both menus and graphics capabilities [123] [32]. *The data base browsingand editing facilities* have been designed to offer all the required options for the entry and management of machining cell data [123]. It is highly menu driven, but also has limited graphics features as mainly provided by the *LatticeBrowser* of LOOPS [53]. *The rule entry editor* is a highly interactive facility which incorporates both menus and natural language based editors [178]. The menus are used to access the appropriate decision points, inspecting the structure and parameters of a particular decision point and invoke the natural language rule editor. The natural rule language editor is provided by the LOOPS package.

The graphics and textual output facilities are a suit of sotware which involves all three types of interactions [223]. The graphics facility is used to setup the dynamic screen display for model running. The natural language facility is used for tracing rules and for interactively querying the relationship between the behaviour of an object and the rules applied on the object. *The statistics output facilities* are responsible for the static presentation and retaining of simulation results. They are mainly based on windows supported by limited menus.

9.4 The Global Menus

The global menus are a suite of menus which act as the control centre for all the user's activities with the modelling system. They provide the user with options to state the modelling level entry, create or choose a model, edit a model, run a model and present the results (Figure 9.2). The options the user needs for manipulating model layout configuration, model data specification and editing, decision and behaviour rules entry, and graphics and textual outputs, are available through the other menus.

The global menus consist of a hierarchy of menus for the level entry, for accessing the edit model menu, the run model menu and the output results menu at a particular modelling level, and for invoking all the options on the three menus. Flexible invoking of all the menus is possible, with both visual and audio help provided against the selections.

The first menu in the global menus is *The Modelling Master Menu* which provides all the options for entering any of the three modelling levels. Help options are also available for explaining the selection of an appropriate modelling level for particular applications. In addition, option is provided for the user to invoke the *Class Browser* associated with the modelling system.

With regard to each modelling level, there is a modelling master menu which controls the access to the corresponding menus for model editing, running and outputing results. *The Edit Model Menu* contains all the options for the design or modification of a model, including access of the Model Configurator, storage and retrieving of a model, hardcopying of model input information, specification of top level system parameters, and invoking of the rule entry editors for both behavioural and decision rules.

The Run Model Menu provides the options the user needs to experiment with a designed model. There are options for creating a *working memory* for a particular model, setting up graphics and textual output facilities, initialising and running the model. Other options are for handling the clock gauge, the utilization gauge and the trace window; blocking, suspending, and stopping the run.

The Output Results Menu contains the options for displaying and retaining the results of a particular run. These include the run time, cell performance, part throughputs, part lead times, part performance, machine performance, work transporter performance, load/unload station performance, station operation schedules and part operation schedules for the level 1 modelling; and the temporary storage performance, the minimum tool requirements, the PTS tool requirements and the tool performance as additional outputs for level 2, and the temporary storage performance, the cell tool requirement, the PTS performance, the STS performance, the tool transporter performance and the tool performance as additional outputs for level 3.

9.5 The Model Configurator

The model configurator is invoked from the global menus, which provides the user with all the facilities for physically configuring a manufacturing cell layout. The user configures a cell layout by interactively selecting the defined icons of the major cell elements, placing them on the screen, drawing the routes of the transporters and finally connecting the icons with the routes by adding paths [10] [123] [32] (Figure 9.2). Each of the graphics items on the screen is a LOOPS object which is automatically placed into the data base browser on creating.

There are three menus associated with the configurator, i.e. the ***Title Bar Menu***, the ***Configurator Object Menu*** and the ***Configurator Class Menu***. The ***Title Bar Menu*** is primarily concerned with creating icons on the screen and assisting in positioning the icons. It also provides the option for invoking the data base browser. Other options available on this menu are for defining a specialized class for an existing one, setting up grid facilities, measuring distance or angle within the configurator, shifting all the items on the screen, flashing the regions occupied by the objects and inspecting the configurator by displaying the associated parameters.

The ***Configurator Object Menu*** provides options for manipulating the objects or icons on the screen. Different options may be displayed for varying objects, but they are mainly for moving, changing, deleting and examining objects. Some objects have the options for adding, moving and deleting paths which connect the object with the routes defined. This menu is displayed when the user left clicks over a selected object on the screen. Objects are selected by clicking over them. Selected objects are marked by reversed corners of their images. Only one object can be the selected object at any one time. Selecting an object de-selects any previously selected object. Other options on this menu include copying object, changing label of an object, changing image of the object, aligning buffer and the main images, stretching the routes and setting the line width of the routes.

The ***Configurator Class Menu*** provides options for the user to edit and change the class definition of a selected object. Notably, since the change of the class definition amy affect may other objects in the modelling environment and other models, it is recommended that the user always specializes a class and edits that specialization.

The options on this menu include editing a class definition, adding a new method to a selected object's class, editing an existing method, changing the default image of a class, changing the default appearance of the routes on the screen and changing the default dashing of the routes.

9.6 Data Browsing and Editing Facilities

The data base browser is invoked form the title bar menu of the model configurator. It provides the user with the options for browsing and editing the static data associated with the production requirements, the machining cell data and the process data. A similar approach has been used by KEE [161] for managing instance objects. It is constructed by specializing the LOOPS class *LatticeBrowser* into the class *DBaseBrowser* through re-defining the three class variables: *MiddleButtonItems, LeftButtonItems, and TitleItem* and the required functions.

Each of the items in the browser has actions associated with the left and middle mouse buttons. When either button is clicked over an item, a menu of options is brought up, adn the user can make a selection of them. These options are mainly concerned with the specification, editing and displaying of the information of the objects, and the addition and deleting of objects and object colletors. In addition, there are options for manipulating the browser itself with the title bar menu (Figure 9.3).

The *Title Bar Menu* provides the options for creating a new data base browser or setting up sub-browsers of the existing browser. Depending on the selection of different modelling levels, the options vary. Setting up of sub-browsers does not dump any other objects shown in the browser, but eases significantly the management of the objects when the browser becomes considerably large.

The *Summary Menu* is displayed corresponding to the left mouse button. At the present time, only one option is available on this menu, which prints a summary of all the variables and their associated values of an object. This helps the user to edit or change the values of the parameters.

The *Edit Menu* contains all the options for editing and changing each of the item in the browser. There are different actions associated with the options on this menu when clicking over the root dat base icon, the item stacks or the instance objects. The edit menu for the data base icon povides the options for adding, deleting or specifying the item stacks. The edit menu for an item stack icon contains the options for the user to create, delete or clear instance objects belonging to the type designated by the stack. The edit menu for an instance object in the browser provides all the options for specifying, editing and inspecting the object. When an option is selected, the user is prompted to type in the desired values for the parameters in the prompt window of the browser.

To help retain the information in the data base, functions have been written to print the object related data in a text file, which can then be subject to hardcopying on the linked printer.

9.7 The Interactive Rule Entry Editor

The interactive rule entry editor of this modelling system has been developed to allow users to enter behaviour rules of cell elements and decision rules with regard to the operation of the cell. With this facility, the user can select a rule for the model to apply from a library of existing ones, modifying a particular rule, or express a new rule within the LOOPS rule language editor.

As shown in Figure 9.4, when entering rules by selecting, decision points with regard to the operation of a model have to be identified and rules concerning each of these points have to be established [165]. In this modelling system, these decision points are designed as conflict sets and the user invokes the rule entry editor through the edit model menu at each of the modelling levels.

When selecting a rule option from the edit model menu at any of the three levels, a library (or menu) of existing rules with regard to the decision point are displayed beneath the global menus, and the command menu is attached to the right-bottom of the rule library menu. To select a decision rule for a particular decision point, left click over an option in the rule library. When a rule is selected, it is shaded into black in the library, and this updates the command menu of the decision point.

The *Command Menu* contains the options for explaining a particular selected rule, editing a selected rule or expressing a new rule. To actually edit a selected rule, the user needs to have certain knowledge about the rule editor. In addition, the user needs to inspect the class around which the edited rule is built. This is done because the user has to know the variables that are accessable within the rule. Besides, the user should have experience with the LOOPS rule langauge so that he can write statements using the variables of the class or any arguments in the rule editor window.

When entering rules by expressing, the user requires the rule language, decision points and data access methods as the supporting facilities. In other words, the user has to know the variables of the decision point class. This can be done by inspecting a decision point or even editing a decision point through the command menu. After that, the user can select the *Add New Decision Rule* option. This results in the opening of the rule editor window and the user can use the LOOPS rule language [53] to express his/her own decisions. After defined, the new decision rule is placed into the rule library automatically. To define a rule which can be called within the new decision rule, the user can define a supporting rule by selecting the *Add Auxiliary Rule* option.

The other options of the command menu are for deleting a decision rule or an auxiliary rule and stopping the entry of rules.

Another option, which is available on the edit model menu for each of the three levels, is *Enter Behavioural Rules*. This option is suggested to be invoked only by the experienced users of the modelling system since it is concerned with all the methods which have been defined around each of the cell element at each level [240].

When the user selects this option, a library of classes which have been defined for a particular level are displayed as a menu. If the user selects one from this menu, a library of rules for this class are displayed beneath the class library, with the command menu being attached to the rule library. The options available on the command menu include inspecting or editing a class, explaining, editing or deleting a selected rule, and adding a new rule.

9.8 Graphics and Textual Output Facilities

When running a designed model, the user needs to be supported by the facilities which can demonstrate the operation of the model in order to understand the behaviour of the model [242]. Additionally, the user needs facilities which can show the relationship between the behaviour of the model and the computer code behind this behaviour [178].

In this modelling system, software has been developed to allow the user to setup windows to show the status of each of the major cell element (Figure V.18). This methodology is similar to the approach used by KBS [223]

When the user selects the *Setup Display* option from the run model menu for each of the three levels, the status windows for pallets, load/unload stations, machine stations, work transporters, temporary storages, tool transporters and the secondary tool store. In addition, there are two windows showing the current time and the next event (or next goal in AI terminology).

Once the status windows are displayed, the values of the variables depicted in the window can change dynamically during the running of the model. To query about the state of a particular object, the user can suspend the modelling and invoke the rule executive window [53] of the selected object. The user can then ask about the state of the object by typing *Why* followed by a particular variable in the window. This leads to the rule to be displayed, the application of which has caused the object to change to the current state as shown in the status window. This explanation facility helps the user to understand the relationship between the computer code and the immediate behaviour of the model and to debug or modify the model.

Another facility which aids the user to examine the operation of the model is the modelling trace option. Since the changes of the behaviour of objects are represented using rules, they can be traced during the running of the model. This is achieved by first selecting the *Trace Modelling* option on the run model menu. To trace the major behaviour rules of a particular instance object, select the *Trace Item* option from the pop-up menu of the status window of the desired object. Then the rules which are applied by the selected object during the modelling will be displayed in the trace window. In addition to the screen displays for a cell element, various LOOPS gauges can be attached to the specific parameters of particular objects.

9.9 Statistics Output Facilities

The statistics output facilities have been developed to statically present the results of a simulation run on the screen, and to retain these results in appropriate files so that they can be printed on the linked printer.

Although there are many possibilities for this purpose, such as the bar charts, line graphics, histograms, scatter diagrams, pie charts, etc. [164] [242], the facilties which have been developed at this stage have taken the text form. This is achieved by printing the statistics collected by the working memory in the summary window. Two menu options, *More* and *Abort*, are provided to help scroll the results in the window.

To retain results, functions have been written to open a text file and to print the desired results in the file. This file can then be hardcopied if so wish.

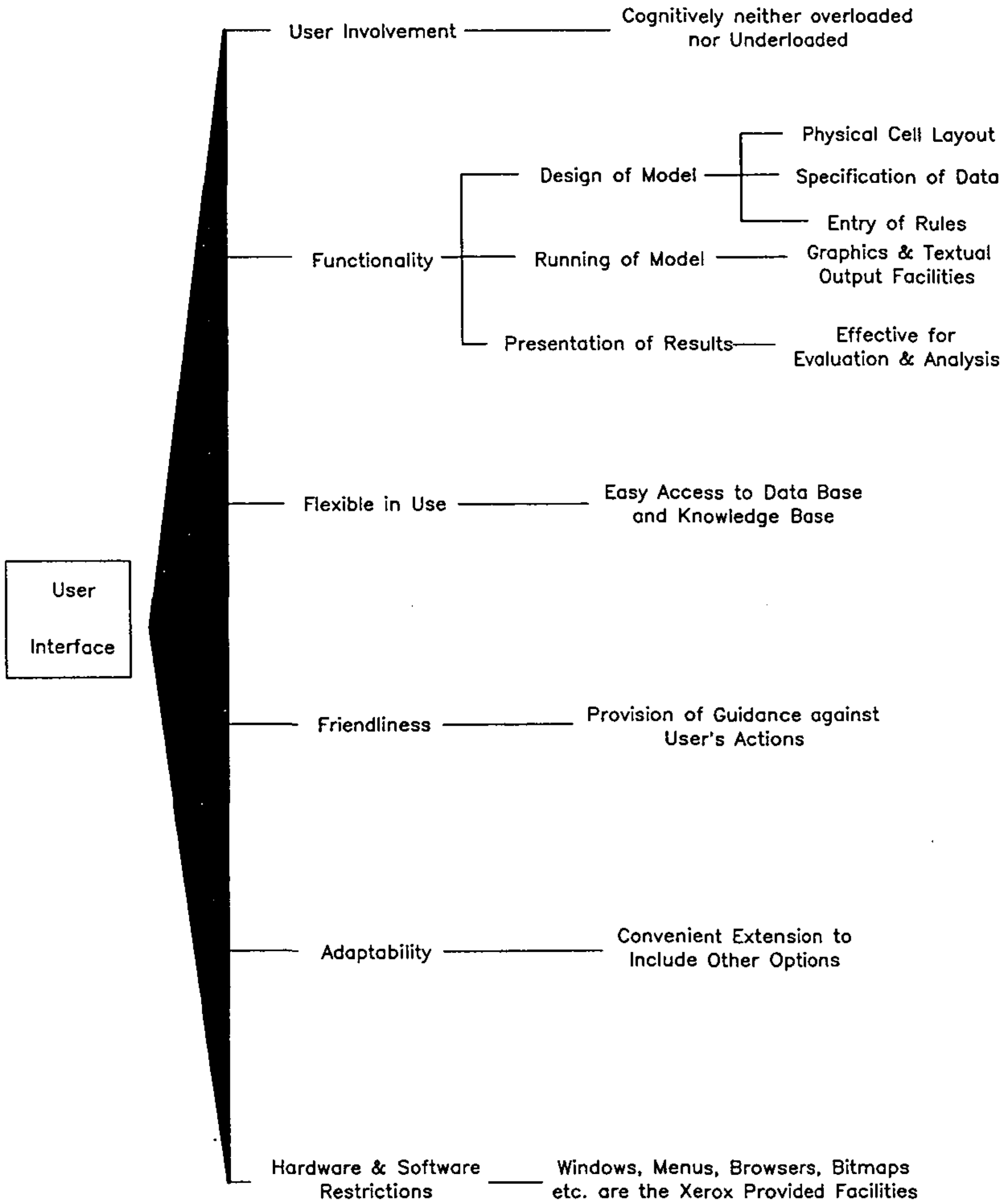


Figure 9.1	*User Requirements for the Interface*	LUT-FMS Research Group

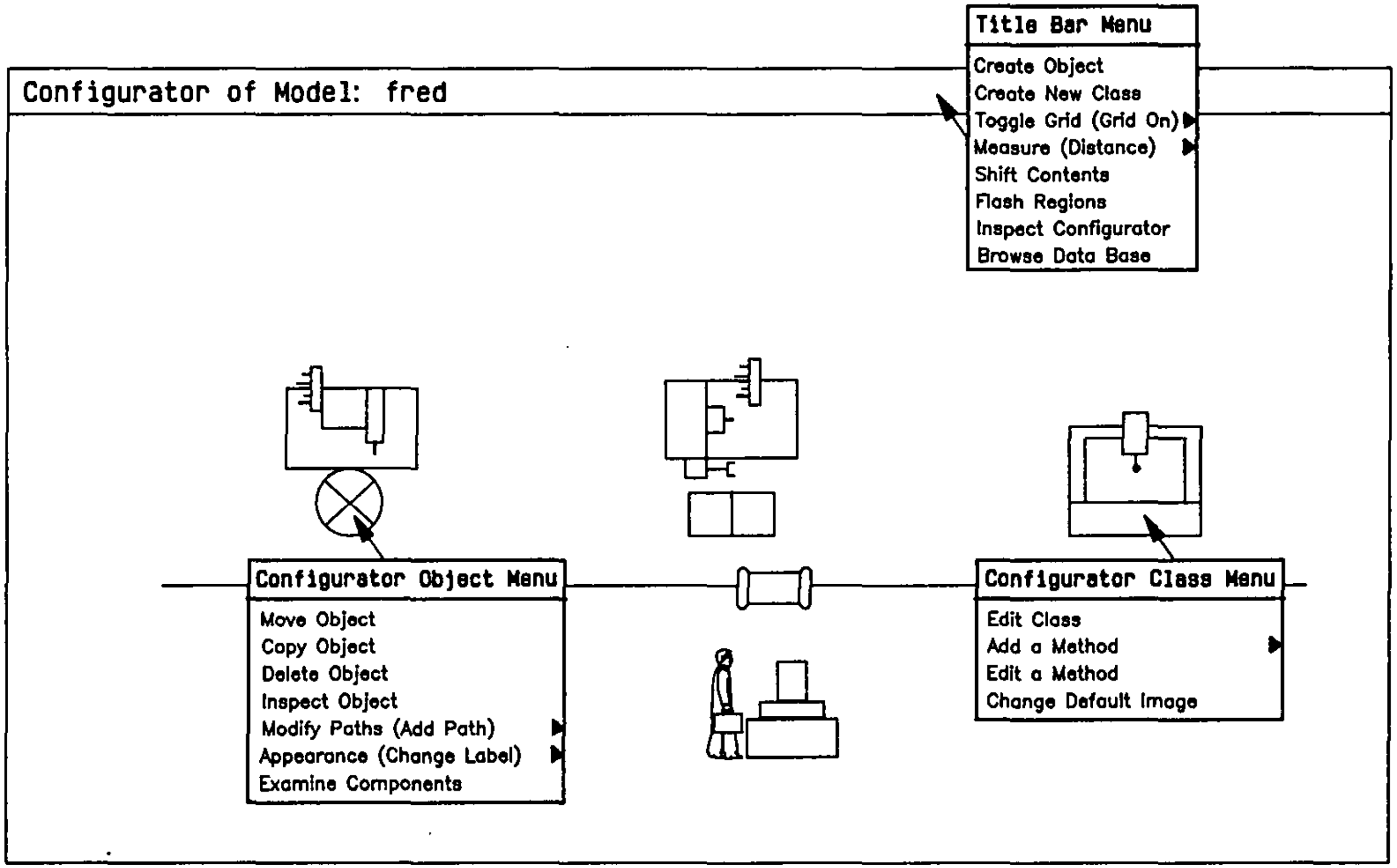


Figure 9.2	*Menus of the Model Configurator*	LUT - FMS **Research Group**

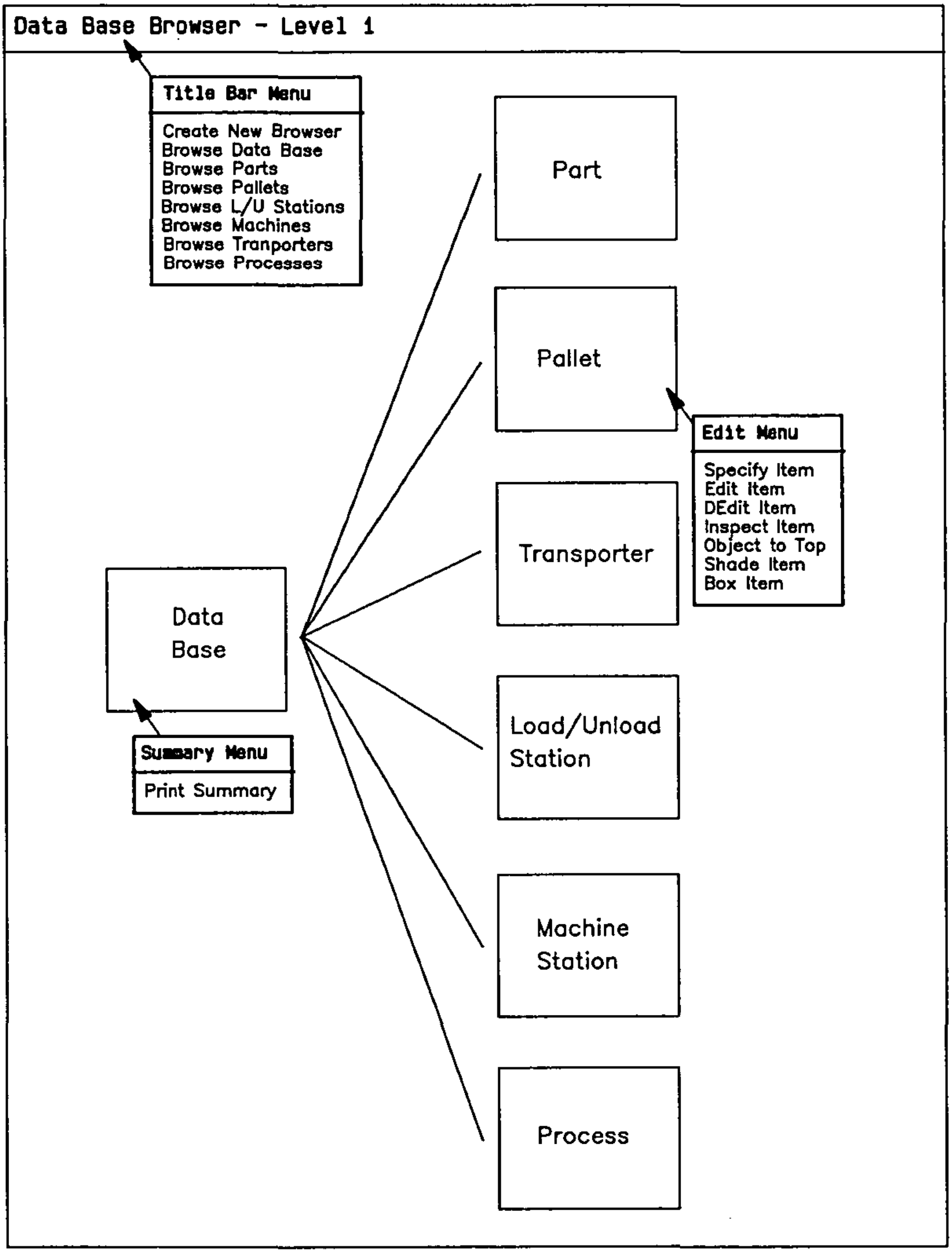


Figure 9.3

Menus of the Data Base Browser

LUT-FMS Research Group

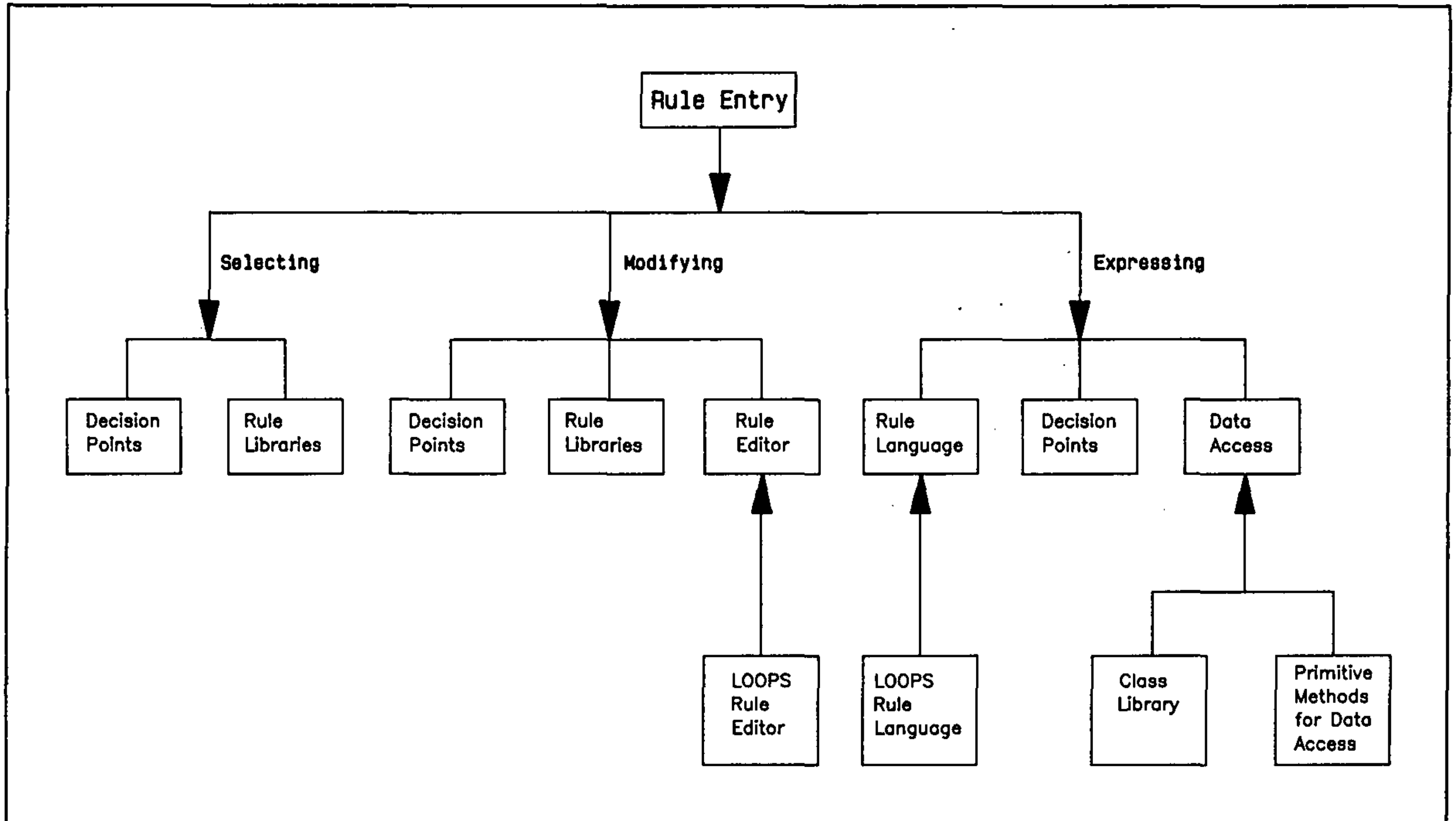


Figure 9.4

Mode of Rule Entry and Its Supporting Facilities

LUT - FMS

Research Group

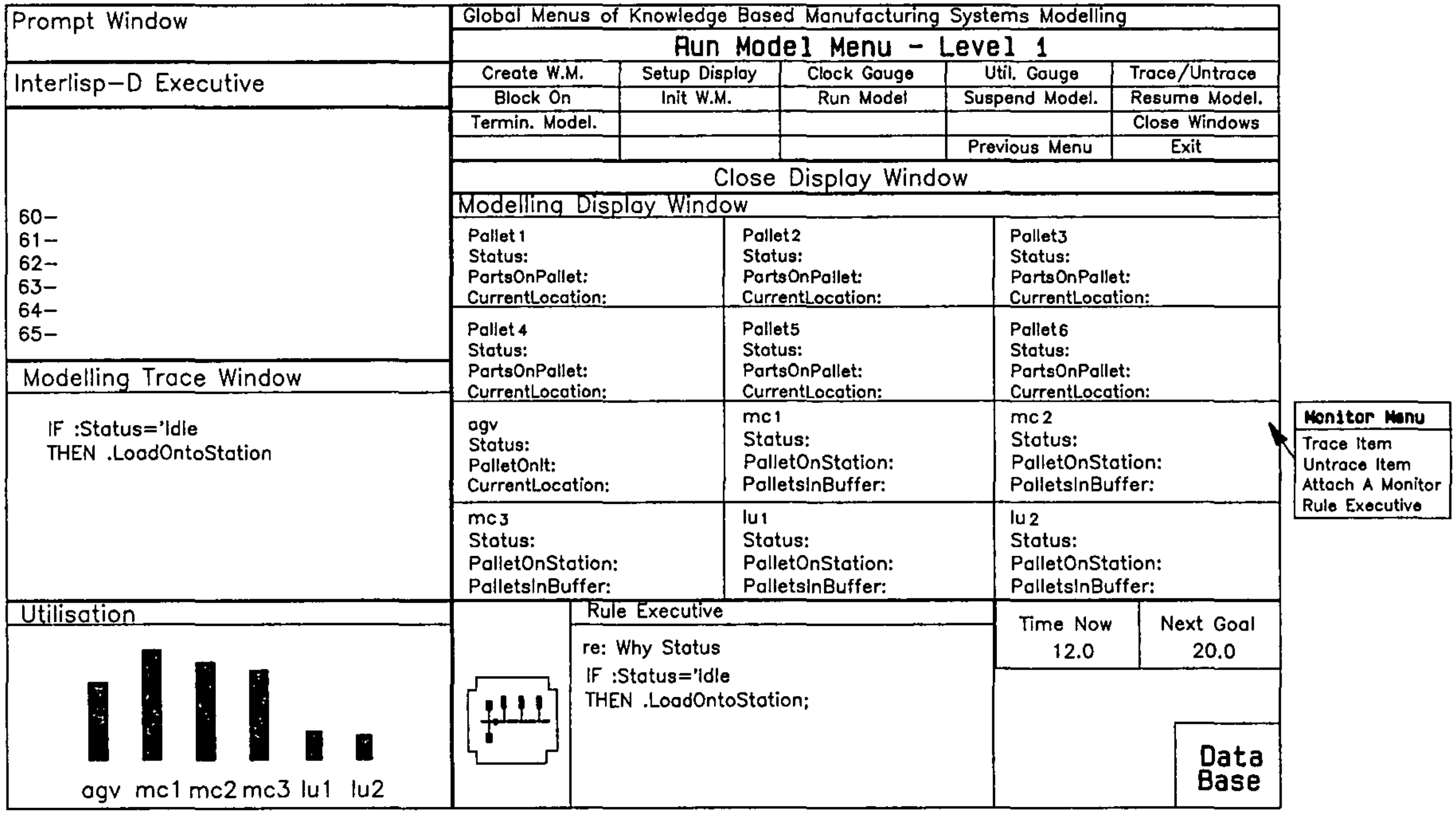


Figure 9.5

Graphics and Textual Output Facilities

LUT - FMS

Research Group

Chapter 10
SYSTEM OUTPUTS

10.1 Introduction

This chapter focuses on the output from the modelling system. First the concepts of manufacturing system performance figures are discussed. Then the output information is presented for each of the three modelling levels with interpretations of the output being given. Use of the outputs from different levels is also indicated.

10.2 Concepts of Manufacturing System Performance Figures

As discussed in Chapter 4, the entire objective of modelling is to help to design a manufacturing system. This is avhieved by providing insight into the behaviour of the system through simulation experiments. The behaviour of the system, however, can only be measured by certain carefully defined performance figures [250]. Thus the relationship between the decisions to be made in the design and the performance figures defined should be established. Suri [265a] has related the design decisions with some typical performance figures:

Design Decisions	*Performance Figures*
- number and type of machines	- utilization
- number of load/unload stations	- production rate
- which part types	- work in process
- alternative routes	- part flow times
- tool allocation	- queues at resource
- number and types of fixtures	- 'flexibility'
- number of pallets	- payback period
- number of transporters	- return on investment
- system layout	- net present value
- buffer size	
- operating policies	

However, it should be realised that this type of side-by-side relation is, to some extent, too rough to influence an actual design, because there is no such linear correlation between these two factors in practice. Although recent study of manufacturing systems using perturbation analysis techniques has shown that mathematical equations can be established to show the relationships [267], it is still unclear how these relationships can be developed if multiple variables are considered. In real systems, a particular performance figure is influenced by many design decisions [41] [66].

The basic requirement of defining these performance figures is that the difference between the actual performance of the designed system and the design performance as measured by these figures should be minimised. This is dependent on two factors. One is how closely the defined figure can be related to the design decisions. Another is how the defined performance figure is actually obtained in the simulation experiments.

The accuracy of the collected statistics depends on the simulation time period and the time period over which these statistics are collected. The operation of a simulation model can be divided into three phases [49]. The run-in period is the time span from the introduction of the first part into the empty system until the steady state is reached. During the steady state period, a fixed number of pallets should be in the system. The run-out period is determined from the time that a pallet taken out of the system is not introduced again into the system with a new part.

From the discussion on the manufacturing performance specifications in Chapter 4, the performance of a system can be measured at three hierarchically related levels:

- overall cell level
- work flow level and
- individual element level.

At the first level, the criterion is how a given production requirment can be met by the designed system [71] [257]. It represents the global capacity of the system subject to the specified production requirement. Depending on the simulation termination condition, there can be two different performance figures defined for measuring this aspect of the syetem. When a constant simulation run time is assumed which is normally less than the total time required for the complete machining of the production requirements, statistics should be collected for the total numer of parts completed by the end of the run [257] [165]. If the simulation run is not to be ended until the total production requirment is completed, then the total throughput time or make span should be recorded [142]. When tool flow is modelled, the tool requirement is also a significant measure for the overall cell performance.

At the second level, the important criterion is the rate at which components flow through the system [165]. It determines the components' delivery capability of the system. Two major performance figures can be defined to measure this aspect of the system: the throughput rate and lead time for each part type.

At the third level, the criterion is concerned with the activities of individual system elements over the simulated time period. Performance figures can be defined for each type of the system elements, such as part, load/unload station, machine station, temporary storage, work transporter, tool, tool transporter and tool store.

It is apparent that the performance output which can be provided by a modelling system is constrained by the assumptions made in the modelling. In the following sections, the output is described for each of the three modelling levels, with references being also made to the rules which facilitate the collection functions for the statistics.

10.3 Outputs of Level 1

The outputs from this level have been divided into three main areas, i.e., ***the overall cell performance, the primary outputs*** and ***the secondary outputs*** corresponding to the three levels to be measured (Figure 10.1).

The overall cell performance is concerned with the global capacity of the cell being modelled. Performance figures which have been defined to measure this capacity include the following:

- make span,
- total part throughput,
- total lateness,
- average part flow time and
- average utilisation of cell elements.

These figures are summaries of the collected statistics, which give an estimation together on the adequacy of the cell to meet production requirements and performance specifications [39]. The make span is defined as the time period after the first part is introduced into the system till the last part is completed. Measures which are discussed below have been defined to support the make span. The total part throughput is measured in terms of the parts produced per shift. Given the production requirment, this measure is determined by the make span, but is another form to show the global capacity of the system. The total lateness is the difference of the make span and the planning horizon. This measure shows how close the actual system capacity is to the expected capacity.

The average part flow time is computed by taking the average over the flow time of all parts for all types. This is based on the part lead times which are collected at the second level. This measure demonstrates on average how long a component should stay in the system. The average element utilisation is defined as the average busy time of a major cell element over the make span. It shows the overall use of the cell elements on producing the specified production requirment.

The primary outputs for cell assessment are concerned with the flow of parts in the cell. Two measures have been defined for assessing this aspect of the system. One is the throughput rate for each part type, and the other is the lead time for each part type [165]. The part throughput rate is the ratio of the batch size of a particular part type to the time period over which this part batch is completed. This is useful for measuring the flow of a particular batch of components through the cell.

The part lead time is defined as the time a component of a particular type spends in the cell. It is measured with regard to the average value, the maximum value, the minimum value, and the mean value for the part type. In addition, the lead time has been divided into the following according to the major activities the component takes when cycling through the cell:

- time machining,
- time transporting,
- time waiting, and
- time fixturing.

These figures are useful for demonstrating the aggregate flow patterns of the components for a particular part type.

The Secondary outputs are used to support the overall cell performance and the primary output information, which are the actual recorded statistics. These include the following categories:

- part performance,
- machine station performance,
- load/unload station performance, and
- transporter performance.

The part performance is converned with the flow patterns of an individual component in the cell. At this level, statistics can be collected with regard to the following major activites:

- time at buffer,
- time load/unload,
- time at machine station,
- time fixturing, and
- time waiting.

The machine station performance is concerned with the activities with which a machine station is involved during the simulation experiment. Therefore, the utilisation of machine stations have been defined to have the following categories:

- time machining,
- time load/unload,
- time idle, and
- spare capacity.

The spare capacity is defined as the time since the last use of a station till the end of the simulation run and the idle time is the total time that a machine is not used. Besides, the maximum queue length at each machine can be recorded which helps to determine the size of the buffer for the machine [165].

Since each load/unload station is assumed to have a infinite buffer at this level, the performance of a load/unload station can be measured with regard to the following:

- time fixturing,
- time load/unload,
- time idle,
- spare capacity, and
- maximum queue length.

The transporter performance is concerned with the utilisation of a transporter during the simulation experiment. It is measured in terms of the following:

- the load running time,
- the empty running time,
- the load/unload time and
- the idle time.

The load running time is collected when the transporter is transporting a pallet of parts to its next destination. The empty running time indicates the time that the transporter spends on travelling in order to pick up its next pallet, which shows the efficiency with regard to the transporter routing decisions.

All the time values for the utilization of the above cell elements can also be shown in terms of percentages over the simulated time period [164].

Refer to figures III.20 to III.23 and III.26 to III.29 of Appendix III for rules which are used to perform the actual collection functions.

10.4 Outputs of Level 2

The outputs which can be provided at this level have also been divided into three categories, i.e. the overall cell performance, the primary outputs and the secondary outputs. The definitions for each of the performance figures are similar to those of level 2. However, since finite local buffer spaces, tool requirements and temporary part storages are additionally considered at this level, additional outputs can be provided (Figure 10.2).

At first, in the overall cell performance category, the minimum cell tool requirement can be predicted with regard to the production requirements [42]. This is useful for determining a necessary package of tools for the cell in order to ensure a smooth work flow within the cell. It also provides a starting point for determining the appropriate tooling strategies for the management of the cell.

With respect to the machine station performance, tool requirements for each PTS can be predicted for the work scheduled to the station [50]. The statistics for the activities of a machine can be further divided into the following:

- cutting time,
- tool exchange time,
- load/unload,
- stationary time,
- idle time, and
- spare capacity.

The stationary time is defined as the time a pallet stays on the table of a machine before it is unloaded into the machine's local buffer. This figure is also useful for determining the buffer capacity of the machine.

The performance of a temporary storage can be estimated according to the following:

- stationary time,
- load/unload time,
- spare capacity, and
- idle time.

The stationary time of a temporary storage has been defined as the time that the storage is used, i.e. with pallets in the store. Further work is required on providing the WIP content of the store.

Refer to figures III.35 to III.42 and III.47 to III.51 for rules which are employed to perform the actual collection functions.

10.5 Outputs of Level 3

Again, the outputs at this level have been categorised as the overall cell performance, the primary outputs and the secondary outputs. The definitions are again similar to those described for the other two levels. However, this level models the exchange of tools between the tool magazine and the spindle, and the transfer of tools to and from the cell secondary tool storage. Therefore, in addition to the outputs from level 2 (exept the minimum cell tool requirement and the PTS tool requirement), outputs can also be provided with respect to tool flow and the influence of tooling on the performance of the cell (Figure 10.3).

As for the output on tool flow, this can include the perceived cell tool requirement under the specified tooling strategies, tool performance, tool transporter utilisation, PTS performance and STS performance [42].

The tool performance is concerned with the use of a particular tool during the simulated period. The figures which have been defined to measure this aspect include:

- number of uses,
- initial tool life,
- permissible life,
- used life,
- final location.

The number of uses is recorded each time when the tool is picked up from the tool magazine and inserted onto the spindle of the machine for performing the next operation or sub-operation. By keeping a record of the used tool life, the final tool life value can be provided. The final location of a particular tool can also be tracked by updating the location of the tool on transferring.

The tool transporter performance is concerned with the activities of the transporter during the simulation run. The following measures have been defined to assess this aspect:

- load running time,
- empty running time,
- load/unload time, and
- idle time.

The load running time is collected when the tool transporter transfers tools from the STS to the PTS or from the PTS back to the STS. The empty running time is the time when an empty tool transporter travels back to the STS from a PTS in order to load requested tools.

The PTS performance is concerned with the activities of the PTS subject to the tool provisions from the STS. The performance figures which have been defined are the follows:

- initial contents,
- final contents,
- changes of worn tools,
- changes of position tools, and
- load/unload time.

The final contents show the number of tools present in the magazine by the end of the simulation run. The changes of worn tools indicate the number of tool changes due to tool wear, and the changes of position tools show the number of tool changes due to the capacity of the magazine [63]. These figures are useful for determining the effects of magazine capacities.

The figures which have been defined to assess the STS performance include the following:

- final contents, and
- load/unload time.

In addition to the above outputs, the part performance has been extended to include the cummulative time that a component spends on waiting for tools, and the machine station performance to include the cummulative time that a machine spends on waiting for tools [84]. These figures are useful for determining the influence of tool flow on the work flow and the cell performance.

Refer to figures III.54 to III.66, III.69 to III.70 and III.73 to III.75 for rules which are used to do the collections.

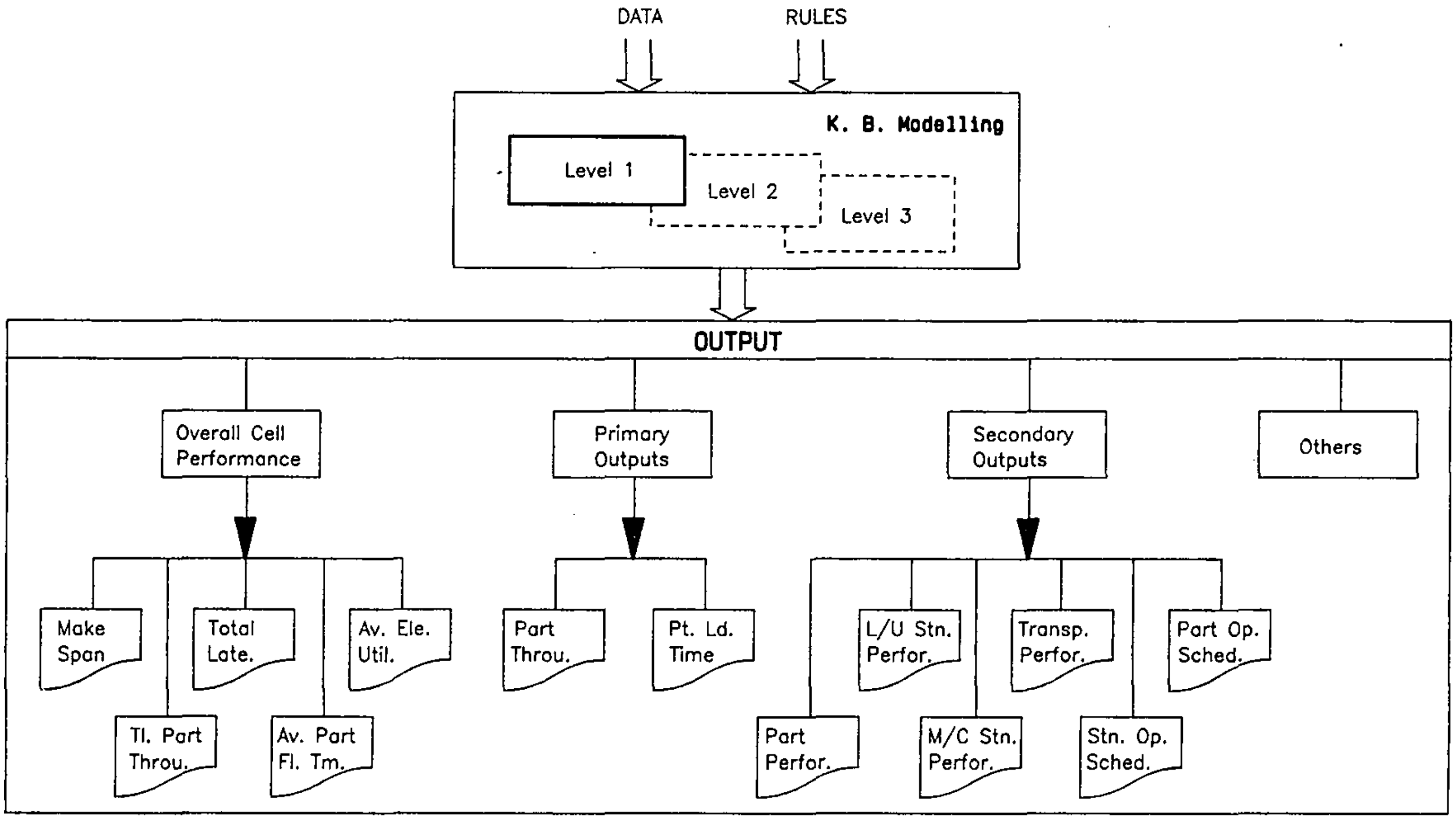


Figure 10.1

Outputs of Level-1 Modelling

LUT - FMS

Research Group

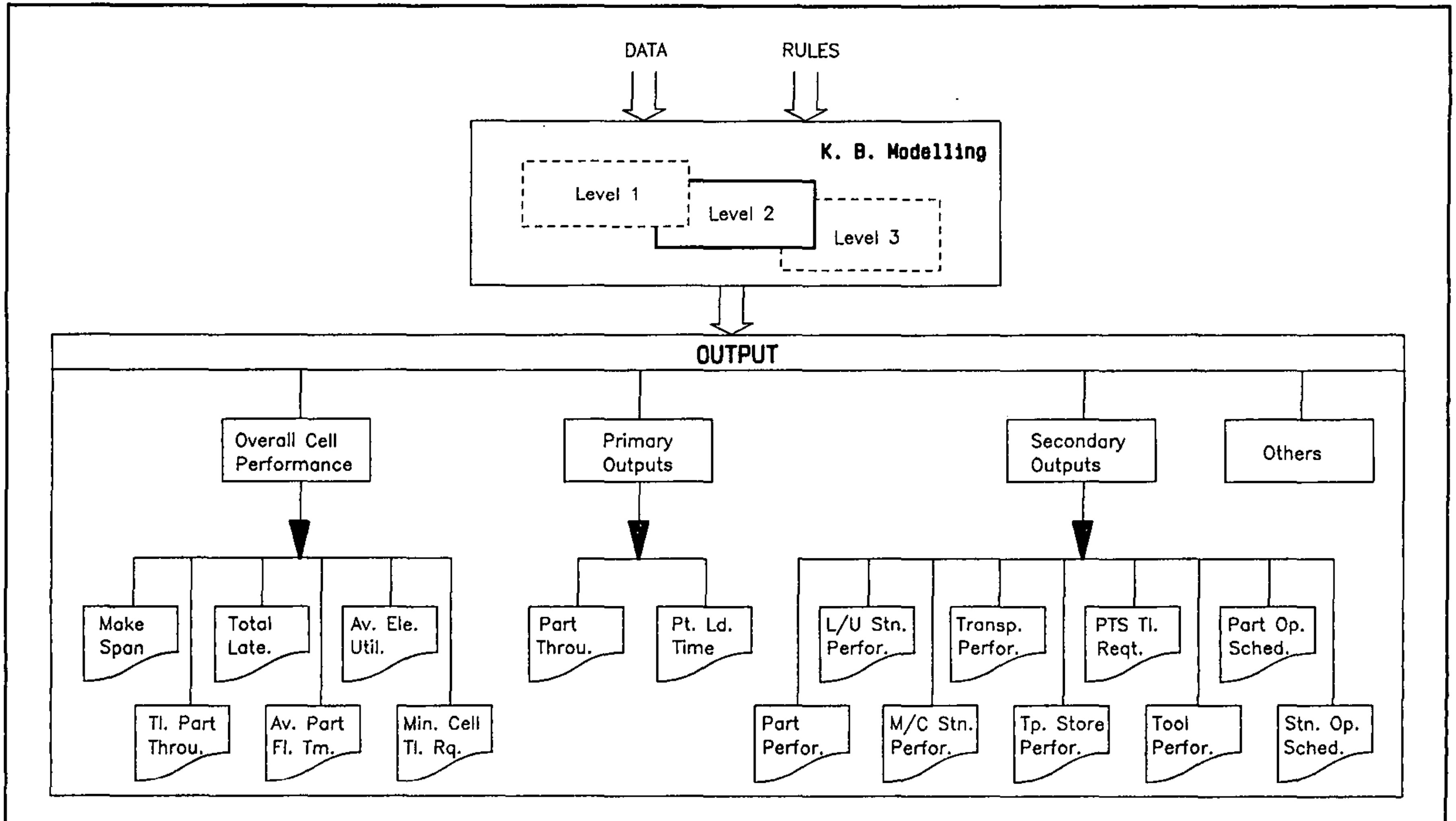


Figure 10.2	*Outputs of Level-2 Modelling*	LUT - FMS Research Group

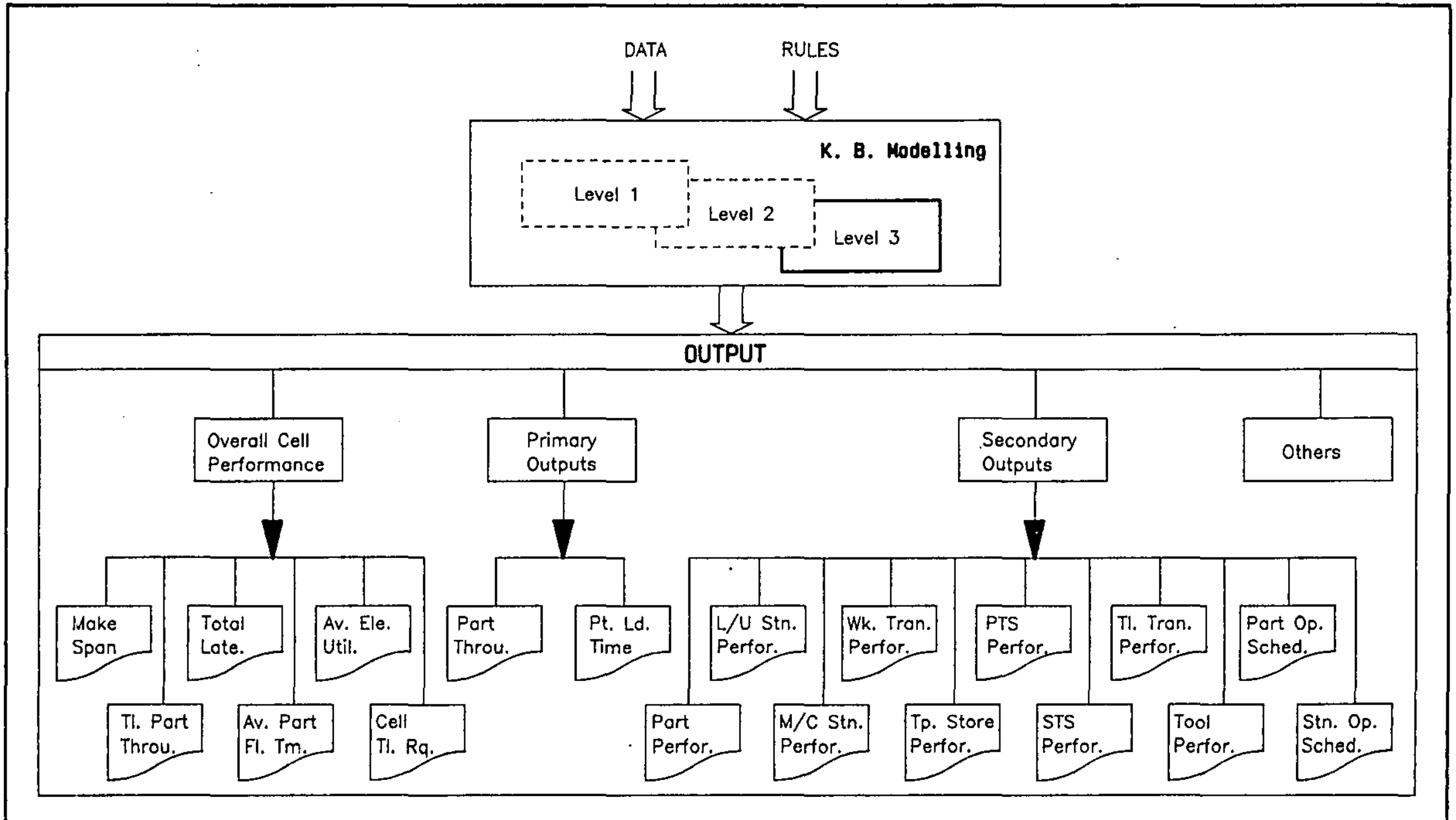


Figure 10.3	*Outputs of Level-3 Modelling*	LUT - FMS Research Group

Chapter 11
WORKSTATION CONSTRAINTS

11.1 Introduction

This chapter discusses the workstation constraints associated with the modelling system. A preliminary assessment is conducted first to show the major characteristics of the facility. Then the advantages of the environment for the development of this work are discussed. Finally, disadvantages of the equipment are also indicated.

11.2 System Characteristics

The Xerox 1186 Workstation is an artificial intelligence development workstation that combines Xerox hardware and software to provide a wide variety of user applications. It consists of a processor unit, a display screen, a keyboard, a three-button mouse, and a floppy disk drive as an optional feature [302]. Recently, there are software packages installed on the Workstation, such as the Common Lisp operating system and the STEM simulation package, which support the use of the equipment (Figure 11.1).

The programming system which runns on the Xerox hardware is called Interlisp-D [302]. It consists of a programming langauge which is symbolic manipulation based, a large number of predefined functions and a programming environment. The language and predefined functions of Interlisp-D are rich, but similar to those of other modern programming languages [300] [75]. The Interlips-D programming environment, on the other hand, is very distinctive. In addition to some basic programming tools, it also provides an integrated set of programming support mechanisms [302], such as the structure editor, the break package, the programmer's assistant, the masterscope, the record/data type package, the file package, the performance analysis, the multiple processes, the windows and the inspector.

LOOPS [53] [261] is an integrated knowledge engineering language developed at Xerox PARC and implemented in and as an extension on top of Interlisp-D [302]. The language is based on object-oriented representation scheme but also supports rule-based, procedure-based and access-based representation methods.

The principal characteristic of LOOPS is the integration of its four programming schemes to allow the paradigms to be used together in knowledge system building.

Procedure-oriented programming: In this paradigm, large procedures are built from small ones by the use of subroutines. Data and programs are kept separate. Most computer languages are like this. The procedure-oriented part of LOOPS is Interlisp-D which provides the solid foundation on which the rest of LOOPS is built.

Object-oriented programming: In this paradigm, information is organized in terms of objects, which combine both instructions and data. Large objects are built up from smaller ones. Objects communicate with each other by sending messages. The conventions for communicating with an object by using messages constitute message protocols. Standardized protocols enable different classes of objects to respond to the same kinds of messages. Inheritance in a class lattice enables the specialization of objects. For the discussion on the use of this paradigm, see Chapters 6 and 7.

Access-oriented programming: This paradigm is useful for programs that monitor other programs. Its basic mechanism is a structure called an active value, which has procedures that are invoked when variables are accessed. A useful way to think of active values is as probes that can be placed on the object variables of a LOOPS program. These probes can trigger additional computations when data are changed or read, Chapter 7. For example, they can drive gauges that display the values of variables graphically (Chapter 9).

Rule-oriented programming: This paradigm is specialized for representing the decision-making knowledge in a program. In LOOPS, rules are organized into rulesets which specify the rules, a control structure, and other descriptions of the rules (refer to Chapter 8). Two key features of the rule language are that it provides tecjniques for factoring control information from the rules, and also dependency-trial facilities, which provide mechanisms for 'explanation' and belief revision.

Obviously, this integration provides the user with a great deal of flexibility. For example, rules and rule sets are considered LOOPS objects and can communicate by object- oriented message passing or by standard subroutine calls, methods can be either LISP procedures or rule sets and can be used with active values to display gauges [53].

Appendix IV lists the major specifications of the Xerox 1186 Workstation and the LOOPS software.

11.3 Advantages of the Facility

As an advanced AI workstation, the Xerox 1186 provides the development of knowledge based systems with the following advantages:

(1) In contrast to the numeric manipulation methods as used by conventional programming systems, such as FORTRAN and PASCAL, One of the basic requirement for building knowledge systems is that the programming language should be symbol-manipulation based so as to allow for more comprehensible representation of the human knowledge. This is well satisfied by the Interlisp-D programming environment.

(2) An important principle of knowledge programming is that different paradigms are appropriate for different purposes. This is in contrast with the use of a single programming paradigm for everything, be it logic programming as in Prolog [75], procedure- oriented programming as in Lisp [300], object-oriented programming as in Smalltalk [117], or rule-oriented programming as in OPS5 [245]. Purely rule-based representation scheme is inadequate in defining terms, describing objects, and identifying relationships [245]. Object-oriented representation scheme, on the other hand, has the shortcomming of being unsuitable to describe decision- making knwoledge, events or interactions between objects [178].

There are also various metrics of cost for applying a programming paradigm across a spectrum of applications. Examples of metrics are the cost of learning, the cost of modifying, the cost of debugging and the cost of running. These costs vary across paradigms and applications because different programming paradigms provide different ways of organizing information in programs. For a given metric and application, some programming paradigms can be more cost-effective than others. By allowing for choice and combination of paradigms, a knowledge programming system enables various costs to be lowered.

In LOOPS, these representation methods are integrated. The object-oriented method provides a rich structural language for describing the objects referred to in rules. It siginificantly help with rule management by providing a means of modularising, organising, indexing, scheduling, and invoking rules according to their intended use. On the other hand, rules can be used to augment the effectiveness of object based representations [245] (Chapter 7).

(3) It provides a very powerful programming environment for creating and debugging knowledge systems. Many of the facilities of Interlisp-D are extended to other paradigms [261], such as the display-oriented break package, editors and inspectors. In LOOPS, this extension has led to the same synergy that is exploited in using multiple paradigms for application programs. For example, the notion of 'breaking' on access to a function is extended to breaking on access to a variable by using active values to invoke the break package, and the notion of tracing is extended to the notion of having gauges that can monitor the values of variables.

(4) It has high resolution graphics facilities. Multiple scrollable windows allow many different processes or activities to be active on the screen at the same time [302]. Bit maps provide the basics for building icons that represent real world objects [105] [32]. Menus allow highly interactive software to be developed which enables friendly communication between the user and the computer. On the whole, a user-friendly interface can be developed by using all these facilities (refer to Chapter 9).

(5) Since many objects have been defined in LOOPS and these objects are accessable by any users, specialization of these objects can help in developing other knowledge systems. For example, the *LatticeBrowser* can be used to develop a specialized browser for managing instance objects (see Chapter 7). *Window* can be specialized to design a map for manufacturing system layout definitions (Chapter 9).

(6) Compared with other available environments, such as KEE [161], LOOPS is fairly easy to learn and bring into effective use. This can be shown through the experience gained during the development of the knowledge based modelling system.

11.4 Disadvantages of the Facility

Based on the experience gained in the development of this work, the following shortages can be recognised for this facility:

(1) Since LOOPS has only reached the stage of development and emerge as a research system [289], it is still relatively slow and inefficient. Evidence can be found in the case studies to be presented in later chapters.

(2) As a result of trying out many new ideas in LOOPS, it includes exotic features known only to a few. This makes the learning and training fairly costly.

(3) The Xerox Workstation has not become widely available in industry, and the LOOPS package is at the present time dedicated to the Workstation though the running of the package on a SUN Workstation may be possible in the near future. Therefore, the application of the work reported in this thesis can be constrained by the computing environment.

11.5 Conclusions

The Xerox Workstation and the LOOPS knowledge engineering language, as used in the development of this work, have proven to be effective facilities for the modelling of complex manufacturing systems. The Workstation provides a very powerful programming environment for creating and debugging knowledge systems. The LOOPS package, on the other hand, has integrated multiple programming paradigms and enables representation of different types of manufacturing knowledge using appropriate programming schemes.

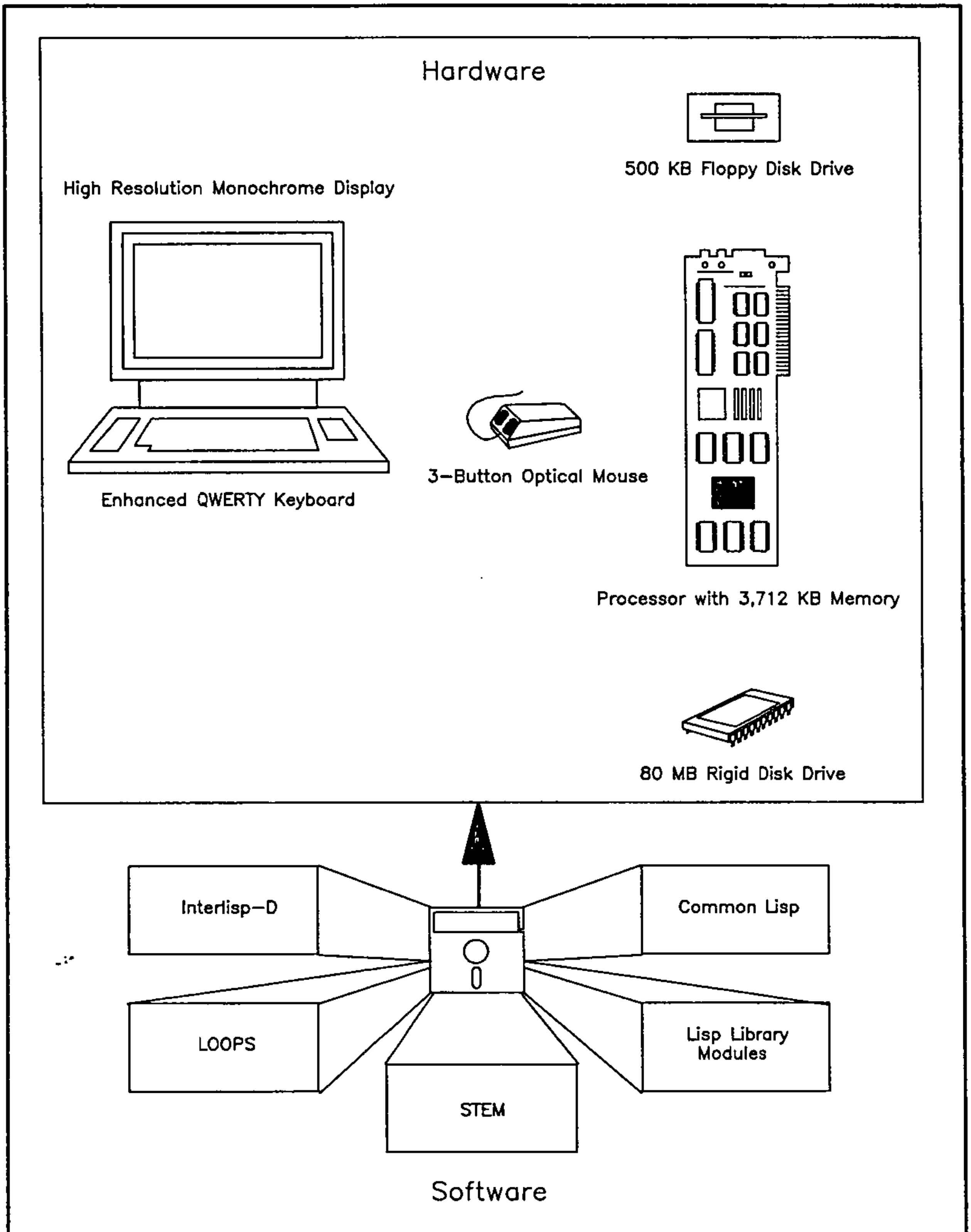


Figure 11.1 *Xerox 1186 WorkStation Configuration*

LUT-FMS Research Group

Chapter 12
THE THREE MACHINE CELL

12.1 Introduction

A flexible manufacturing cell for an initial industrial case study is described in this chapter. The data is supplied by a British company. Major aspects to be covered include the components to be manufactured, the machining process for the parts, the system design and the operation of the cell.

12.2 Initial Comments

This cell is introduced in order to bring the established modelling system under a preliminary test with realistic industrial data and operational strategies. The emphasis of the test is to demonstrate the difference among the three levels of modelling by concentrating on the study of the physical structure of the cell.

This chapter focuses on the description of the cell and the overview of its operation. The operational constraints are more closely described in the next chapter.

12.3 The Components

The parts concerned are of five families, all of which are specific to the final product that is assembled using these components. A fixed number of parts from each family are required for every product. Therefore cell output can be reliably predicted over a specified period. Seven parts are specified as candidates for the FMC in view of their complexity and size. One of the 7 parts has been split up into two because it entails two visits to a machine. Details of daily requirements of parts are summerized in Figure 12.3. Appendix VI.1 summarizes the process details for the seven components.

All parts are cast iron castings. The operations involve the full range of work carried out on machining centres, such as milling, drilling, boring, reaming and tapping.

12.4 The Machining Process

It is recognised that minimising the movement between stations and pooling machines into groups can maximise the performance of the system, such as station utilisation and production rate. Therefore, each part, except part 3, is to complete all its processes on just one machine tool. This can be achieved by holding inventory of partially machined components on each pallet which has more than one position in which machining is carried out. Part 3, however, requires refixturing at the load/unload station and carries out its second operation on the same machine.

All components are produced on cube-type fixtures so that if necessary more than one machining operation may be carried out on the same fixture, or where batch sizes are greater than one, components may be located in a similar orientation on different faces of the cube for the same operation. In this study, one component per fixture is assumed, through the modelling system can handle the situation where any number of components may be specified for a fixture.

All supporting activites, such as fixturing, defixturing, inspection and cleaning of finished parts, are performed by the cell operators. This makes the cell effectively an autonomous machining cell, which takes in raw castings and supplies finish machined components direct to the point of use.

12.5 The System Design

The cell, shown in Figure 12.1, comprises two load/unload stations and three Makino MC 1210 horizontal machining centres. Work fixturing and defixturing is performed manually at either of the two load/unload stations. The machines are tooled similarly and each part is assigned to two machines based on the considerations that all parts may still be processed if one machine breaks down, and that if a part can be processed by more than one machine, it should spend less time queueing.

All three machining centres have a tool changer and a tool magazine of capacity 120 tools. Tool change between the magazine and the spindle is performed by a double-ended arm, which selects the required tool from the magazine whilst machining is in progress, and then changes the tool when the spindle has stopped.

There are 18 pallets stands (which can be expanded to 22) used as temporary storages to park both loaded and empty fixtures until either of the load/unload stations becomes available, or the assigned machine, or rather its input buffer. Each part has an individual fixture which is assigned to a pallet stand. Thus each pallet stand forms a unique storage location for the part assigned.

Each machine is supported by two pallet buffers, one for input and another for output. This frees the machine from waiting for the work transporter before part load/unload can occur.

Parts are transferred between the load/unload stations, pallet stands and machines by one automatic guided vehicle (AGV) following a vehicle track. There are altogether 6 pallets in the system, one for each of the four part families and two for the fifth family. Figure 12.3 shows the pallet type required for each part type.

Each machining centre is equipped with a Fanuc System 11 controller which has sufficient memory to accommodate the programs for all the parts to be produced at that machine. Tool life monitoring is also performed by the controller. A central computer is used to co-ordinate the cell's operation, which keeps track of all the parts in the system and schedules the machining operations. Tool flow management in the cell is totally manual, i.e. no computer is used to control the flow of tools between tool stores.

12.6 Operation of the Cell

12.6.1 Initial Comments

An overview of the operation of the three machine cell is given below. Both work and tool flow management are considered. The description of the cell operational strategies both in the general statement form and in the exact rule language form is given in the next two chapters.

12.6.2 Work Flow

The work flow management of the cell is characterised by the the fact that only one transporter is employed, each of the three machines has a dual pallet exchange buffer and twenty-two pallet stands are used as temporary work storages (Figure 12.2).

This physical structure dictates that after a new part is released into the cell, it must be palletised at one of the load/unload stations. Since the machining operation of each part has been assigned to two machines, the control computer can then check if the input and output buffers of any of the two machines are clear. If so and the AGV is free, the loaded pallet is transported to the selected machine. The input buffer of a machine can be loaded even if the machine is in process, so long as the output buffer is empty.

If none of the machines is available, the computer then requests the tranporter to move the pallet to the assigned pallet stand, and the pallet will be waiting at the stand until the required machine and the AGV become available.

Once a loaded pallet has been transfered to the input buffer of the machine, it will be queueing there until the part, which is currently in process, has completed its operation and has been unloaded into the output buffer of the machine.

As soon as a part is loaded onto the table of the machine, the machining process starts. When the machining of a part is finished, the computer then has to check the output buffer of the machine. If it is clear, the completed part can then be unloaded, otherwise it will have to wait at the machine table.

When a part has completed its machining operation, the computer then checks whether there is a free load/unload station. If there is one, the part is moved to the load/unload station and the depalletisation process can starts. If not, the part is moved to the assigned pallet stand in order to free the output buffer of the machine.

For any pallets waiting at the pallet stands, when the required stations become available, they are transported to these stations.

Notably, since there is only one AGV, any transporting activities can not start until the AGV becomes free, even if the destinations of the pallets are available.

12.6.3 Tool Flow

Since each machining centre is equipped with a very large capacity magazine which can accommodate a wide spectrum of tool types and sister tooling for critical tools, no tool flow was originally planned. The tool provision strategy is such that all tools required at each machine should be permanently assigned to the relevant magazines and attended to (changed, reground or reset) when their individual tool lives expired.

Each tool has a fixed position in a tool magazine, and will always be replaced in the same tool pocket after removed for use at the spindle or tool attention. Therefore the cell is susceptible to errors is a tool is placed in a wrong pocket by an operator, as the machine makes no physical identification of a tool other than from its position in the magazine.

Tool change is done manually and can be done while machining is in process. Before tools are replaced in a magazine, their lengths and diameters offsets are entered directly into the controller memory from the tool presetting station.

Tool usage time is recorded for each tool within the relevant controller's memory. When a preset tool life limit is exceeded for a particular tool, an alarm is triggered to notify the operator and the machine will not load that tool again until its life usage is reset. The tool life assigned to each tool is not a absolute machining time limit (although this would be so if sufficiently accurate data were known) but a maximum time between checks on the tool's condition.

Often when a tool life alarm is set off, the operator will inspect the tool and find it to be capable of further use, and therefore will reset the tool life without changing the tool. As confidence is built up as to the performance of specific tools over time, the tool lives should be extended until they represent a true tool life limit, after which a tool should be reground or reset.

As mentioned above, this cell does not have an automated tool flow management system. Thus the following cell level tool flow network is proposed for the management of tool flow in the cell (Figure 12.4). It assumes that an STS is used as the store of tools transported to the PTSs and the destination of tools returned from the PTSs. However most tools use indexible inserts which are changed by the cell operator when required without returning to the STS.

12.7 Comments on Modelling

Modelling of the above described cell will be done in three chapters. In Chapter 13, the behavioural rules that define the basic operation of the cell are given. Chpater 14 concentrates on the decision rule and data input requirement of the cell model for each of the three levels. The results of the modelling are then presented in Chapter 15, which also discusses the difference among the three levels based on the results obtained.

- 3 Makino MC1210 Machining Centres
  with 120 Primary Tool Store
  and Dual Pallet Exchange Buffer
- 1 Rail Guided Vehicle
- 6 Pallets
- 8 Part Types
- 2 Load / Unload Stations
- 22 Temporary Storage Stations
- 1 Secondary Tool Store
- 3 Men

Figure 12.1	*The Three Machine Cell Elements*	**LUT-FMS** Research Group

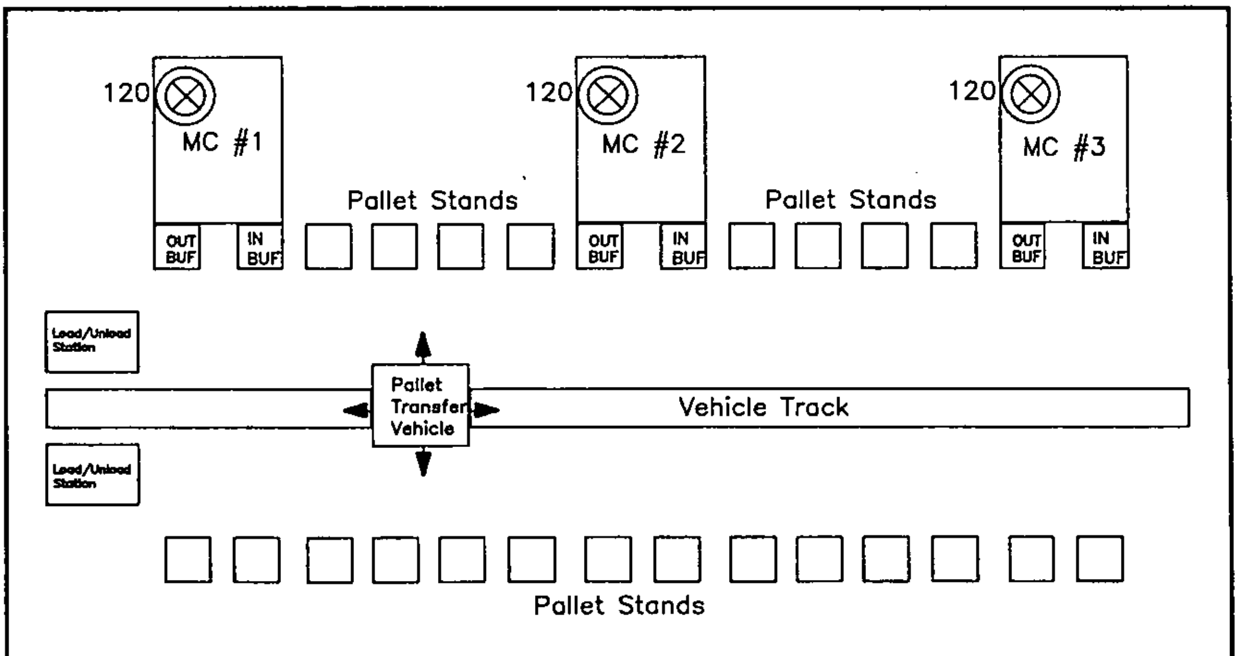


Figure 12.2	*The Three Machine Cell - Part Flow*	LUT - FMS **Research Group**

Part Type	Quantity/Day
1	4
2	4
3	8
4	8
5	5
6	3
7	4
8	4

Pallet Type	Part Types
1	1 & 2
2	3
3	4
4	5 & 6
5	7
6	8

Figure 12.3 *The Three Machine Cell Part/Pallet Information* **LUT-FMS** Research Group

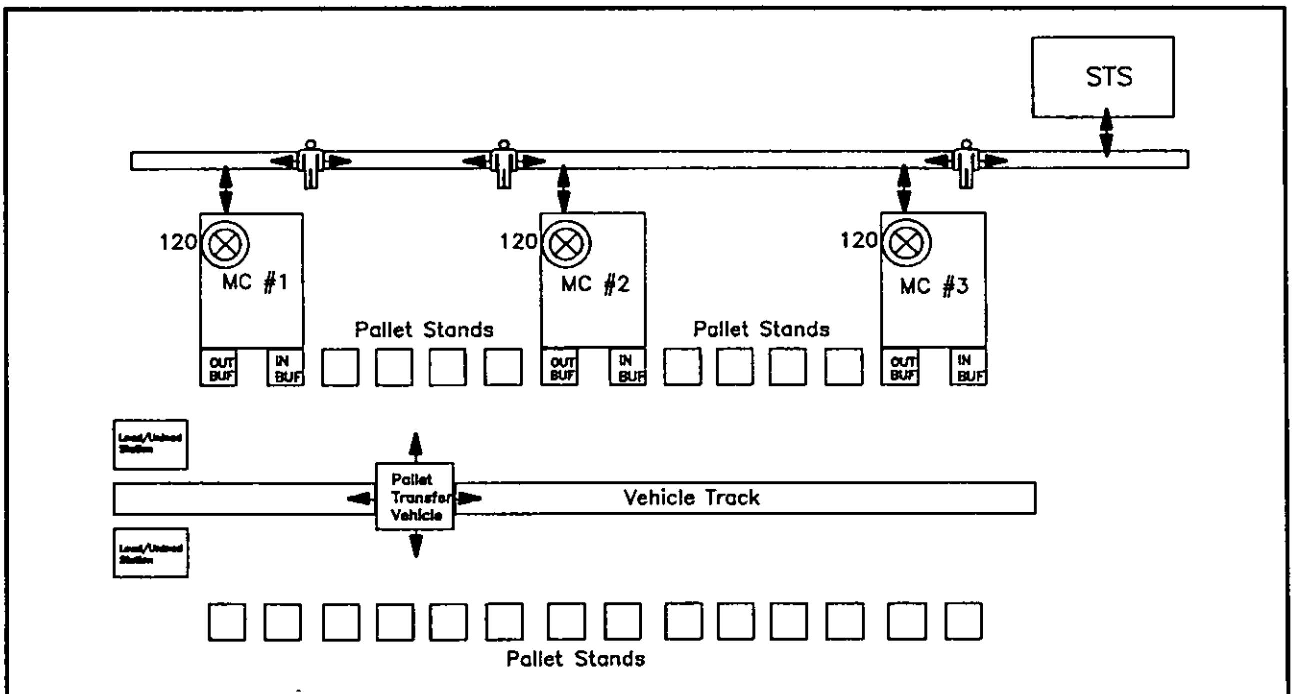
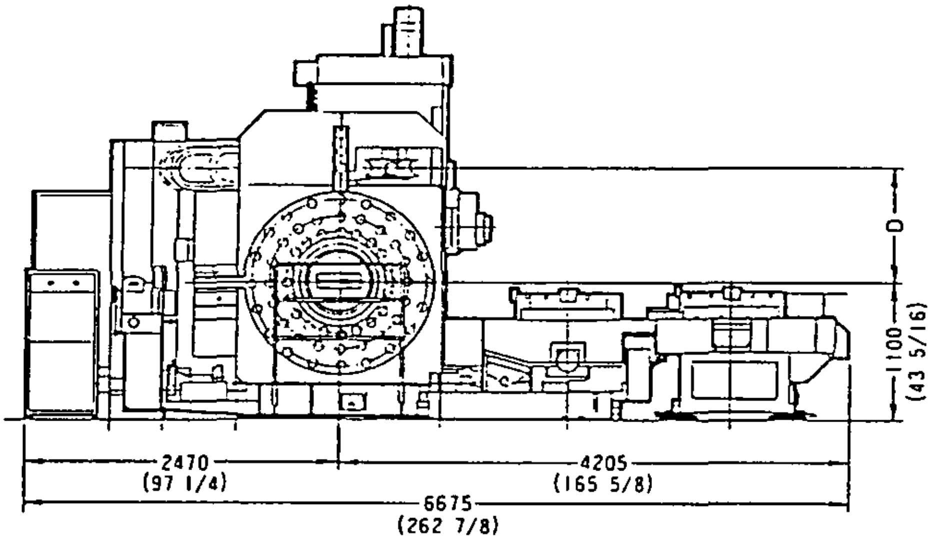


Figure 12.4 *The Three Machine Cell - Part & Tool Flow* LUT - FMS **Research Group**



machine number:	1, 2, 3
machine desc.:	MC1210
machine id.:	MC1, MC2, MC3
prim. stores:	1
PTS capacity:	120
PTS index time:	0.5
tool exchange time:	0.26
tool complement fixted/variable:	V
PTS fixted/movable:	F
number of spindles:	1
number of type in cell:	3
machine load time:	1
machine unload time:	1
buffer type:	dual pallet exchange
buffer load time:	0.27
buffer unload time:	0.27

Figure 12.5	*Description of the MC1210 in the Three Machine Cell*	LUT-FMS Research Group

Chapter 13
BEHAVIOURAL RULES OF THE THREE MACHINE CELL MODEL

13.1 Introduction

In this chapter, a description of the behavioural rules of the three machine cell model both in general statement form and in rule language form is given. These rules, obtained from the rule library, define the case study in the modelling system consistent with the description of the cell's operation given in chapter 12.

13.2 Initial Comments

The behavioural rules for both work and tool flow are given below, which dictate the basic operation of the cell. These rules are decribed in statement form in the text, whereas the exact rules that will be used in the modelling system for the cell study are given in figures. For those of the rules, which have been detailed in the Appendixes and are used in combination to represent a rule statement, they are listed in tables.

The role of these behavioural rules has been discussed in Chapter 8. With regard to the user interactions involved in the entry of these rules, refer to Chapter 9.

13.3 Work Flow Rules

13.3.1 Scope of Work Flow Rules

As discussed in the previous chapters, each level models a cell at a different level of abstraction or detail which involves the use of varying behavioural rules in the modelling system to represent the operation of the cell. Thus these rules are given below in separate sections against each of the three levels.

The scope of the work flow behavioural rules is to model the basic work flow activities (or actions in the modelling terminology) in the cell. These include work fixturing and defixturing at the load/unload stations, work transporting between stations, work queueing in buffers and work processing at machines during the whole cycle of work flow within the cell.

13.3.2 Rules of Level 1

The work flow management of the cell is characterized by the following rules at this level:

Rule 1: The cell has a fixed number of pallets, i.e. as soon as a pallet takes a part out of the cell, a new part is loaded on the pallet unless all the parts for the pallet have been completed (Table 13.1).

Rule 2: If there is an empty pallet/fixture at the load/unload station, then a new part is palletised and unloaded into the local buffer of the station on completion of the palletisation (Table 13.2).

Rule 3: After a new part has been loaded onto the pallet, the AGV will move the pallet to the infinite local buffer of the assigned machine when the AGV is available (Table 13.3).

Rule 4: When a machine becomes free, one of the pallets waiting in the queue is loaded and machined. On completion of the machining, the pallet is unloaded immediately into the buffer of the machine (Table 13.4).

Rule 5: After the part on a pallet is machined, the AGV will move the pallet to the infinite local buffer of one of the load/unload stations when the AGV is available (Table 13.3).

Rule 6: When a load/unload station becomes available, the machined part (on pallet) is loaded to the station and is depalletised. This completes the cycle of the part in the cell (Table 13.5).

Notably, since level 1 does not model the capacity of station local buffers, and the temporary storages, no rules are given above which describe the operation of these aspects. This is a violation of the cell operation described in Chapter 12.

13.3.3 Rules of Level 2

The following rules are given which are used to manage the basic work flow functions in the three machine cell at level 2:

Rule 1: The cell has a fixed number of pallets, i.e. as soon as a pallet takes a part out of the cell, a new part is loaded on the pallet unless all the parts for the pallet have been completed (Table 13.6).

Rule 2: If there is an empty pallet/fixture at the load/unlod station, then a new part is palletised (Table 13.7).

Rule 3: After the new part has been loaded onto the pallet, the control computer then checks to see if the input buffer of an assigned machine is available. If it is available, the AGV will move the pallet to the machine (Table 13.8).

Rule 4: A loaded pallet/fixture can not be placed in the input buffer of the machine until the output buffer is cleared. An input buffer may be loaded while the machine is in cycle, provided the output buffer is empty (Figure 13.1).

Rule 5: If the input buffer is not available, the computer checks the assigned pallet stand. If the stand is available, the pallet is moved to the pallet stand by the AGV (Table 13.9).

Rule 6: If no pallet stand is available for the first pallet in the queue, the computer will repeat the process for the part waiting in the next full pallet/fixture (Figure 13.2).

Rule 7: When a machine becomes available, the pallet in the input buffer of the machine is loaded and the machining of the part on the pallet starts immediately (Table 13.10)

Rule 8: Following machining of the part on a pallet, the pallet/fixture is moved to the output buffer if it is free (Table 13.11).

Rule 9: If either of the load/unload stations and the AGV are available, the pallet is moved back to the load/unload station on completion of the machining of the part on the pallet/fixture (Table 13.12).

Rule 10: If no load/unload station is available but the assigned pallet stand is free, the machined part with the pallet is moved to the pallet stand (Table 13.13).

Rule 11: If the required load/unload station or the assigned machine and the AGV become available, the pallet waiting at its assigned stand is moved to its destination (Table 13.14).

Rule 12: After a pallet is moved back to the load/unload station, it is depalletised there immediately and the pallet becomes free (Table13.15).

Notably these rules are totally consistent with the operation of the cell described in Chapter 12.

13.3.4 Rules of Level 3

The work flow pattern at this level is similar to that of level 2, but special considerations have to be given to the influence of machine's tool availability on the flow of work in the cell. Thus the work flow behavioural rule of the three machine cell at this level are given as the follows:

Rule 1: The cell has a fixed number of pallets, i.e. as soon as a pallet takes a part out of the cell, a new part is loaded on the pallet unless all the parts for the pallet have been completed (Table 13.16).

Rule 2: If there is an empty pallet/fixture at the load/unlod station, then a new part is palletised (Table 13.17).

Rule 3: After the new part has been loaded onto the pallet, the control computer then checks to see if the input buffer of an assigned machine is available. If it is available, the AGV will move the pallet to the machine (Table 13.18).

Rule 4: A loaded pallet/fixture can not be placed in the input buffer of the machine until the output buffer is cleared. An input buffer may be loaded while the machine is in cycle, provided the output buffer is empty (Figure 13.3).

Rule 5: If the input buffer is not available, the computer checks the assigned pallet stand. If the stand is available, the pallet is moved to the pallet stand by the AGV (Table 13.19).

Rule 6: If no pallet stand is available for the first pallet in the queue, the computer will repeat the process for the part waiting in the next full pallet/fixture (Figure 13.4).

Rule 7: When a machine becomes available, a check is made of tools present in the PTS of the machine against the tools required by the part on a pallet waiting to be loaded in the input buffer of the machine. If some of the required tools are not available, then a tool requirement is generated for the part (Figure 13.5).

Rule 8: If all the required tools are available, the pallet is loaded and the machining process starts (Table 13.20).

Rule 9: Following machining of the part on a pallet, the pallet/fixture is moved to the output buffer if it is free (Table 13.21).

Rule 10: If either of the load/unload stations and the AGV are available, the pallet is moved back to the load/unload station on completion of the machining of the part on the pallet/fixture (Table 13.22).

Rule 11: If no load/unload station is available but the assigned pallet stand is free, the machined part with the pallet is moved to the pallet stand (Table 13.23).

Rule 12: If the required load/unload station or the assigned machine and the AGV become available, the pallet waiting at its assigned stand is moved to its destination (Table 13.24).

Rule 13: After a pallet is moved back to the load/unload station, it is depalletised there immediately and the pallet becomes free (Table13.25).

13.4 Tool Flow Rules

13.4.1 Scope of Tool Flow Rules

Since no rules are provided by the company with regard to the flow of tools, the rules described below are proposed to govern the behaviour of tool flows in the cell. Although tool flow modelling is facilitated only at level 3, level 2 considers tool requirements planning which generates the minimum and maximum cell tool requirements for a scheduled work list. Therefore behavioural rules for both of these levels are given.

The scope of these behavioural rules is to model the basic tool flow activities between the STS and PTSs and the tool change startegies at the machine PTS.

13.4.2 Rules of Level 2

The maximum and minimum tool requiremnents planning is performed by the following rules:

Rule 13: When the machining process starts at a machine, update the machine's tool list, i.e. for each of the required tools, update the tool life used if an existing tool is usable, otherwise create a new tool for the machine and updates its life used (Table 13.26).

Rule 14: On start of the machining process on a part, update the cell's tool list. That is, for each of the required tools, update the tool life used if an existing tool in the list has enough life left, otherwise create a new tool for the cell and updates its life used (Table 13.27).

13.4.3 Rules of Level 3

The management of tool flow in the three machine cell can be considered as characterized by the following rules:

Rule 14: If a part is waiting for tools and there is a free tool transporter, the transporter is sent back to the STS and loaded with tools required by the part (Table 13.28).

Rule 15: A loaded tool transporter then moves the tools to the machine where the part is waiting for tools (Table 13.29).

Rule 16: If there are any spare positions in the PTS, then the tools are loaded from the transporter to fill these positions (Figure 13.6).

Rule 17: If there are not enough spare positions in the magazine, worn tools or, if necessary, unrequired tools are taken out from the PTS and exchanged with the tools on the transporter (Figure 13.7).

Rule 18: The transporter then transfers the tools, if any, back to the STS (Table 13.30).

Table 13.1: The Three Machine Cell Model at Level 1 - Behaviural Rule 1

Figure No.(s) in Append. III	RuleSet Name	Rule Number
15	Pallet.TestStartOfAction1	1
16	Pallet.CheckLoadUnloadStation	1

Table 13.2: The Three Machine Cell Model at Level 1 - Behavioural Rule 2

Figure No.(s) in Append. III	RuleSet Name	Rule Number
15	Pallet.TestStartOfAction1	1
21	Pallet.StartAction1	1
27	Pallet.EndAction1	1

Table 13.3: The Three Machine Cell Model at Level 1 - Behavioural Rule 3

Figure No.(s) in Append. III	RuleSet Name	Rule Number
15	Pallet.TestStartOfAction1	2
22	Pallet.StartAction1	2
28	Pallet.EndAction1	2

Table 13.4: The Three Machine Cell Model at Level 1 - Behavioural Rule 4

Figure No.(s) in Append. III	RuleSet Name	Rule Number
15	Pallet.TestStartOfAction1	3
23	Pallet.StartAction1	3
29	Pallet.EndAction1	3

Table 13.5: The Three Machine Cell Model at Level 1 - Behavioural Rule 5		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
15	Pallet.TestStartOfAction1	4
24	Pallet.StartAction1	4
30	Pallet.EndAction1	4

Table 13.6: The Three Machine Cell Model at Level 2 - Behaviural Rule 1		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	1
16	Pallet.CheckLoadUnloadStation	1

Table 13.7: The Three Machine Cell Model at Level 2 - Behavioural Rule 2		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	1
36	Pallet.StartAction2	1
48	Pallet.EndAction2	1

Table 13.8: The Three Machine Cell Model at Level 2 - Behavioural Rule 3		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	2 & 4
37 & 38	Pallet.StartAction2	2 & 3
48 & 49	Pallet.EndAction2	2 & 3

Table 13.9 The Three Machine Cell Model at Level 2 - Behavioural Rule 5

Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	3 & 4
37 & 38	Pallet.StartAction2	2 & 3
48 & 49	Pallet.EndAction2	2 & 3

Rule 3 of Station.NextStationAvailable2:

```
IF :BufferType='ExchangeStore
   .InputBufferAvailable
   OutputBufferAvailable
THEN self;
```

Figure 13.1	*The Three Machine Cell at Level 2 - Behavioural Rule 4*	**LUT-FMS Research Group**

InferenceEngine.TestStartOfPalletsActions2:

```
THEN (— self ExecuteObjects
        'TestStartOfAction2
        $InstWorkingMemory
        :AllocatedPallets);
```

InferenceEngine.StartPalletsActions2:

```
THEN (— self ExecuteObjects
        'StartAction2
        $InstWorkingMemory
        :AllocatedPallets);
```

InferenceEngine.EndPalletsActions2:

```
THEN (— self ExecuteObjects
        'EndAction2
        $InstWorkingMemory
        :AllocatedPallets);
```

Figure 13.2	*The Three Machine Cell at Level 2 - Behavioural Rule 6*	**LUT-FMS Research Group**

Table 13.10: The Three Machine Cell Model at Level 2 - Behavioural Rule 7

Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	5
39	Pallet.StartAction2	4
49	Pallet.EndAction2	4

Table 13.11: The Three Machine Cell Model at Level 2 - Behavioural Rule 8

Figure No.(s) in Append. III	RuleSet Name	Rule Number
32	Pallet.TestStartOfAction2	7
41	Pallet.StartAction2	6
50	Pallet.EndAction2	6

Table 13.12: The Three Machine Cell Model at Level 2 - Behavioural Rule 9

Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	2 & 4
37 & 38	Pallet.StartAction2	2 & 3
48 & 49	Pallet.EndAction2	2 & 3

Table 13.13: The Three Machine Cell Model at Level 2 - Behavioural Rule 10

Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	3 & 4
37 & 38	Pallet.StartAction2	2 & 3
48 & 49	Pallet.EndAction2	2 & 3

Table 13.14: The Three Machine Cell Model at Level 2 - Behavioural Rule 11

Figure No.(s) in Append. III	RuleSet Name	Rule Number
31	Pallet.TestStartOfAction2	2 & 4
37 & 38	Pallet.StartAction2	2 & 3
48 & 49	Pallet.EndAction2	2 & 3

Table 13.15: The Three Machine Cell Model at Level 2 - Behavioural Rule 12		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
32	Pallet.TestStartOfAction2	9
43	Pallet.StartAction2	8
52	Pallet.EndAction2	8

Table 13.16: The Three Machine Cell Model at Level 3 - Behaviural Rule 1		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	1
16	Pallet.CheckLoadUnloadStation	1

Table 13.17: The Three Machine Cell Model at Level 3 - Behavioural Rule 2		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	1
55	Pallet.StartAction3	1
63	Pallet.EndAction3	1

Table 13.18: The Three Machine Cell Model at Level 3 - Behavioural Rule 3		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	2 & 4
56 & 57	Pallet.StartAction3	2 & 3
63 & 64	Pallet.EndAction3	2 & 3

Table 13.19: The Three Machine Cell Model at Level 3 - Behavioural Rule 5

Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	3 & 4
56 & 57	Pallet.StartAction3	2 & 3
63 & 64	Pallet.EndAction3	2 & 3

Rule 3 of Station.NextStationAvailable2:

```
IF :BufferType='ExchangeStore
   .InputBufferAvailable
   OutputBufferAvailable
THEN self;
```

Figure 13.3	*The Three Machine Cell at Level 3 - Behavioural Rule 4*	**LUT-FMS Research Group**

InferenceEngine.TestStartOfPalletsActions3:

```
THEN (- self ExecuteObjects
        'TestStartOfAction3
        $InstWorkingMemory
        :AllocatedPallets);
```

InferenceEngine.StartPalletsActions3:

```
THEN (- self ExecuteObjects
        'StartAction3
        $InstWorkingMemory
        :AllocatedPallets);
```

InferenceEngine.EndPalletsActions3:

```
THEN (- self ExecuteObjects
        'EndAction3
        $InstWorkingMemory
        :AllocatedPallets);
```

Figure 13.4	*The Three Machine Cell at Level 3 - Behavioural Rule 6*	**LUT-FMS Research Group**

Table 13.20: The Three Machine Cell Model at Level 3 - Behavioural Rule 8

Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	5
58	Pallet.StartAction3	4
64	Pallet.EndAction3	4

Rule 1 of DecisionCentre.DetectToolingConflicts:

```
THEN (— self ExecuteObjects
        'CheckToolsAvailability
        $InstWorkingMemory:
        StartablePallets);
```

Figure 13.5	*The Three Machine Cell at Level 3 - Behavioural Rule 7*	**LUT-FMS Research Group**

Table 13.21: The Three Machine Cell Model at Level 3 - Behavioural Rule 9

Figure No.(s) in Append. III	RuleSet Name	Rule Number
54	Pallet.TestStartOfAction3	7
60	Pallet.StartAction3	6
65	Pallet.EndAction3	6

Table 13.22: The Three Machine Cell Model at Level 3 - Behavioural Rule 10

Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	2 & 4
56 & 57	Pallet.StartAction3	2 & 3
63 & 64	Pallet.EndAction3	2 & 3

Table 13.23: The Three Machine Cell Model at Level 3 - Behavioural Rule 11		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	3 & 4
56 & 57	Pallet.StartAction3	2 & 3
63 & 64	Pallet.EndAction3	2 & 3

Table 13.24: The Three Machine Cell Model at Level 3 - Behavioural Rule 11		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
53	Pallet.TestStartOfAction3	2 & 4
56 & 57	Pallet.StartAction3	2 & 3
63 & 64	Pallet.EndAction3	2 & 3

Table 13.25: The Three Machine Cell Model at Level 3 - Behavioural Rule 13		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
54	Pallet.TestStartOfAction3	9
62	Pallet.StartAction3	8
67	Pallet.EndAction3	8

Table 13.26: The Three Machine Cell Model at Level 2 - Behavioural Rule 13		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
39	Pallet.StartAction2 Pallet.ScheduleSubOps2 (A Lisp procedure)	4

Table 13.27: The Three Machine Cell Model at Level 2 - Behavioural Rule 14		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
39	Pallet.StartAction2 Pallet.CellToolRequirement (A Lisp procedure)	4

Table 13.28: The Three Machine Cell Model at Level 3 - Behavioural Rule 14		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
68	ToolTransporter.TestStartOfAction	1 & 2
70	ToolTransporter.StartAction	1
74	ToolTransporter.EndAction	1

Table 13.29: The Three Machine Cell Model at Level 3 - Behavioural Rule 15		
Figure No.(s) in Append. III	RuleSet Name	Rule Number
69	ToolTransporter.TestStartOfAction	3
71	ToolTransporter.StartAction	2
75	ToolTransporter.EndAction	2

Rule 1 of ToolTransporter.FillSparePositions:

IF .RequiredTools
.RequiredPositions>:CurrentLocation.SparePositions
THEN sparepos←:CurrentLocation.SparePositions
tools←(← self FindANumberOfTools sparepos)
:CurrentLocation:ToolsInPTS←
(APPEND :CurrentLocation:ToolsInPTS tools)
(ChangeToolLocation tools :CurrentLocation);

Rule 2 of ToolTransporter.FillSparePositions:

IF reqdtools←.RequiredTools
THEN :CurrentLocation:ToolsInPTS←
(APPEND :CurrentLocation:ToolsInPTS reqdtools)
:ToolsOnIt←(LDIFFERENCE :ToolsOnIt reqdtools)
(ChangeToolLocation reqdtools :CurrentLocation);

Figure 13.6	*Tool Flow of the Three Machine Cell - Operational Rule 6*	LUT-FMS **Research Group**

Rule 1 of ToolTransporter.ExchangeTools:

THEN .ExchangeWithWornTools
.FillSparePositions;

Rule 2 of ToolTransporter.ExchangeTools:

IF .ThereAreToolsLeft
THEN .ExchangeWithUnusableTools;

Figure 13.7	*Tool Flow of the Three Machine Cell - Operational Rule 7*	LUT-FMS **Research Group**

Table 13.30: The Three Machine Cell Model at Level 3 - Behavioural Rule 18

Figure No.(s) in Append. III	RuleSet Name	Rule Number
69	ToolTransporter.TestStartOfAction	4
71	ToolTransporter.StartAction	3
76	ToolTransporter.EndAction	3

Chapter 14
DECISION RULE AND DATA INPUT TO THE THREE MACHINE CELL MODEL

14.1 Introduction

This chapter focuses on the decision rule and data input of the three machine cell model for each of the three levels. These rules, which are not provided by the company and thus are not reflected in the description of the cell's operation in Chapter 12, need to be entered by the user through the Workstation keyboard.

14.2 Scope of the Input

As discussed in Chapter 8, each modelling level requires a different level of information input. However, to compare the three levels, the inputs of these levels have to be made mostly compatible to each other.

It is obvious that the behavioural rules, as described in the previous chapter, are not adequate for the operation of the cell because non of the rules handles the conflicts among the basic work and tool flow activities in the cell. Thus decision rules have to be entered to control these aspects of the cell's operation.

These decision rules, which are not available from the company, are proposed and given below in both general statement form and rule language form in which they will be used in the modelling system for the case study.

Major aspects of the data input include the part and process information, and parameters of the cell elements modelled at each level.

14.3 Decision Rule Input

14.3.1 Initial Comments

The decision rules for the operation of the three machine cell for each of the three

levels are given in this section. Although the modelling system allows the convenient entry of complex alternative decision rules, only one combination of rules are proposed below to cater for the comparison of the three levels. The full investigation of these rules will be done in the next case study given in Chapters 16 and 17.
These rules are described in statement form in the text, while the exact rules which will be used in the modelling system for the cell study are presented in figures.

The role of these decision rules is to model the interactions among the basic work and tool flow activities which are modelled by the behavioural rules as given in Chapter 13. For a detailed discussion of this issue, refer to Chapter 8. The user interactions involved in the entry of these rules have been described in Chapter 9.

14.3.2 Rule Input of Level 1

Since a fixed number of pallets have been assumed for the cell, the load/unload station priority rule for the introduction of pallets into the cell does not need to be entered. Furthermore, the transporter priority rule is not necessary in this case as the cell only has one transporter.

Therefore control of the three machine cell at this level is characterised by the following decision rules:

(1) Part Release Rule:

Parts are released according to the shortest machining time rule, i.e. part that has the shortest total processing time on the assigned stations is released first if there are several parts waiting for a pallet in order to be released (Figure 14.1).

(2) Pallet Priority Rule to Allocation to Transporter:

If there are more than one pallets waiting to be transported, the pallet that has the longest waiting time is selected and allocated to the transporter (Figure 14.2).

(3) Pallet Priority Rule for Station Loading:

If there are several pallets in the local buffer of a station and are waiting to be loaded, the one which has the longest waiting time is loaded first when the station becomes available (Figure 14.2).

(4) Station Priority Rule for Assignment of Pallet:

If there are more than one stations which can be assigned to an operation of the part on pallet, the station which has the least workload is selected. The workload of a station is computed by taking into account the cummulative busy time and the total processing time of the parts queueing in the buffer of the station (Figure 14.3).

14.3.3 Rule Input of Level 2

As mentioned above, since the cell has a fixed number of pallets and only one work transporter, there is no need to enter the load/unload station and transporter priority rules. Besides, as the input buffer of the three machines can only accommodate one pallet and the load/unload stations do not have any local buffers, the pallet priority rule for station loading is not necessary.

Thus the following decision rules need to be entered to control the operation of the cell:

(1) Part Release Rule:

If there are several parts that can be released when a pallet becomes available, the one which has the shortest total processig time is released first (Figure 14.1).

(2) Pallet Priority Rule for Allocation to Transporter:

If there are more than one pallets waiting to be transported, the one that has been waiting for the longest is selected and allocated the transporter first (Figure 14.2).

(3) Station Priority Rule for Assignment of Pallet:

If there are several stations which can be assigned to a pallet, the station which has the least workload is selected. The workload for a machine is computed by taking into account the cummulative busy time of the machine and the processing time of the part queueing in the input buffer of the machine. As for the load/unload stations, this is the cummulative busy time of the station (Figure 14.3).

14.3.4 Rule Input of Level 3

Again, the load/unload station and work transporter priority rules do not need to be entered. However, since there are three men in the cell which are used as tool transporters, the tool transporter priority rule has to be entered. In addition, the pallet priority rule for allocation to tool transporters are also required to be specified.

As a result, the control of the cell at this level can be facilitated by the following decision rules:

(1) Part Release Rule:

If there are several parts that can be released when a pallet becomes available, the one which has the shortest total processing time is released first (Figure 14.1).

(2) Pallet Priority Rule for Allocation to Transporter:

If there are more than one pallets waiting to be transported, the one that has been waiting for the longest is selected and allocated the transporter first (Figure 14.2).

(3) Station Priority Rule for Assignment of Pallet:

If there are several stations which can be assigned to a pallet, the station which has the least workload is selected. The workload for a machine is computed by taking into account the cummulative busy time of the machine and the processing time of the part queueing in the input buffer of the machine. As for the load/unload stations, this is the cummulative busy time of the station (Figure 14.3).

Notably, this is one of the the decision points where the influence of tools on work flow can be represented as rules. However, to make the cell's operation of this level compatible to that of the other two levels, the above rule is entered. A detailed investigation of this aspect is to be conducted in the next case study given in later chapters.

(4) Tool Transporter Priority Rule for Assignment of Pallet:

If there are several tool transporters available, the priority for the three transporters (1, 2 and 3) is in decreasing order (Figure 14.4).

(5) Pallet Priority Rule for Allocation of Tool Transporter:

If there are a number of pallets waiting for transporters so that tools can be tranported for the part on the pallet, the one which has the longest waiting time is selected and allocated to the tool transporter (Figure 14.5).

14.4 Data Input

14.4.1 Scope of Data Input

The scope of the data input for each of the three levels has been discussed in Chapter 8 and the user interactions involved in the data input process have been described in Chapter 9.

As considerable data is involved in the modelling of the cell for each level, the detailed description of the information is given in Appendix VI.2. In the following sections, consideration is only given to the peculiar aspects associated with the data input requirement for each level.

14.4.2 Data Input of Level 1

The data input requirement of the three machine cell model at this level includes the production requirements information, the machine information, the load/unload station information, the transporter information, the pallet information and the process information (Appendix VI.2). For all the stations, the user does not need to specify the type and size(s) of the local buffers. And as the load/unload stations do not have any local buffers in the real cell, the station exchange time is specified as 0, though at this level loading/unloading activities are assumed for these stations.

An average time value, rather than a distance matrix, is specified for the transporting activities between stations. This is one of the major approximations which have been made in the modelling system.

For the process information, the first and the last operations correspond to the palletisation and depalletisation processes respectively. The time of the first operation is specified as 0, because no specific operation times were provided for palletisation and depalletisation, and only the total time for re-palletisation was available.

A total time is required for each machining operation, which is obtained by summing up all the tooling operaion times and the associated tool change times given by the company.

Figure 14.6 summarizes the system parameters and strategies necessary for the operation of the three machine cell at level 1.

14.4.3 Data Input of Level 2

The data input of the three machine cell model for level 2 consists of the production requirements information , machine data, load/unload station data, work transporter data, temporary storage data, pallet data, tool data and process information (Appendix VI.2).

The production requirements and the transporter data are identical to what is specified for level 1, but special consideration has to be given to the specification of machines and load/unload stations. A dual-type pallet exchange buffer is specified for each of the three machines, and the two load/unload stations are specified to have no local buffers and therefore, the station exchange time is also 0. Besides, tool exchange time and index time have to be entered for each machine.

Since each pallet stand can only accommodate one pallet, the capacity of the twenty-two temporary storages is set to 1. For each of the tools , the tool life and the maximum percentage tool life utilisation are also need to be entered. Details of tool information can be found in Appendix VI.1.

As mentioned in the previous chapter, each pallet may be sent to a unique pallet stand. Therefore, a specific temporary storage has to be assigned to each pallet. This is not required at level 1.

In addition to the above peculiar considerations, the process information for this level needs to be specially specified. For each machining operation, the total time is no longer required, in stead a list of sub-operations have to be defined. The sub-operations of a machining operation are the tooling operations, and the cutting time and tool type identity have to be entered for each of them.

For a summary of the major cell parameters and strategies at this level, see Figure 14.7.

14.4.4 Data Input of Level 3

To model the three machine cell at level 3, the following information is required: production requirements data, machine data, load/unload station data, work transporter data, temporary storage data, pallet data, tool data, tool transporter data, STS data and process information (see Appendix VI.2).

As can be expected, except the additional information on the machine PTS capacities, tool transporters and the STS, the other information is identical to what is specified for the level-2 modelling in the previous section.

The PTS capacity of the three machines needs to be specified because the tool provision to a machine has to take into account the size of the magazine. For each tool transporter, the average time between tool stores is also required.

Figure 14.8 summarizes the system parameters and strategies that are considered in the level-3 modelling.

14.5 Conclusions

It is evident, from the above discussion, that the input effort required for the three levels is significantly different from each other. This is due to the fact that first each level models a different range of cell elements that need to be defined, secondly for the same element there may be a different list of parameters to be specified at each level, and thirdly there can be a differing number of decision rules to be entered for each level and the complexity of these rules can also be varying from level to level.

It was estimated, based on the author's experience, that the input time for level 1 was about 30 minutes, whereas 90 and 110 minutes were required for level 2 and level 3 respectively. The difference between level 1 and level 2 or 3 is mainly caused by the requirement of differing levels of process information, and the extra tool information required for level 2 and 3. The difference between level 2 and level 3 is not as significant as the above because the additional input effort required for level 3 is only used for the specification of tool transporters and the STS, plus the entry of some additional decision rules. However, if tool requirements planning is not to be conducted at level 2, the difference between level 2 and level 3 will be considerable since there will be no need to enter the enormous tool data for level 2.

Rule 1 of ReleaseConflictSet.

ShortestTotalProcessingTime:

```
IF partselected←(PickLowObj self
                 'GetShortestTotalTime
                 :FilteredPartTypes)
THEN :PartTypeSelected←partselected;
```

Figure 14.1	*The Three Machine Cell Model - Part Release Rule*	**LUT-FMS Research Group**

Rule 1 of PalletConflictSet.

FirstInFirstOut:

```
IF palletselected←(PickLowObj self
                   'GetFinishTime
                   :Pallets)
THEN :PalletSelected←palletselected;
```

Figure 14.2	*The Three Machine Cell Model - Pallet Prior. Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

LeastWorkLoad:

```
IF nextstation←(PickLowObj self
                'GetWorkLoad
                :NextStations)
THEN :NextStationSelected←nextstation;
```

Figure 14.3	*The Three Machine Cell Model - Stn. Prior. Rule*	**LUT-FMS Research Group**

Rule 1 of ToolTransporterConflictSet.

SpecifiedPriority:

```
IF transporter-( PickHiObj    self
                 'GetPriority
                 :ToolTransporters)
THEN :ToolTransporterSelected-transporter;
```

Figure 14.4	*The Three Machine Cell Model - Tool Transp. Rule*	LUT-FMS **Research Group**

Rule 1 of ToolingConflictSet.

FirstInFirstOut:

```
IF pallet-(PickLowObj self
                 'GetFinishTime
                 :Pallets)
THEN :PalletSelected-pallet;
```

Figure 14.5	*The Three Machine Cell Model - Palt. Tooling Rule*	LUT-FMS **Research Group**

Planning Horizon: 24 Hours (1440 mins)

Releasing: Shortest Total Processing Time

Station Selection: Least Workload

Scheduling: FIFO

FMC: - 3 machines with infinite part buffers
- 1 rail guided vehicle
- 2 load/unload stations with infinite part buffers
- 6 pallet types (one for each)
- 40 parts

Figure 14.6	*K. B. Modelling Level 1 of 3 Machine Cell - System Strategies*	LUT-FMS Research Group

Planning Horizon: 24 Hours (1440 mins)

Releasing: Shortest Total Processing Time

Station Selection: Least Workload

Scheduling: FIFO

FMC: - 3 machines with dual pallet exchange buffer
- 1 rail guided vehicle
- 2 load/unload stations
- 22 temporary storages
- 6 pallet types (one for each)
- 40 parts

Operation Assignment: part by part

Tool Life Management:
- permissible life 90%
- tool life as specified
- machine rationalisation

Tool Assignment: all required tool are available in each PTS

Figure 14.7	*K. B. Modelling Level 2 of 3 Machine Cell - System Strategies*	LUT-FMS Research Group

Planning Horizon: 24 Hours (1440 mins)

Tool Management: Workpiece-Oriented

Releasing: Shortest Total Processing Time

Station Selection: Least Workload

Scheduling: FIFO

Tool Issue: Differential Kitting

FMC: - 3 machines with dual pallet exchange buffer & 120-tool PTS
- 1 rail guided vehicle
- 2 load/unload stations
- 22 temporary storages
- 6 pallet types (one for each)
- 40 parts
- mutually exclusive part & tool flow
- STS of unlimited capacity
- 3 tool transporters

Operation Assignment: part by part

Tool Life Management:
- permissible life 90%
- tool life as specified
- cell rationalisation

Tool Assignment: all required tools are available in STS

Figure 14.8	*K.B. Modelling Level 3 of 3 Machine Cell - System Strategies*	LUT-FMS Research Group

Chapter 15
RESULTS OF THE KNOWLEDGE BASED MODELLING OF THE THREE MACHINE CELL

15.1 Introduction

The work reported here is the results of the modelling of the three machine cell. The operation of each of the three levels is described. The results obtained are summarised and discussed against the major aspects of the cell. The emphasis is to compare the results of the three levels. Criteria are then identified which help the user to select an appropriate modelling level for particular applications.

15.2 Initial Comments

As mentioned in Chapter 12, the purpose of this case study is to compare the three levels of modelling. Thus only one run is planned for each level, based on the operational rules and data input described in the previous two chapters.

Since the operational conditions have been made compatible among the three levels, the results presented here are expected to be able to demonstrate the effects of those assumptions which are made within each of the three levels. In the meanwhile, as each level is functionally self-contained, the relative merits of each level can also be shown through the comparison of the insight provided by the three levels.

15.3 Operation of the Multiple Levels

15.3.1 Initial Comments

Basically speaking, the computational performance of a particular level depends on the speed of rule matching and execution. However, rule matching and execution is determined by the number and complexity of the rules which are used to represent the operation of the level. The number and complexity of the rules contained withtin aparticular level, on the other hand, are mainly influenced by three factors, i.e. the number of objects contained in the working memory, the number of considered parameters of the objects in the modelling and the number of modelled states of the cell elements.

This section discusses the operation of the three levels, with emphasis placed on the investigation of the difference among the computational performances of the three levels.

15.3.2 Operation of Level 1

There are altogether six cell elements which are considered in the modelling at this level (Figure 15.1). The parameters modelled for each of these objects have been described in Chapter 8. As discussed in Chapter 14, a pallet can be in any of the eight states, i.e. *AwaitingForPalletisation, Palletisation, AwaitingForTransfer, Transfer, AwaitingForProcessing, Processing, AwaitingForDepalletisation and Depalletisation.* Therefore, there are twelve transformational rules used in the modelling. Among these rules, four are used for the testing of start of pallets' actions, four for the start of actions, and four for the end of actions.

As a result of the above considerations, the computer run time of the three machine cell model has been found to be 12 minutes. Compared with the 24 hour planning horizon, this figure offers a very promising potential for using level 1 to quickly estimate the performance of a cell.

15.3.3 Operation of Level 2

Although twenty-two temporary storages and ninety-three tools are contained in the working memory of level 2 besides the elements modelled at level 1, only siz temporary storages have been assigned to pallets. Thus there are altogether 105 cell elements which are modelled (Figure 15.2). The parameters of each of these objects have been described in Chapter 8.

At this level, a pallet can be in any of the eleven states, i.e. *AwaitingForPalletisation, Palletisation, AwaitingFor- Transfer, ReadyForTransfer, TransferAndUnloading,. AwaitingFor- Processing, LoadingAndProcessing, AwaitingForUnloading, UnloadingFromStation, ReadyForDepalletisation and*

Depalletisation. There are nineteen transformational rules to be used for the modelling of these states. Among these rules, seven are used for testing the start of pallets' actions, six for the start of actions, and six for the end of actions.

Due to the above factors considered in the modelling, the computer run time of the three machine cell model at this level has been shown to be 43 minutes for the 24 hour period production. In comparison with level 1, this is 31 minutes slower. However this run time is still more than 30 times faster than the real production.

15.3.4 Operation of Level 3

At this levcel, four more cell elements are added to the working memory, that is, the STS and three tool transporters (Figure 15.3). For a detailed description of the parameters of these objects modelled, refer to Chapter 8.

In addition to the eleven states of a pallet, the tool transporter can be in six possible states, i.e. *Idle, Arriving- AndLoading, ReadyForPTS, TransferAndExchange, ReadyForSTS and TransferAndUnloading*. This results in ten more transformational rules to be used in the modelling. Among these rules, four are used for testing the start of tool transporters' actions, three for the start of actions, and three for the end of actions.

As a result, 53 minutes are required to run the model at this level for the 24 hour production. This is 41 minutes and 10 minutes longer tahn level 1 and level 2 respectively.

15.4 Summary and Discussion of Results

15.4.1 Initial Comments

A summary and dissicusion of the results obtained from each of the three levels of modelling follows. In particular, the results are interpreted against the structure and operation of the cell. The detailed listing of these results is presented in Appendix VI.3.

15.4.2 Output of Level 1

A summary of the results obtained from the level 1 modelling run is given in Figure 15.4. It is found that the make span is 1447.35 minutes, and the total lateness is therefore 7.35 minutes. This production lateness is caused by the third component of part 6. From the station or part operation schedules, it can be easily seen that this is mainly due to the successive release and assignment of the components of part 6, which has a very long machining time, to machine 3. In addition, since only one pallet is assigned to parts 5 and 6, part 6 can not be released into the cell until all the components of part 5 have been completed evenif machine 1 is available.

As a results of the long machining time for all parts, the average part lead time is 154.33 minutes. Among the eight part types, part 4 has the highest throughput rate, i.e. 4 components can be completed withtin a shift. This is because of the fact that part 4 has the shortest machining time. As part 6 has the longest machining time and is released after part 5, its throughput rate is the lowest (0.99 part/shift), though the waiting time of this part is the shortest (1.78 minutes on the average). The lead time of part 5 is the longest, because it has a very long machining time and waiting time in the cell and the sequence of parts' release into the cell has little influence on the part lead times.

From the part performance figures it can be seen that the time a part is at a station is equal to its machining time. For the part can be immediately unloaded into the machine's infinite local buffer on completion of the part's operation.

Since the selection of a station for operation assignment is done according to least workload rule, the use of the three machines and two load/unload stations is found to be well- balanced, with the slight longer use of machine 3 being caused by the long machining time of part 6. Besides, the utilization of the three machining centres is above 77%, leading to the conclusion that there is not much redundancy on the use of three machines in the cell.

The utilization of the two load/unload stations is fairly low ranging from 34.2% to 36.27%. Even if only one AGV is used, its utilization is still extremely low (4.42%). This is because the average transfer time between stations is assumed to be 0.8 minutes which is far shorter than the machining time of the components.

The maximum queue lengths of the three machines are 3, 2 and 3 respectively, and the two load/unload stations also have 2 and 3 pallets at maximum in their local buffers respectively. However, as parts have much longer processing time at machines than at load/unload stations, from the part operation schedlues it can be found that the queueing time of parts at the load/unload stations are far shorter than at machine stations. Therefore if a tow-position buffer is to be provided for the three machines, without blockage, no buffers will be needed at the two load/unload stations because there are only six pallets in the cell. This conclusion coincides with the structure of the actual cell.

15.4.2.1 Long-Term Manufacturing Performance Forecast

From the above results it can be seen that the level 1 modelling is fairly efficient as a result of the major assumptions made in the modelling. Thus level 1 may well be applicable in the prediction of long-term manufacturing performance.

In this case, the production requirement is typically planned for 1 to 6 months, and the total production requirement is usually introduced into the system in batches. Modelling of this type of production calls for an adequately efficient modelling process, with certain approximations accepted. Therefore, experiments can be planned for the three machine cell by increasing the total production requirement from daily to 1 month's, 3 months' or 6 months', other parameters unchanged. Unfortunately, due to the technical problems of the Workstation, these experiments were not completed and no results could be presented here. Further work will fully cover these experiments.

15.4.3 Output of Level 2

A summary of the results from the modelling run at level 2 is shown in Figure 15.5. The make span for the specified production requirement is 1522.09 minutes, and this indicates that the production requirement can not be met within 24 hours under the selected decision rules. This lateness is caused by a component of part 3 and one of part 6.

As a result of the long processing time, part 5 has the longest lead time (196.09 minutes). Although part 6 has no waiting time, its throughput rate is the lowest (0.95

parts/shift) as the components have the longest machining time and are released after the components of part 5.

From the part performance listing, it can be seen that only four components have been sent to the temporary storages though all components can be assigned. This indicates that unless more pallets are introduced for the cell, the twenty-two pallet stands can not be adequately utilized. Additionally seven components stayed at the machine stations longer than the machining times because they could not be unloaded until the output buffer of the stations became clear.

Again the workload assignment to the three machining centres or the two load/unload stations are well-balanced. The stationary time of the machines is no longer zero as a result of the influence of the limited output buffer capacity. Compared with the three machines, the load/unload stations have a longer stationary time because these two stations do not have any local buffers. Since some components have been sent to the temporary storages, more movements were involved and thus the AGV has a longer load run time.

From the temporary stoarge performance listing it was found that only three pallet stands have been used and the utilization of these three stands is very low ranging from 0.14% to 2.35%. Thus it can be concluded that with only six pallets in the cell, there is little need for the use of temporary storages.

The minimum cell tool requirement is 91 tool types and 118 tools. According to the work schedlued to each machine, the tool requirement for the three machines are 85, 54 and 74 tools respectively, resulting in a total of 213 tools required at the maximum. This indicates that if a 120-tool magazine is provided for each machine with all the required tools loaded at the begining of the 24 hour production period, there will be no need for an expensive tool flow management system for the cell.

15.4.4 Output of Level 3

Figure 15.6 summarizes the results obtained from the level-3 modelling run. It is found that the make span is 1737.62 minutes, causing a total lateness of 297.62 minutes. The total throughput rate is eleven parts per shift and the average part lead time is 171.19 minutes.

As a result of the influence of the tool availability on the work flow, part type 5 has the longest lead time (236.73 minutes on the average) even though they are released before part 6. The waiting time of part 6 is still the shortest (4.64 minutes on the average).

From the part performance listing, it can be seen that more components were sent to the temporary storages. This is simply because most components have to wait in the input buffer of the machines until the required tools are transported to the PTS. The time that components spent on waiting for tools ranges from 0 to 48.72 minutes.

For the three machining centres, although the use of them is also well-balanced, their utilization is relatively low since each of the machines has to wait for tools before loading and machining can start.

The utilization of the two load/unload stations and the AGV is fairly low, but the workload assigned to the stations is balanced. From the temporary storage performance figures it is found that all six assigned pallet stands have been utilized (ranging from 0% to 6.44%). It is interesting to see that one component of part 4 was re-loaded onto the transporter immediately after it was unloaded to the pallet stand since one of the machines became available by then. Therefore the stationary time of pallet stand 12 is 0 but the load/unload time 0.54 minutes.

Under the differential kitting tool issue strategy, the cell tool requirement generated was 92 tool types and 212 tools. This is similar to the maximum tool requirement produced at level 2, because all the cutting tools necessary for the machining of the components can easily accommodated within the 120-tool capacity magazine. Very little tool flow was evident in the cell. This can be more easily seen in the machine PTS performance listing.

A maximum of 81 tools are required on any one machine over the 24 hours. Neither worn tools nor position tools were changed at each machine, leading to the conclusion that no tools flew back to the STS (therefore the final content is zero). This implies that no tools were shared across the three machines. Thus it can be suggeted that a magazine of 90-tool capacity would be adequate for each of the three machining centres.

The utilization of the tool transporters is extremely low, ranging from 0.09% to 0.92%. Thus it can be concluded that with these low utilizations the operators can easily perform tool load/unload and transportation.

No tools were found to be worn on completion of the production requirements, though a number of tools, not classed as worn, possess tool life insufficient for further use.

15.4.4.1 The Machine PTS Capacity

From the above results, it can be seen that the 120-tool magazine capacity for the three machines has resulted in no tool sharing between machines. To enable comprehensive tool flow between tool stores (i.e. from the PTSs to the STS), the magazine capacity need to be varied, and it can be foreseen that different values of this parameter should bring differing tool flow performance.

Experiments using the level 3 with varying sizes of magazine capacity specified for the three machines have been performed. Description of these experiments and discussion of the results obtained are presented in Appendix VI.4.

15.5 Comparison of Results

15.5.1 Scope of the Comparison

In this section, a comparison of the results obtained from the three levels of modelling is conducted. First the work flow patterns within the three levels are critically compared, with the difference interpreted. Then the performance of the cell and cell elements is compared in order to show the measure of consistency of the three levels on providing insight of the modelled cell.

15.5.2 Work Flow Patterns

To compare the work flow patterns of the three levels, pallet 1 (i.e., part types 1 and 2) were selected and brought under consideration. For each of the parts 1 and 2, the three major operations defined were represented graphically by narrow right hatch bars, while the empty bars were used to represent all the other activities, such as load/unload, transport and waiting.

As shown in Figure 15.7, part 10001 was palletised 5.35 minutes later at levels 2 and 3 than at level 1. This was because at level 1, parts which were released before part 10001 could be immediately unloaded into the assumed local buffers of the load/unload stations without delay on completion of palletisation.At the other two levels, however, part 10001 had to wait until the two earlier released parts (parts 40001 and 80001) were finished palletisation and transported to their next destinations.

The machining operation of part 10001 at levels 1 and 2 was started at the same time and at the same machine. At level 3, however, since a pallet can only be loaded after the required tools have been transported and the time that the palllet should stay in the input buffer of the machine depends both on the availability of the tool transporters and on the number of tools to be transported, part 10001 was assigned to machine 2 rather than machine 3 as a result of the fact that machine 3's input buffer became available later than that of machine 2. For part 80001 was residing at machine 3 and part 40001 was at machine 2, and they requested 17 and 12 tools to be transported respectively though the tool transporters did not cause any delays. The depalletisation activity of part 10001 was performed 4.14 minutes after the completion of the machining operation at all three levels.

Following the completion of part 10001, part 10002 was released and allocated to pallet 1. From Figure 15.8 it can be seen that this part was assigned to machine 2 at levels 1 and 3, but machine 3 at level 2. The reason for this was that at level 1 the assignment of machines was puerly based on the workload of the machine, and therefore machine 2 was selected even though machine 3 was the first candidate considered. At level 3 part 10002's machining operation was not delayed due to tools availability, because the tools transported for part 10002 can also be used for this part. However, the reson that machine 2 was selected in stead of machine 3 at level 3 was that machine 2 had less workload than machine 3 at the point that the decision was made. Notably at level 2 large delay existed for the machining operation, which was caused by the fact that although machine 3 had the least workload, the operation time of pallet 6 (i.e., part 80002) which was scheduled just before part 10002 happened to be so long (109.92 minutes) that part 10002 had to wait in the input buffer of machine 3.

As shown in Figure 15.9, part 10003's machining operation was significantly delayed at level 3. This was mainly due to that it was the first component of part type 1 assigned to machine 3 and therefore it had to wait at the machine's input buffer until the unavailable tools had been transported to the machine. In addition, this part stayed at the temporary storage for a considerable time before being transported to the machine, and thus it had an extra transport activity between the temporary storage and the machine.

The last component of part type 1 was part 10004, which spent most of its waiting time in the local buffer of machine 3 at level 1 before being loaded to the machine. Although at level 3 this part did not wait for tools, it still waited at the input buffer of machine 3 (Figure 15.10).

Since pallet 1 can only take part types 1 and 2, the first component of part 2 was released into the cell immediately after the four components of part 1 were finished. Part 20001 spent fairly similar amount of time queueing at the three levels before it was loaded for the machining operation. As part type 2 only required four more tools than part 1 (Table VI.1 and VI.2), part 20001 only spent 6.16 minutes on waiting for tools (Figure 15.11).

Although part 20002 was assigned to machine 2 rather than 3 at level 3, it could still use the tools of part type 1 and therefore it only took 6.16 minutes for part 20002 to wait for the transport of unavailable tools (Figure 15.12). Apart from this, this component did not queue for a long time in the machine's buffer at all three levels.

As shown in Figure 15.13, part 20003 was also smoothly scheduled through the cell without ant delay at levels 2 and 3, whereas at level 1, it spent 11.90 minutes queueing before its machining operation was started. Although all the tools required by part 20004, the last component for pallet 1, were available at the PTS of machine 2, this part had to request some new tools because the tools in the magazine did not have sufficient tool life left for the complete machining of the component. This was shown in Figure 15.14. At level 2, part 20004's machining operation was considerably delayed at machine 2 as a result of the use of the least workload station selection rule which selected machine 2 though machine 3 became available 22.75 minutes earlier than machine 2.

15.5.3 Comparison of the Cell Performance

As a result of the difference in the work flow patterns at the three levels, the cell performance provided by these levels can be expected to be diffrent. In this section, the measure of the diffrence of the three levels with regard to the cell performance will be shown by comparing the performance figures collected. Besides, tool requirements obtained from the level 2 and level 3 modelling are compared, with tool usage frequency and tool life utilization being indicated as well.

15.5.3.1 Overall Cell Performance

The comparison of the overall cell performance is shown in Figure 15.15. As can be expected, for the same production requirements, the make span at level 3 was the longest due to the influence of tool flow on the work flow in the cell. Level 2's make span is 74.74 minutes longer than that of level 1. One of the reasons was that the limited buffers of the stations are a constraining factor for work flow. Although at level 2 the provision of temporary pallet stands can reduce this constraint, the use of these stands has to rely on the availability of the work transporters.

The total throughput rate of level 1 is the highest, and the level 2's is higher than that of level 3. The explanations for this diffrenec in make span also applies here. Due to the same reason, the total lateness and the averagepart flow time are different for the three levels, with level 1 having the shortest and level 3 the longest.

As shown in Figure15.15, the average utilization of the major cell elements is the highest at level 1 and the lowest at level 3. This is mainly due to the fact that at levels 2 and 3 the utilization of the stations was constrained by the limited buffers and by the arrival of tools respectively. A detailed comparison of the utiliozation of the major cell elements will be given in the later sections.

15.5.3.2 Part Throughput and Lead Times

As shown in Figure 15.16, the throughput rate for each part type at level 1 was generally higher than that of the other two levels. The only exceptions are part types 4 and 7. From AppendixVI.3it can be found that part 4 generally spends more time on waiting at level 1 than at the other two levels, and part 7 spends more time on waiting at level 1 than at level 2. The reason for this can be that at level 1, parts tend to queue at stations bu the FIFO station loading rule can delay parts 4 and 7 which have the shortest machining times. At levels 2 and 3, however, since the input buffer size is 1, the FIFO rule had no chance to be applied.

From Figure 15.16 it can also be seen that the throughput rate for each part type at level 2 was higher than that of level 3, with parts 2 and 3 being two exceptions. Again from Appendix VI.3 it was found that the waiting time of parts 2 and 3 at level 2 was longer than at level 3. This was due to that the selection of the station for a part had a significant effect on the performance of the cell. Although these two parts had to spend time on waiting for tools at level 3, the time these parts spent on queueing at buffers could be longer than the total waiting time at level 3 if a station was selected at level 2 which was machining a part that had a long processing time.

Generally speaking, the average lead time for each part type at level 1 is the shortest, and the parts lead time of level 2 should be shorter than that of level 3 (Figure 15.17). The above explanations for throughputs also apply here.

15.5.3.3 Machine Station Utilization

From Figures 15.18, 15.19 and 15.20 it was found that machine 1 was utilized most at level 2, machine 2 at level 1 and machine 3 at level 3. The major reason for this variation was the assignment of alternative machines to an operation and the use of the station availability and selection rules. At level 1, a station is assumed to be always available, and therefore the station selection ruleplayed a key role in assigning parts to machines. In the level-2 modelling, however, a machine is said to be available only if the input buffer and the output buffer are clear, whereas at level 3 although the conditions for machine availability are identical to those for level 2, the status of the input buffer was greatly affected by the fact that parts have to wait in the input buffer of the machines until the required tools have been transported to the PTS.

As a result, thirteen, twelve and twelve parts were assigned to machine 1 at levels 1, 2 and 3 respectively; forteen, fifteen and sixteen parts to machine 2 at levels 1, 2 and 3 respectively; and thirteen, thirteen and twelve parts to machine 3 at levels 1, 2 and 3 respectively.

15.5.3.4 Transporter Performance

As shown in Figures 15.21, the utilization of the AGV for the three levels was very similar though it was used increasingly more from level 1 to level 3. The diffrence between levels 1 and 2 was that some components hasd to be transported to the temporary storages as result of the constraints of the limited buffers of the stations. Thus in addition to the transporting parts between stations, the AGV was also used to move between stations and temporary storages.

At level 3, since parts spent more time in the buffers of the machines, they were more frequently transported to the temporary storages, and therefore the AGV was more utilized.

15.5.3.5 Load/Unload Station Performance

From Figures 15.22 and 15.23, it can be seen that the utilization of the two load/unlooad stations was also very similar for all three levels. The slight difference was made by the fact that the palletisation and depalletisation activities were aissgned to the two load/unload stations and thus there was a need for applying the station selection rule. At level 1, the least workload rule is the only rule for station selection, while at levels 2 and 3, this rule was applied only if the two stations were both available, i.e. in an idle state.

Due to the above fact, different components can be assigned to a load/unload station at different levels, thus leading to the slight variation on the use of the stations .

15.5.3.6 Temporary Storage Performance

As discussed before, temporary storages were modelled at levels 2 and 3. A comparison of the utilization of the temporary storages for these two levels is shown in Figure 15.24. As can be expected, the six assigned pallet stands were utilized more at level 3 than at level 2. The reason for this was because the input buffers of the three machines were more frequently occupied by components at level 3, and therefore components had to be more frequently transported to the temporary storages and stayed there until the input buffers were cleared.

The only exceptions was pallet stand 7, which was utilized for 2.14 minutes at level 2 but 0.00 minutes at level 3. Again this can be seen as the effect of the asignment of alternative stations to components and the application of the station selection rule.

15.5.3.7 Tool Reuqirements and Sister Tooling Prediction

A comparison of the tool requirement generated from the level-2 modelling and the level-3 modelling is shown in Figure 15.25. The difference between the minimum requirement (118 tools) and the maximum requirement (213 tools) or the actual requirement (212 tools) was evident. However, there was almost no difference between the maximum tool requirement generated from level 2 and the actual tool requirement produced at level 3. This coincides with the specification of the 120-tool capacity magazine for each machine, which enables maximum machining flexibility of the machines but results in very little tool flow evidence in the cell, and thus there would be no savings achieved by sharing tools across the machines.

Although the total number of tools required was very similar under both the level-2 maximum modelling and the level-3 modelling, the number of sister tools required for some tool types could differ (Figure 15.26). The major reason for this include part assignment to machines and machine selection.

15.5.3.8 Tool Usage Frequency and Tool Life Utilization

Since different tools were used in the modelling at levels 2 and 3, and the same tool number could be different tool types for different levels, there was no way to match the tool numbers generated at these levels. Therefore, the comparison of the performance of each individual tool can only be made in an aggregate way.

Figure 15.27 shows the usage frequency of tool numbers from 1 to 8 for both levels. The tool life utilization of these tools is depicted in Figure 15.28.

From the above two figures it can be observed that the main factors which determine the tool usage frequency should be the cutting times of the components and the permissible tool life values. The latter is, however, significantly affected by the specification of the maximum percentage tool life utilization. As no tool flow was considered at level 2 and no tools flew back to the STS at level 3, a considerable number of tools were found to be under-utilized.

15.5.4 Conlusions

As mentioned in Chapter 13 and 14, in oder to compare the three levels of modelling, the operational conditions of the three machine cell have been made compatible for these levels by selecting identical decision rules. Since the performance of the cell at each level depends on the work flow patterns, and the work flow patterns are controlled by the decision rules applied and are influenced by the operational assumptions made within each level, the performance as provided by each level can vary.

From the discussion and comparisons conducted in this chapter, it can be concluded that the less the assumption made in the modelling, the more realistic the performance provided by the modelling. From levels 1 to 3, as more and more system operational details are considered, the performance provided by the modelling is decreasingly optimistic but increasingly realistic.

Another important conclusion is that the assignment of alternative stations to an operation of a part in conjunction with the assumptions of the modelling level can significantly influence the work flow pattrens and hence the performance of the cell. Therefore, in order to model a cell realistically under station pooling at different levels, the station selection rules must be carefully developed.

15.6 Comments on Modelling Levels

Based on the discussion on the input requirements and the results of the case study in the previous two chapters and this chapter, it can be concluded that the three levels of modelling require three levels of input, gives three levels of running performance and provides three levels of output. Thus this hierarchical modelling method can ne used at various stages in the design of a flexible machining facility.

To help to choose an appropriate modelling level for the user's particular applications, the following principles can be proposed based on the case study:

(1) Level 1 is the most aggregate among the three levels with regard to the input and output requirements, but it is the most efficient with respect to the computational performance. Therefore, this level is well suited to the basic sizing of a manufacturing system in the early design stage. It is best used to help determine the number of equipments and the rough size of local buffers for the stations. In addition, it can be used for long-term manufacturing performance forcasts.

(2) After the basic size of a manufacturing system has been deterimed using level 1 of the knowledge based modelling system or other modelling methods (see Chapter 5), the user can select level 2 to study the detailed work flow of the system. The effects of variuos types and sizes of the local buffers of the stations and temporary storages on the performance of the system can effectively be assesses. If a prelimibary tool requirements planning is required, the user can also choose this modelling level. The tool requirments generated can help the user to determine appropriate tool management strategies.

(3) For some systems, tool flow management can be a crucial issue. For example, if the total number of tools required at a machine well exceeds the capacity of an available tool magazine, then a secondary tool store and the associated tool provision strategies may have to be considered. In this case, the user can choose level 3 of the modelling system

15.7 Concluding Remarks

In this case study, although the cell has been choosen to be modelled for three shifts, the total number of jobs was only forty. The machining time of a part was so long that the work flow in the cell was not comprehensive enough to assess the response of the modelling system with regard to alternative operational rules. In the level-3 modelling it was found that because of the specification of the large magazines for the three machines, tool flow only occured in terms of providing tools from the STS to the PTSs. There were no tools flowing back the STS and used again by other machines or components.

As a result, there is a need to extend the three machine cell to a case which can be used to comprehensively assess the modelling system subject to more complex work flow and operating strategies. Besides, full-scale experiments are required with regard to the integrated flow of both work and tool in the extended cell.

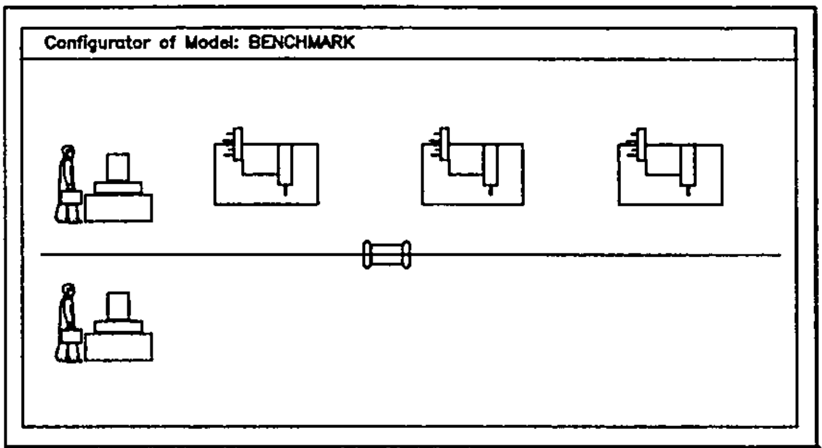


Figure 15.1 | Configuration of the Three Machine Cell at Level 1 | LUT - FMS Research Group

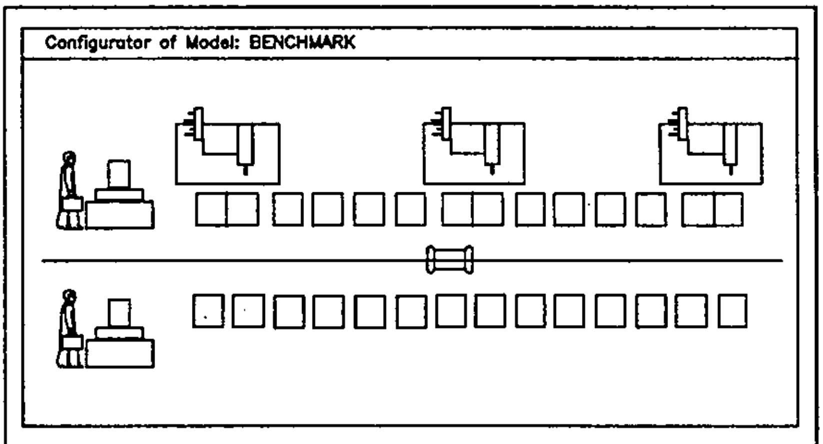


Figure 15.2 | Configuration of the Three Machine Cell at Level 2 | LUT - FMS Research Group

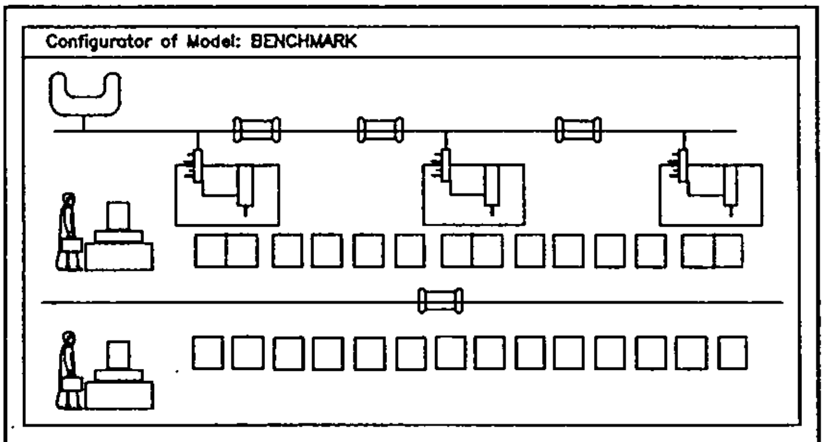


Figure 15.3 | Configuration of the Three Machine Cell at Level 3 | LUT - FMS Research Group

	Required	Actual
Total Part Throughput:	40	40
Time Horizon (mins):	1440.00	1447.35

Machine Performance

	Machine 1 Time	Machine 1 %	Machine 2 Time	Machine 2 %	Machine 3 Time	Machine 3 %
Machining	1125.2	77.7	1124.6	77.7	1252.4	86.5
Load/Unload	26.0	1.8	28.0	1.9	26.0	1.8
Idle	296.1	20.5	294.7	20.4	169.0	11.7
Max. Queue	3		2		3	

Transporter Performance

	Transporter 1 Time	Transporter 1 %
Load Run	64.0	4.4
Empty Run	56.0	3.9
Load/Unload	43.2	3.0
Idle	1284.2	88.7

Figure 15.4 *K. B. Modelling Level 1 of the 3 Machine Cell - Summary of Outputs* **LUT-FMS Research Group**

	Required	Actual
Total Part Throughput:	40	40
Time Horizon (mins):	1440.00	1522.09

Minimum Cell Tool Requirement: 118

Machine Performance

	Machine 1 Time	Machine 1 %	Machine 2 Time	Machine 2 %	Machine 3 Time	Machine 3 %
Cutting	963.6	63.3	854.5	56.1	963.7	63.3
Tool Change	238.6	15.7	247.0	16.2	234.8	15.4
Load/Unload	24.0	1.6	30.0	2.0	26.0	1.7
Stationary	6.0	0.4	4.8	0.3	5.4	0.4
Idle	289.8	19.0	385.9	25.4	292.2	19.2
Tools Used	85		54		74	

Transporter Performance

	Transporter 1 Time	Transporter 1 %
Load Run	67.2	4.4
Empty Run	57.6	3.8
Load/Unload	45.4	3.0
Idle	1351.9	88.8

Figure 15.5 *K. B. Modelling Level 2 of the 3 Machine Cell - Summary of Outputs* **LUT-FMS Research Group**

	Required	Actual
Total Part Throughput:	40	40
Time Horizon (mins):	1440.00	1737.62

Cell Tool Requirement: 212

Machine Performance

	Machine 1 Time	Machine 1 %	Machine 2 Time	Machine 2 %	Machine 3 Time	Machine 3 %
Cutting	920.7	53.0	841.7	48.4	1019.3	58.7
Tool Change	228.8	13.2	247.8	14.3	244.0	14.0
Load/Unload	24.0	1.4	32.0	1.8	24.0	1.4
Stationary	10.57	0.6	5.4	0.3	6.4	0.4
ForTools	135.9	7.8	98.0	5.6	129.9	7.5
Idle	417.7	24.0	512.8	29.5	314.0	18.1
Tools Used	81		55		76	
Worn Tools	0		0		0	

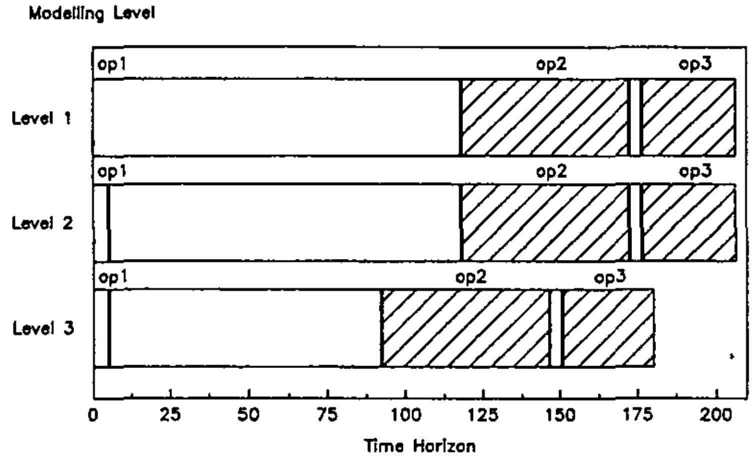
Work Transp. Perfom.

	Transp. 1 Time	Transp. 1 %
Load Run	76.0	4.4
Empty Run	66.4	3.8
Load/Unload	51.3	3.0
Idle	1543.9	88.9

Tool Transp. Perform.

	Transp. 1 Time	Transp. 1 %
Load Run	16.0	0.9
Empty Run	16.0	0.9
Load/Unload	179.4	10.3
Idle	1526.3	87.8

Figure 15.6 *K. B. Modelling Level 3 of the 3 Machine Cell - Summary of Outputs* **LUT-FMS Research Group**



Level \ Sched. / Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu1	0.00	0.00	mc3	118.34	172.62	lu1	176.76	206.76
Level 2	lu1	5.35	5.35	mc3	118.34	172.62	lu1	176.76	206.76
Level 3	lu1	5.35	5.35	mc2	92.45	146.73	lu2	150.87	180.87

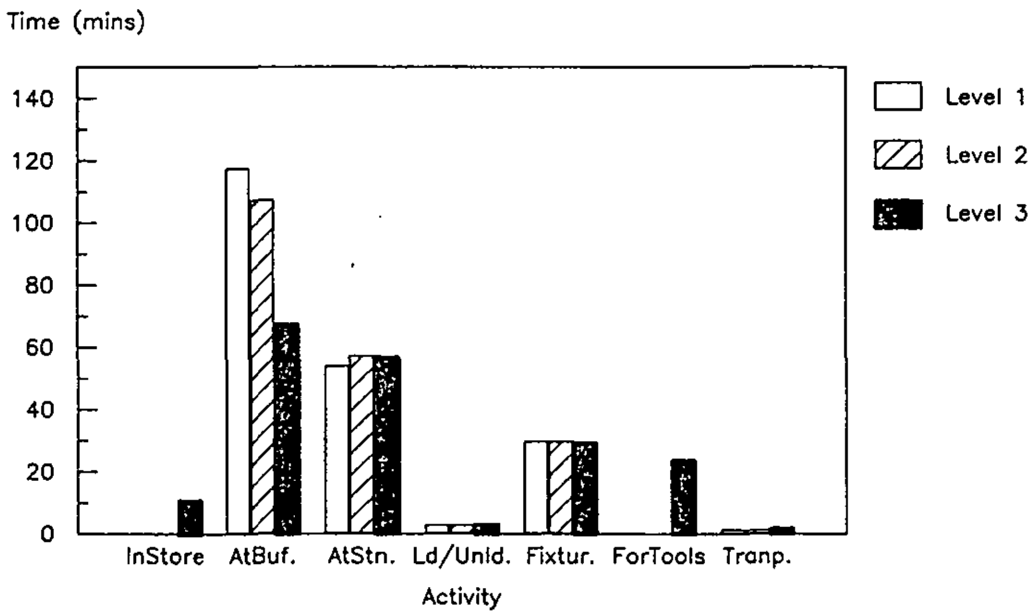
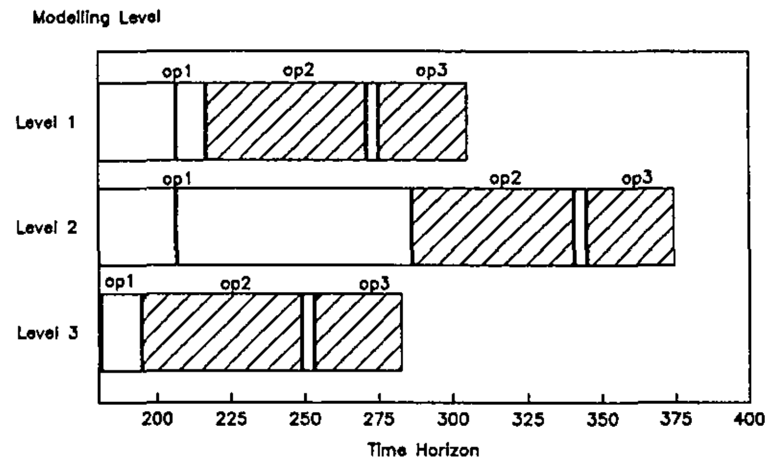


Figure 15.7

Work Flow Patterns and Part Performance - part 10001

LUT-FMS Research Group



Level \ Op. / Sched.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu1	206.76	206.76	mc2	216.79	271.07	lu1	275.21	305.21
Level 2	lu1	206.76	206.76	mc3	286.54	340.82	lu1	344.96	374.96
Level 3	lu2	180.87	180.87	mc2	194.80	249.08	lu1	253.22	283.22

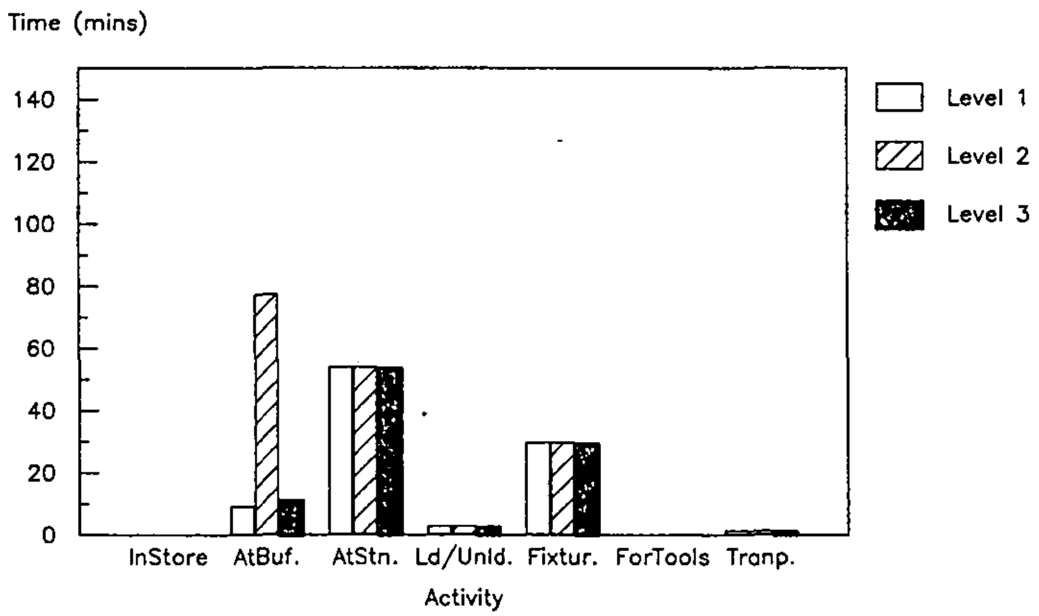
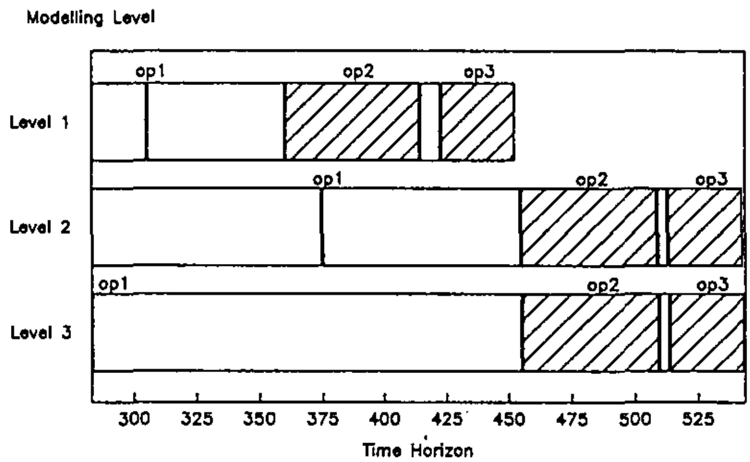


Figure 15.8	*Work Flow Patterns and Part Performance - part 10002*	LUT-FMS Research Group



Level \ Sched. / Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu1	305.21	305.21	mc3	360.19	414.47	lu2	422.72	452.72
Level 2	lu1	374.98	374.98	mc3	454.74	509.02	lu1	513.16	543.16
Level 3	lu1	283.22	283.22	mc3	455.31	509.59	lu2	513.73	543.73

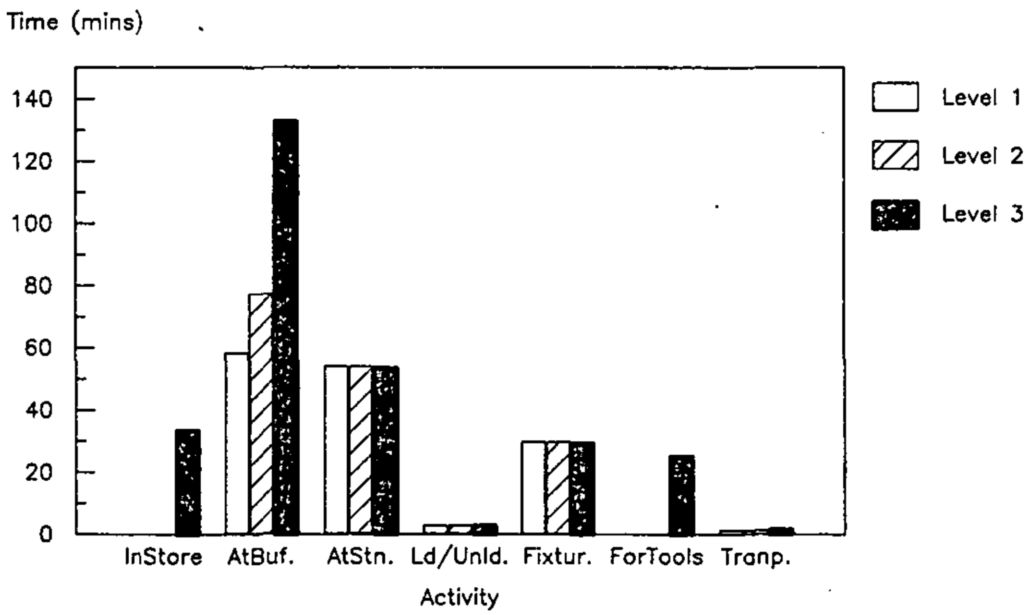
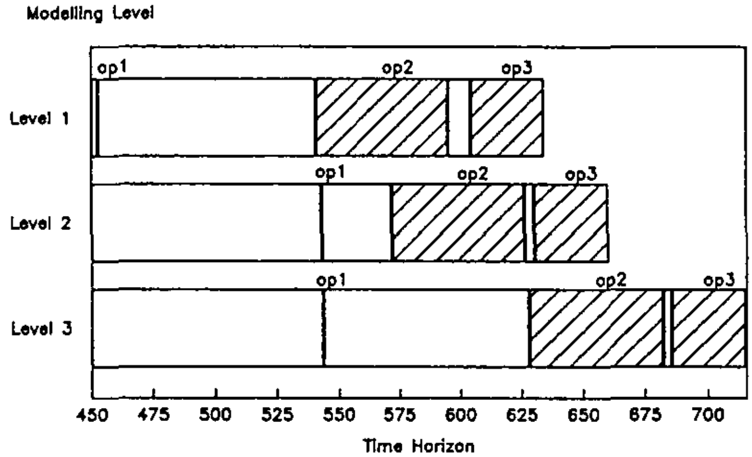


Figure 15.9	*Work Flow Patterns and Part Performance - part 10003*	LUT-FMS Research Group



Level \ Sched. / Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu2	452.72	452.72	mc3	540.96	595.24	lu1	604.51	634.51
Level 2	lu1	543.16	543.16	mc3	572.10	626.38	lu1	630.52	660.52
Level 3	lu2	543.73	543.73	mc3	628.15	682.43	lu2	685.77	715.77

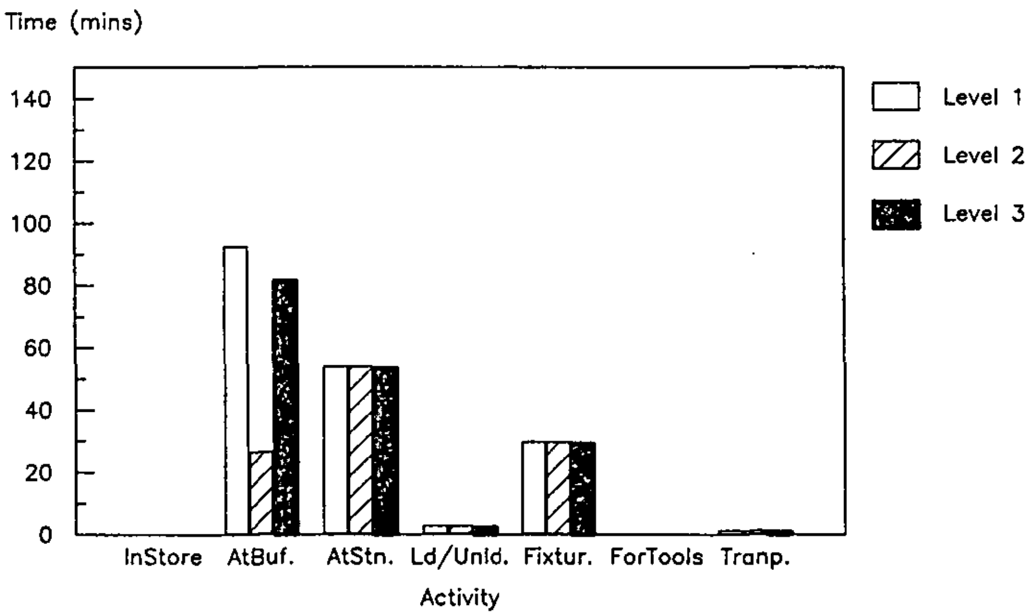
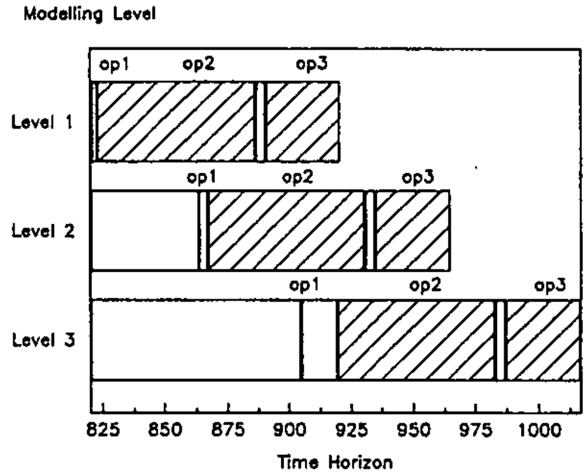


Figure 15.10

Work Flow Patterns and Part Performance - part 10004

LUT-FMS Research Group



Level \ Sch-ed. \ Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu2	820.58	820.58	mc3	822.72	886.19	lu1	890.33	920.33
Level 2	lu1	863.44	863.44	mc2	867.13	930.60	lu2	934.74	964.74
Level 3	lu1	904.76	904.76	mc2	919.24	982.71	lu1	986.85	1016.85

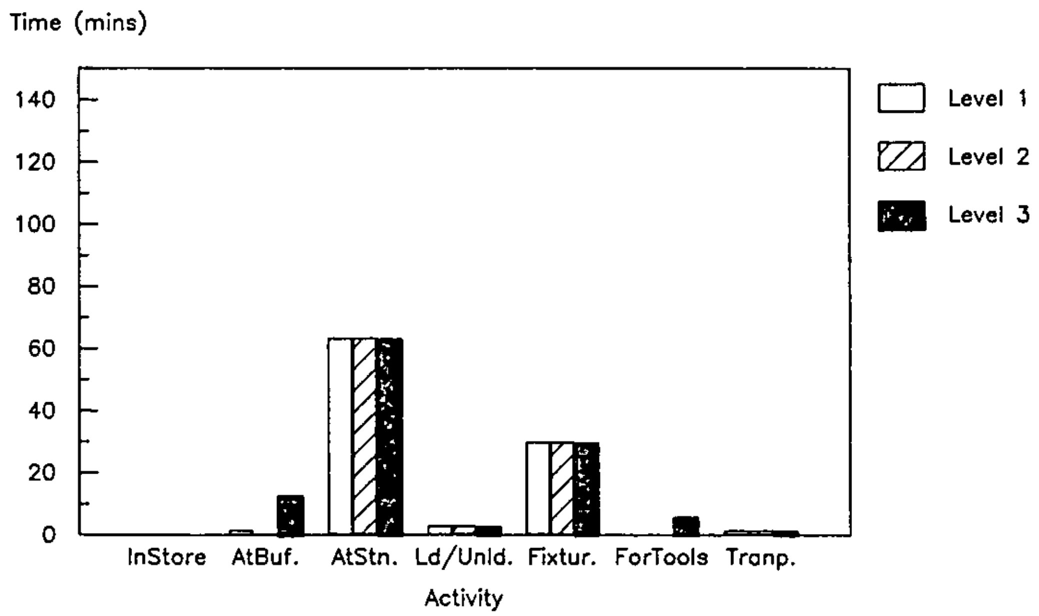
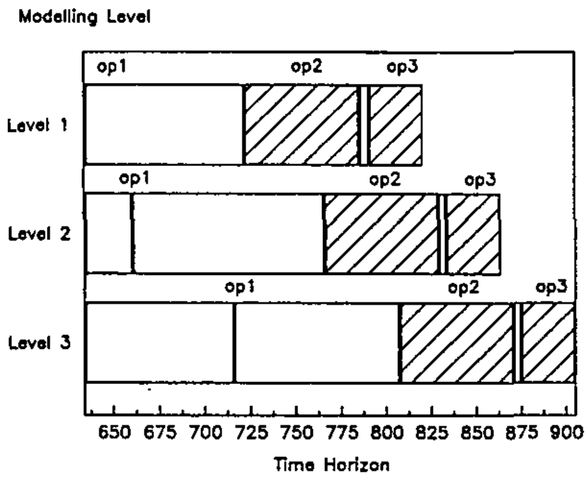


Figure 15.11

Work Flow Patterns and Part Performance - part 20002

LUT-FMS Research Group



Level \ Sch-ed. \ Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu1	634.51	634.51	mc3	721.73	785.20	lu2	790.58	820.58
Level 2	lu1	660.52	660.52	mc3	765.83	829.30	lu1	833.44	863.44
Level 3	lu2	715.77	715.77	mc3	807.15	870.62	lu1	874.76	904.76

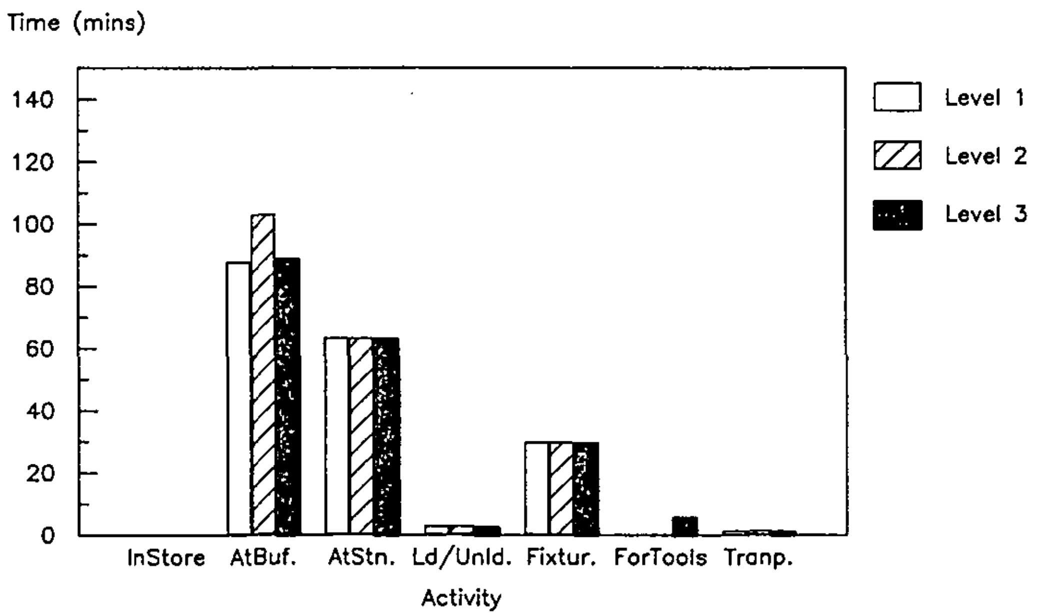
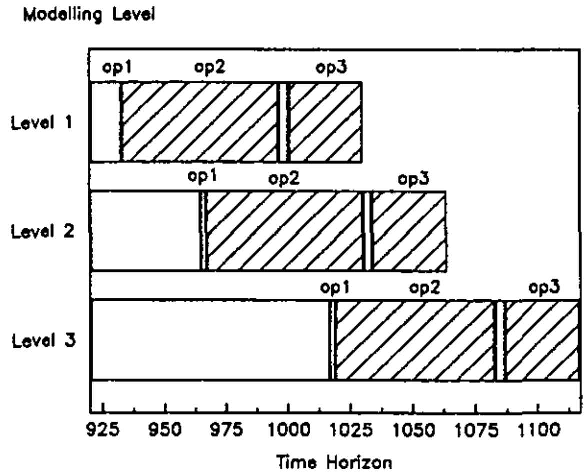


Figure 15.12

Work Flow Patterns and Part Performance - part 20001

LUT-FMS Research Group



Level \ Op. / Sched.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu1	920.33	920.33	mc2	932.77	996.24	lu2	1000.38	1030.38
Level 2	lu2	964.74	964.74	mc2	966.88	1030.32	lu1	1033.69	1063.69
Level 3	lu1	1016.85	1016.85	mc2	1018.99	1082.46	lu1	1086.60	1116.60

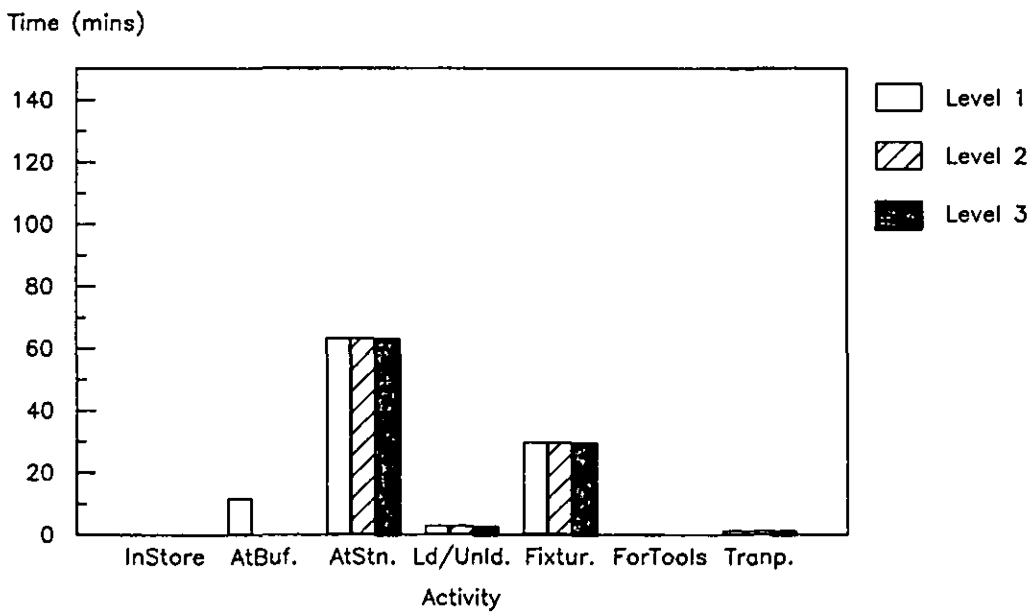
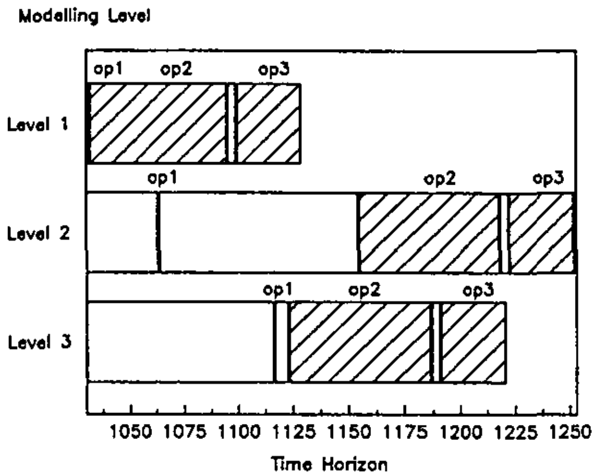


Figure 15.13	*Work Flow Patterns and Part Performance - part 20003*	LUT-FMS Research Group



Level \ Sched. Op.	op1			op2			op3		
	Station	Start Time	Finish Time	Station	Start Time	Finish Time	Station	Start Time	Finish Time
Level 1	lu2	1030.38	1030.38	mc2	1031.72	1095.19	lu1	1099.33	1129.33
Level 2	lu1	1063.69	1063.69	mc2	1154.86	1218.33	lu2	1222.47	1252.47
Level 3	lu1	1116.60	1116.60	mc2	1123.38	1186.85	lu2	1190.99	1220.99

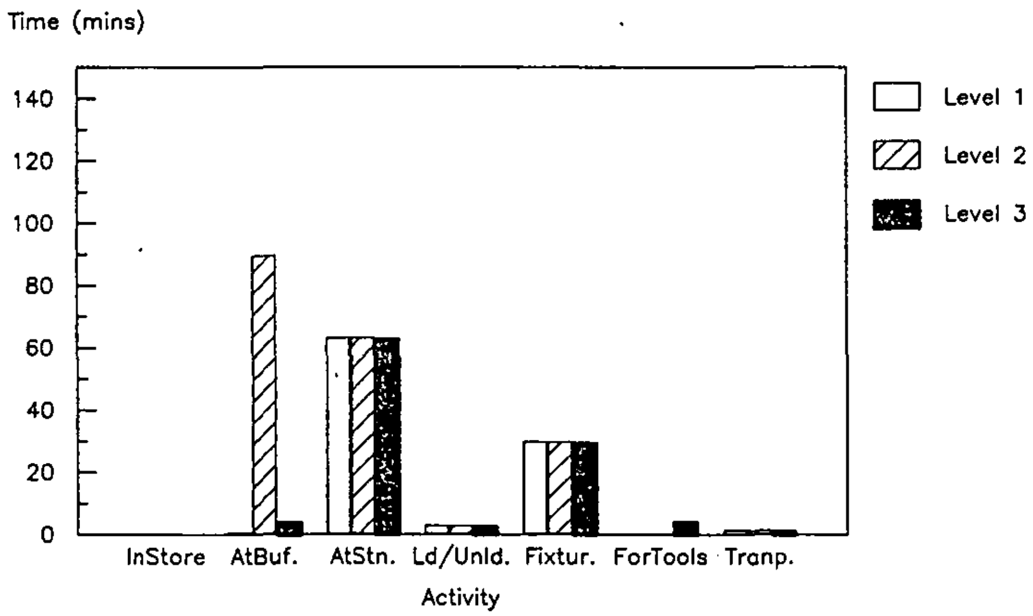


Figure 15.14	*Work Flow Patterns and Part Performance - part 20004*	LUT-FMS Research Group

Level	Make Span (mins)	Total Throughput (parts/shift)	Total Lateness (mins)	Av. Flow Time (mins)	Average Util. (%)
1	1447.35	13	7.35	154.33	63.59
2	1522.09	12	82.09	161.21	61.31
3	1737.62	11	297.62	171.19	57.98

Figure 15.15 *Comparison of Overall Cell Performance* LUT - FMS **Research Group**

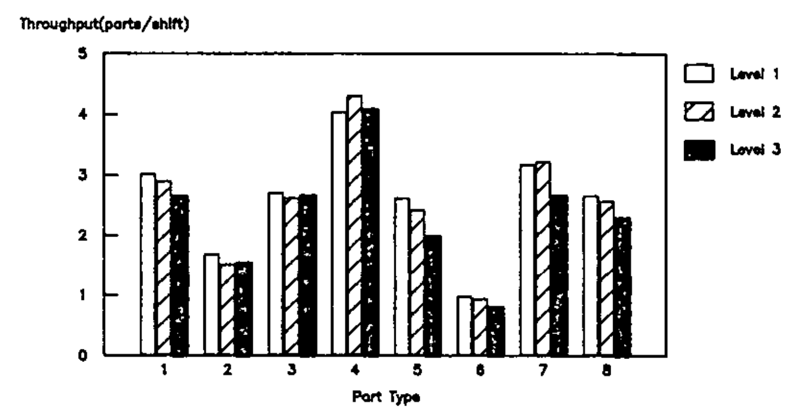


Figure 15.16 *Comparison of Part Throughputs* LUT - FMS **Research Group**

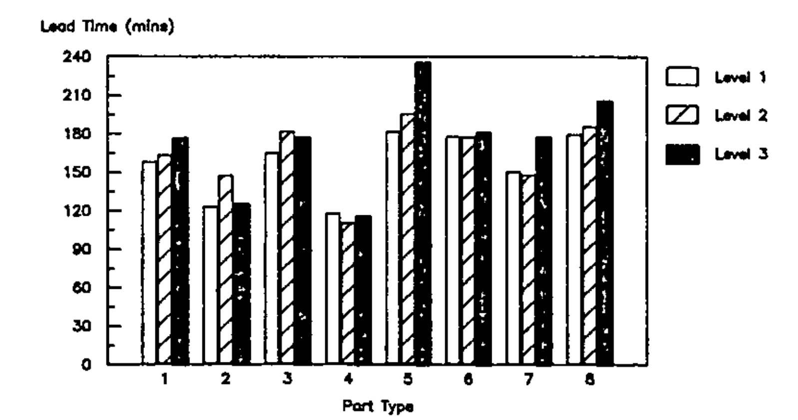


Figure 15.17 *Comparison of Part Lead Times* LUT - FMS **Research Group**

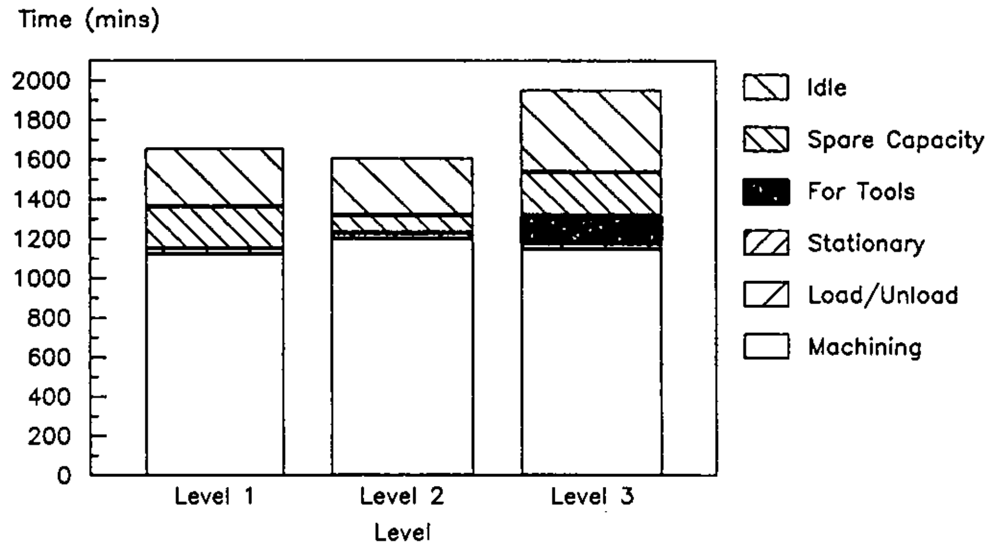


Figure 15.18	*Comparison of Machine Utilization - Machine 1*	LUT - FMS **Research Group**

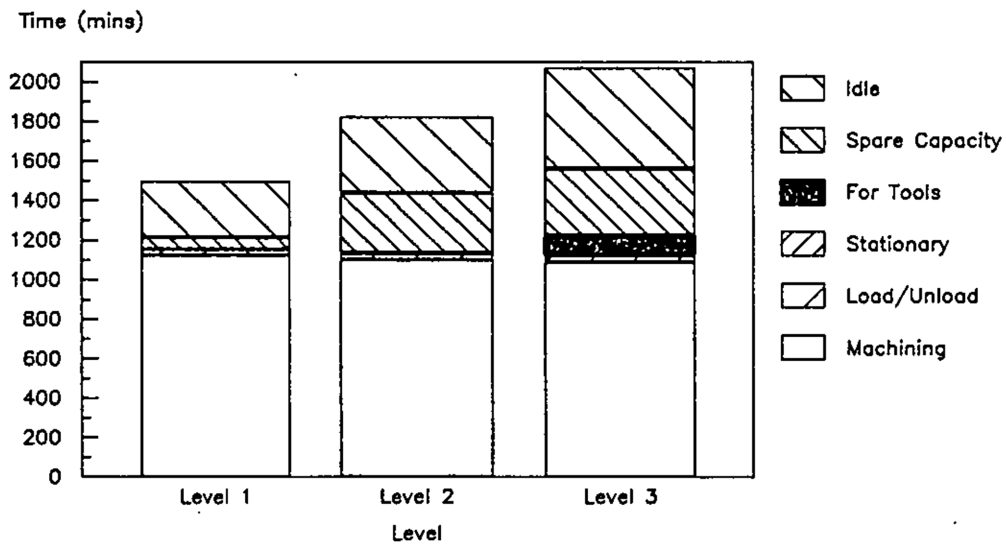


Figure 15.19	*Comparison of Machine Utilization - Machine 2*	LUT - FMS **Research Group**

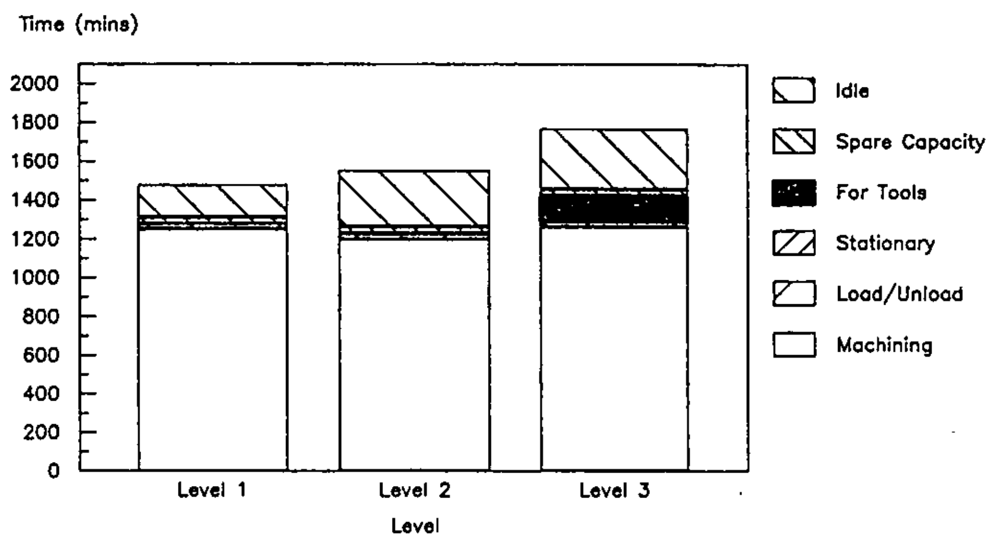


Figure 15.20	*Comparison of Machine Utilization - Machine 3*	LUT - FMS Research Group

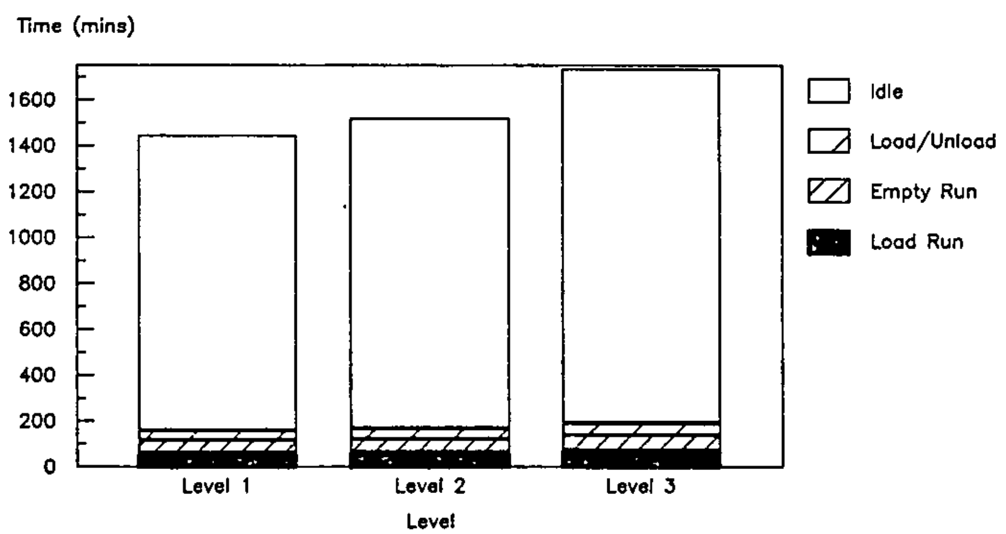


Figure 15.21	*Comparison of Transporter Performance*	LUT - FMS Research Group

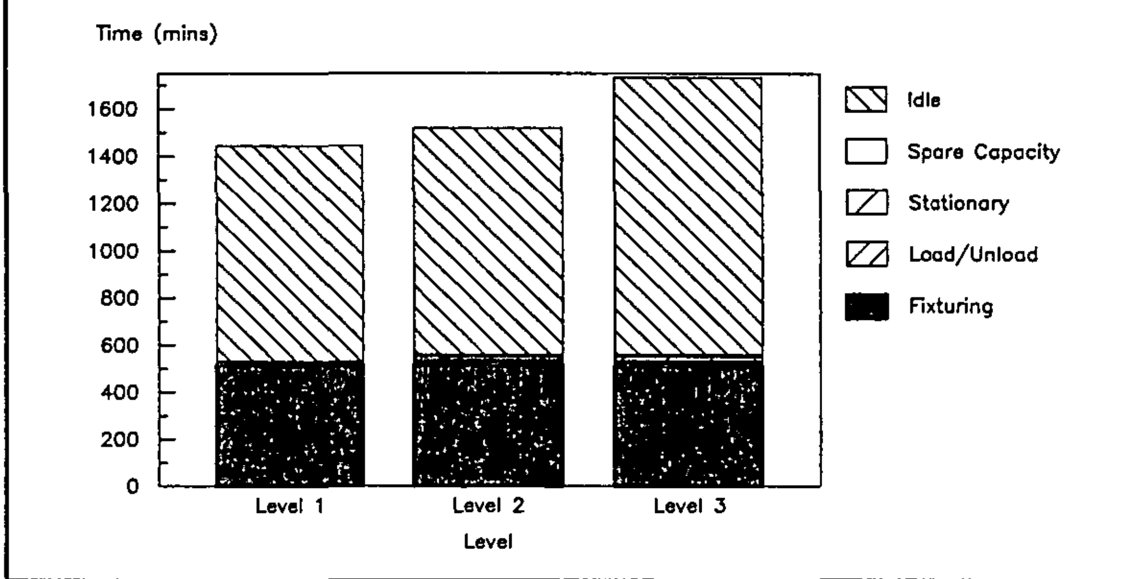


Figure 15.22

Comparison of Load/Unload Station Performance - Load/Unload Stn. 1

LUT - FMS Research Group

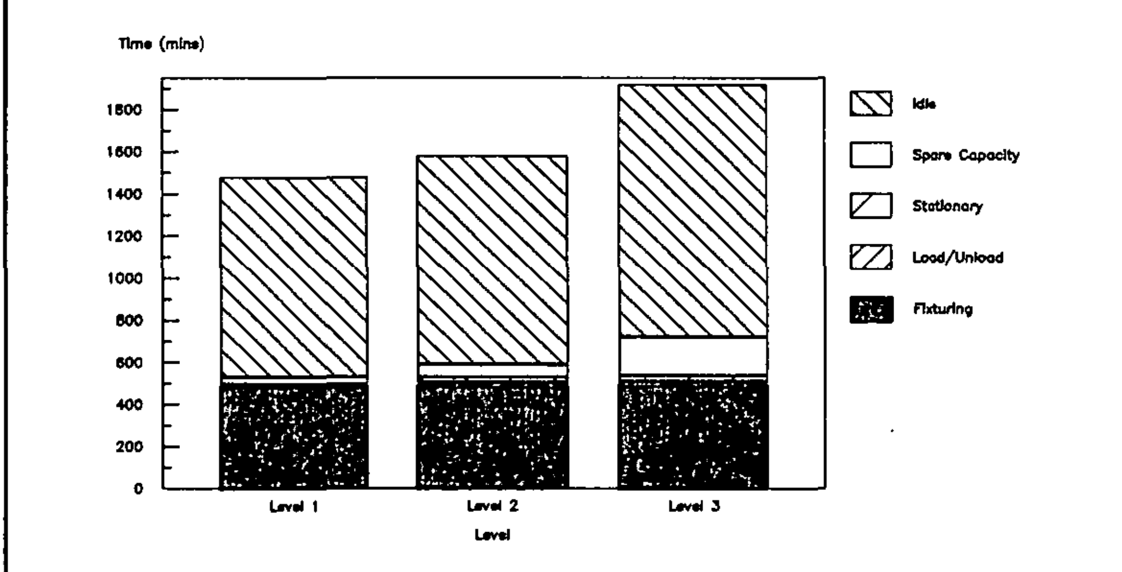


Figure 15.23

Comparison of Load/Unload Station Performance - Load/Unload Stn. 2

LUT - FMS Research Group

Assigned Pallet Stands	Stationary (mins)		Load/Unload (mins)		Spare Capacity (mins)		Idle (mins)	
	Level 2	Level 3	Level 2	Level 3	Level 2	Level 3	Level 2	Level 3
ps 4	0.00	93.27	0.00	2.16	1522.09	1044.36	1522.09	1642.19
ps 6	2.14	40.66	0.54	1.08	1034.86	1084.00	1519.41	1695.88
ps 7	2.14	0.00	0.54	0.54	653.89	1643.88	1519.41	1737.08
ps 8	0.00	17.97	0.00	1.08	1522.09	1277.88	1522.09	1718.57
ps 11	35.71	111.86	1.08	2.16	1082.38	916.85	1485.30	1623.60
ps 14	0.00	45.83	0.00	1.08	1522.09	1416.94	1522.09	1690.71

Figure 15.24	*Comparison of Temporary Storage Performance*	LUT - FMS Research Group

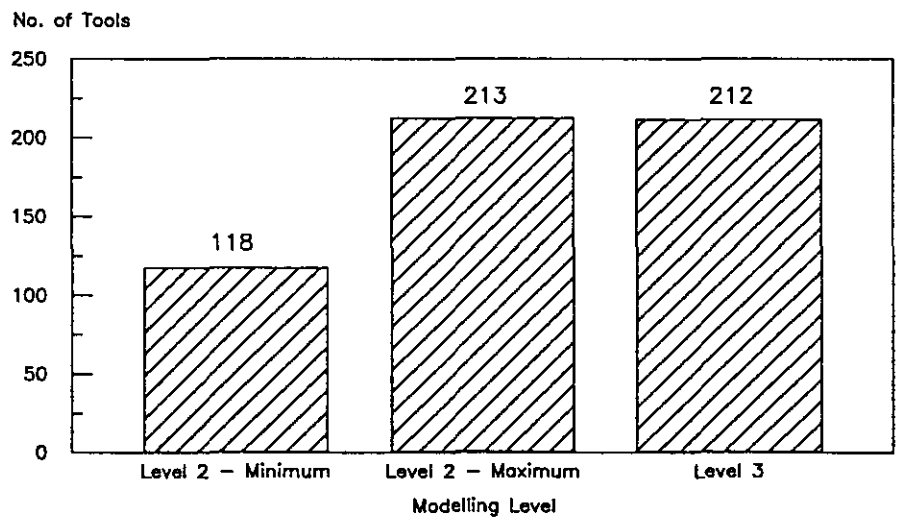


Figure 15.25	*Comparison of Tool Requirement*	LUT - FMS Research Group

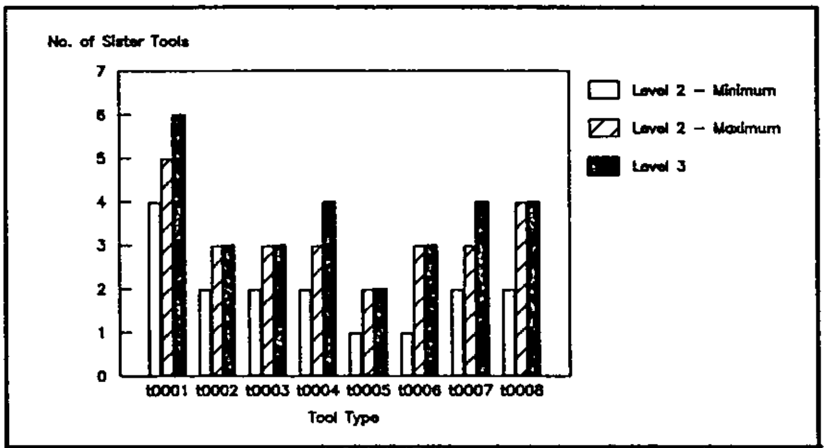


Figure 15.26	*Comparison of Sister Tools*	**LUT - FMS** Research Group

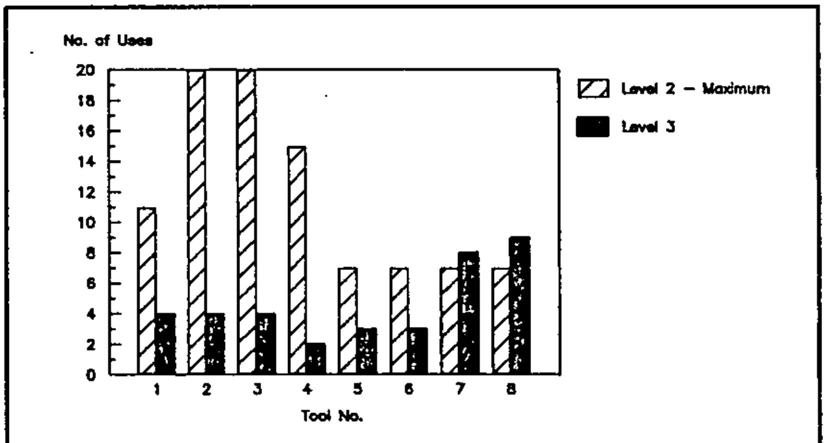


Figure 15.27	*Comparison of Tool Usage Frequency*	**LUT - FMS** Research Group

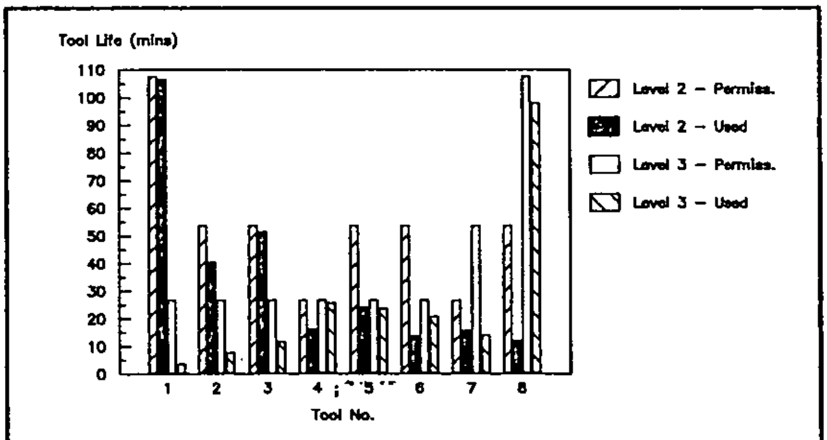


Figure 15.28	*Comparison of Tool Life Utilization*	**LUT - FMS** Research Group

Chapter 16
THE EXTENDED CELL AND THE INPUT TO THE MODELLING SYSTEM

16.1 Introduction

In this chapter, the extended cell is introduced for the comparative study which is to be carried out with the knowledge based modelling system, the Emulator and the the tool flow modelling system. A description of the extended cell is given first. Then the decision rule and data input to the knowledge based modelling system is illustrated. Results of the knowledge based modelling of the extended cell are to be presented in the next chapter.

16.2 The Extended Cell

16.2.1 Scope of the Extention

In order to bring the modelling system into a more comprehensive test, the three machine cell as studied in the previous chapters is to be extended in several directions. Firstly a more complex parts spectrum is to be specified, which wii enable more complex work flow in the cell. Then two more machining centres will be added so that the cell can have the adequate machining capacity for the new production requirements. To demonstrate the capability of the modelling system in providing insight into the tool flow behaviour of a machining cell, an appropriate tool magazine capacity is to be specified. In addition, alternative decision rules will be expressed and studied with regard to the operation of the cell.

16.2.2 The Components

There are twenty parts which have been specified for the extended cell based on the parts information of the three machine cell. Details of the daily requirements of these parts are summarized in Figure 16.4. In total, 101 parts are required to be produced. For the process details of each of these twenty parts, refer to Tables VII.5 to VII.24 in Appendix VII.1.

16.2.3 The Cell Design

The extended cell comprises of five Makino MC 1210 horizontal machining centres, see Figure 16.1, serviced by a rail guided vehicle. Work fixturing and defixturing is performed mannually at either of the two load/unload stations (Figure 16.2).

Each machine is also equipped with a 40 tool capacity tool magazine or primary tool store. Tool change between the PTS and the spindle is performed by a double ended arm which selects the required tool from the magazine whilst machining is in progress, then changes the tool when the spindle has stopped.

The cell has 18 pallet stands, each of which can be used as a temporary storage to accommodate a pallet. Pallet interchange beween the pallet transporter and the machine is executed via a pallet changer. This provides a dual-type pallet buffer for each of the five machines.

The cell uses ten pallets, each of which can be used for any parts. The operating strategy is such that a fully machined part is obtained for each visit to a machine. Work flow within the cell is carried out by the AGV, while the other supporting activities , such as inspection and cleaning, are wholly mannual. Tool transfer between the STS and PTSs, loading and unloading are all carried out by the three men available in the cell.

16.2.4 Basic Operation of the Cell

The basic operation of the extended cell is similar to that of the three machine cell, see section 12.6. Thus the behaviural rules for the three machine cell, as described in Chapter 13, will all be used in the extended cell as well.

However, two addtional production strategies have been specified in oder to control thw work flow in the cell. The first strategy corresponds to batch production, where components are released into the cell batch by batch. Here, a batch is defined as a group of identical components. As shown in Table VII.1of AppendixVII.1, twenty batches have benn specified corresponding to the twenty part types. The size of each of these batches is also illustrated in Table VII.1.

The second strategy is for kit production, where kits of parts are brought through the cell in order to meet certain assembly requirement. A kit, in this case, can be defined as an order which consists of components of different part types. Table VII.2 in Appendix VII.1 shows the five part kits specified for this case study. The order quantity for each of these kits is shown in Table VII3. In contrast to the batch production, components are released into the cell kit by kit in kit production. The kit machining list has been specified as shown in Table VII.4.

16.3 Decision Rule Input

16.3.1 Initial Comments

The decision rules for the operation of the extended cell for each of the three levels are given in this section. In order to investigate the influence of alternative control strategies on the performance of the cell, varying combinations of decision rules are to be proposed below. Each combination of decision rules in conjunction with the specified data input constitutes a specific run in the modelling experiment.

Althgough different rules can be entered for each decision point, alternative rules are mainly developed in this case study with regard to three decision points as a result of the existing of a huge number of combinations of rules. One is the batch or kit production decision, another is the station selection decision and the third is the part scheduling decision. Each of the developed rules is described in statement form in the text, while the exact rule which will be used in the modelling system for the cell study are presented in figures.

The role of these decision rules is to model the interactions among the basic work and tool flow activities which are handled by the behavioural rules as discussed in the previous section.

16.3.2 Rule Input of Level 1

16.3.2.1 Batch Production

The following runs are planned for the modelling of the extended cell for batch production at level 1:

(1) ***Run 1:***

- *Part Release Rule:* Parts are released into the cell according to the order number, i.e. part that is the first on the machining list is released first.

- *Pallet Priority Rule for Allocation to Transporter:* If there are more than one pallets waiting to be transported, the one that has the longest waiting time is selected and allocated to the transporter (Figure 16.6).

- *Pallet Priority Rule for Station Loading:* If there are several pallets in the local buffer of a station and are waiting to be loaded, the one which has the longest waiting time is loaded first when the station becomes available (Figure 16.6).

- *Station Priority Rule for Assignment of Pallet:* If there are more than one stations which can be assigned to an operation of a part, the station which has the shortest queue length is selected (Figure 16.7).

(2) ***Run 2:***

All the rules applied in Run 1 are also used for this run, except the following variation:

- *Station Priority Rule for Assignment of Pallet:* If there are more than one stations which can be assigned to an operation of a part, the station which is the earliest available is selected. The available time of a station is computed by taking into account the finish time of the work currently being processed and the total processing time of the components queueing in the buffer of the station (Figure 16.8).

(3) ***Run 3:***

All the rules used in Run 1 are also used here, except the following variation:

- *Station Priority Rule for Assignment of Pallet:* If there are several stations available, the one which has the least work load is selcted. The workload of a station is computed by taking into account the cummulative busy time and the total processing time of the parts queueing in the buffer of the station (Figure 16. 9).

16.3.2.2 Kit Production

The following runs are planned for the modelling of the extended cell for kit production at level 1:

(1) ***Run 4:***

This run uses the same rules as those of Run 1.

(2) ***Run 5:***

This run uses the same rules as those of Run 2.

(3) ***Run 6:***

This run uses the same rules as those of Run 3.

16.3.3 Rule Input of Level 2

16.3.3.1 Batch Production

The following runs are planned for the modelling of the extended cell for batch production at level 2:

(1) ***Run 7:***

- *Part Release Rule:* Parts are released into the cell according to the order number, i.e. part that is the first in the machining list is released first.

- *Pallet Priority Rule for Allocation to Tranporter:* If there are more than one pallets waiting for a transporter, the pallet that has the longest waiting time is selcted and allocated to the transporter (Figure 16.6).

- *Station Priority Rule for Assignment of Pallet:* If there are several stations available, the one which has the most spare spaces is selected. The spare spaces of a station is computed by taking into account the position on the station and the spare positions of its local buffer (Figure 16.10).

- *Temporary Storage Use Rule:* Parts are not to be sent to any temporary storages in this run.

(2) ***Run 8:***

This run uses the same rules as Run 7, except the following variation:

- *Station Priority Rule for Assignment of Pallets:* If there are more than one stations which can be assigned to an operation of a part, the station which is the earliest available is selected. The available time of a station is computed by taking into account the finish time of the work currently being processed and the total processing time of the components queueing in the buffer of the station (Figure 16.8).

(3) ***Run 9:***

This run uses the same rules as Run 7, except the following variation:

- *Station Priority Rule for Assignment of Pallets:* If a part can be assigned to more than one stations, the one that has the least workload is selected. The workload of a station is computed by taking into account the cummulative busy time and the total processing time of the parts queueing in the buffer of the station (Figure 16. 9).

(4) ***Run 10:***

This run uses the same rules as Run 7, except the following variaition:

- *Pallet Priority Rule for Allocation to Tranporter:* If there are several pallets waiting for a transporter, the one that has the shortest remaining processing time is selected and allocated to the transporter. The remanining processing time of a part is computed by summing up the operation times of the remaining operations (Figure 16.11).

16.3.3.2 Kit Production

The following runs are planned for the modelling of the extended cell for kit production at level 2:

(1) ***Run 11:***

This run uses the same rules as Run 7.

(2) ***Run 12:***

This run uses the same rules as Run 8.

(3) ***Run 13:***

This run uses the same rules as Run 9.

(4) ***Run 14:***

This run uses the same rules as Run 10.

16.3.4 Rule Input of Level 3

16.3.4.1 Batch Production

The Following runs have been planned for the modelling of the extended cell for batch production at level 3:

(1) ***Run 15:***

- *Part Release Rule:* Parts are released into the cell according to the order number, i.e. part that is the first in the machining list is released first.

- *Pallet Priority Rule for Allocation to Tranporter:* If there are more than one pallets waiting for a transporter, the pallet that has the longest waiting time is selected and allocated to the transporter (Figure 16.6).

- *Station Priority Rule for Assignment of Pallet:* If there are several stations available, the one which has the most spare spaces is selected. The spare spaces of a station is computed by taking into account the position on the station and the spare positions of its local buffer (Figure 16.10).

- *Temporary Storage Use Rule:* Parts are not to be sent to any temporary storages in this run.

- *Tool Transporter Priority Rule for Assignment of Pallet:* If there are more than one tool transporters available, the priority for the three transporters (1, 2 and 3) is in decreasing order (Figure 16.12).

- *Pallet Priority Rule for Allocation of Tool Transporter:* If there are several pallets waiting for a tool transporter, the one that has the longest waiting time is selected and allocated to the tool transporter (Figure 16.13).

(2) ***Run 16:***

All the rules applied in Run 15 are also used in this run, except the following variation:

- *Station Priority Rule for Assignment of Pallets:* If there are more than one stations which can be assigned to an operation of a part, the station which is the earliest available is selected. The available time of a station is computed by taking into account the finish time of the work currently being processed and the total processing time of the components queueing in the buffer of the station (Figure 16.8).

(3) ***Run 17:***

All the rules applied in Run 16 are also used in this run, except the following variation:

- *Machine Priority Rule for Assignment of Pallet:* If there are several machines available, the one which has fewest tools unavailable in the PTS is selected and allocated to the pallet (Figure 16.14).

16.3.4.2 Kit Production

The following runs are planned for the modelling of the extended cell for kit production at level 3:

(1) ***Run 18:***

This run uses the same rules as Run 15.

(2) ***Run 19:***

This run uses the same rules as Run 16.

(3) ***Run 20:***

This run uses the same rules as Run 17.

16.4 Data Input

16.4.1 Scope of Data Input

The data input requirement of the extended cell model is similar to that of the three machine cell model. As considerable data is involved in the modelling for each of the three levels, the detailed description of the information is given in Appendix VII.2. In the following sections, consideration is only given to the peculiar aspects associated with the data input requirement of the extended cell for each level.

Notably, the same set of data is to be used by the runs planned for a particular level, though different rules are to be applied.

16.4.2 Data Input of Level 1

The data input requirement of the extended cell model at this level also includes the production requirements information, the machine information, the load/unload station information, the transporter information, the pallet information and the process information (see Appendix VII.2). The station exchange time of the two load/unload stations is specified as 0 because of the fact that these stations do not have any local buffers in the designed cell.

For the process information, the first and the last operations correspond to the palletisation and depalletisation activities respectively. Each of the these two operations is assigned to the two load/unload stations. The second operation is assigned to the five machines, with the total machining time being used as the duration of this operation. A summary of the cell parameters necessary for the operation of the cell at this level is shown in Figure 16.15.

16.4.3 Data Input of Level 2

The data input of the extended cell model for level 2 consists of the production requirements information, machine data, load/unload station data, work transporter data, temporary storage data, pallet data, tool data and process information (Appendix VII.2).

The production requirements and the transporter data are identical to what is entered for level 1, but a dual-type pallet exchange buffer is specified for each of the five machines. In addition, tool exchange time and index time have to be specified. More tools need to be entered, with the tool life and the maximum percentage tool life utilization being specified. Details of tool information can be found in Appendix VI.1.

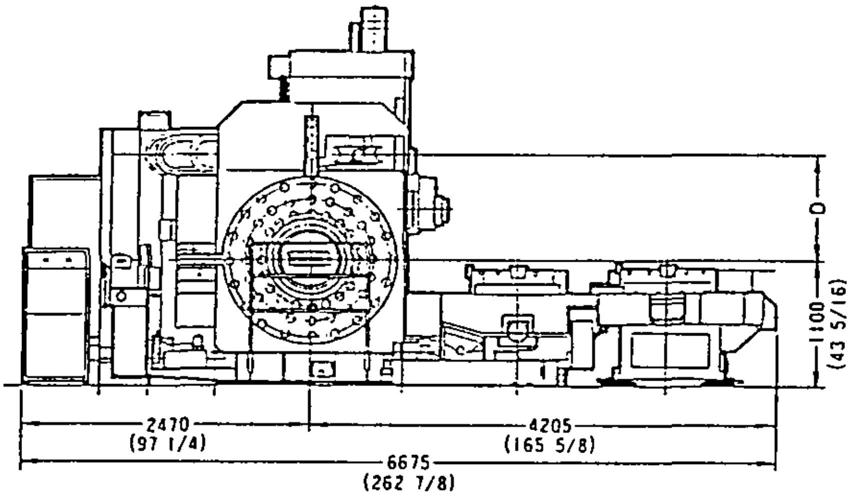
To allow maximum flexibility, each pallet is assigned with all the 18 pallet stands. Again three operations are defined for each part, with the first and the third corresponding to the palletisation and depalletisation activities respectively. For the second operation, the tooling operations are defined as a list of sub-operations. The assignment of these operations to the load/unload stations or machines is the same as that of level 1. For a summary of the major cell parameters required at this level, see Figure 16.16.

16.4.4 Data Input of Level 3

To model the extended cell at level 3, the following data is required: production requirements data, machine data, load/unload station data, work transporter data, temporary storage data, pallet data, tool data, tool transporter data, STS data and process data (see Appendix VII.2).

For each of the five machining centres, the PTS capacity is specified as 40 assuming that comprehensive tool flow could occur, which will bring the level 3 of the modelling system under a more serious test.

Three tool transporters have been specifed corresponding to the three men available in the cell. Again an average transfer time between tool stores is assumed. For a summary of the major cell parameters that are considered in the level 3 modelling, see Figure 16.17.



machine number:	1, 2, 3, 4, 5
machine desc.:	MC1210
machine id.:	MC1, MC2, MC3, MC4, MC5
prim. stores:	1
PTS capacity:	40
PTS index time:	0.5
tool exchange time:	0.26
tool complement fixted/variable:	V
PTS fixted/movable:	F
number of spindles:	1
number of type in cell:	5
machine load time:	1
machine unload time:	1
buffer type:	dual pallet exchange
buffer load time:	0.27
buffer unload time:	0.27

Figure 16.1	*Description of the MC1210 in the Extended Cell*	LUT-FMS Research Group

- 5 Makino MC1210 Machining Centres with 120 Primary Tool Store and Dual Pallet Exchange Buffer
- 1 Rail Guided Vehicle
- 10 Pallets
- 20 Part Types
- 2 Load / Unload Stations
- 18 Temporary Storage Stations
- 1 Secondary Tool Store
- 3 Men

Figure 16.2	*The Extended Cell Elements*	**LUT-FMS** Research Group

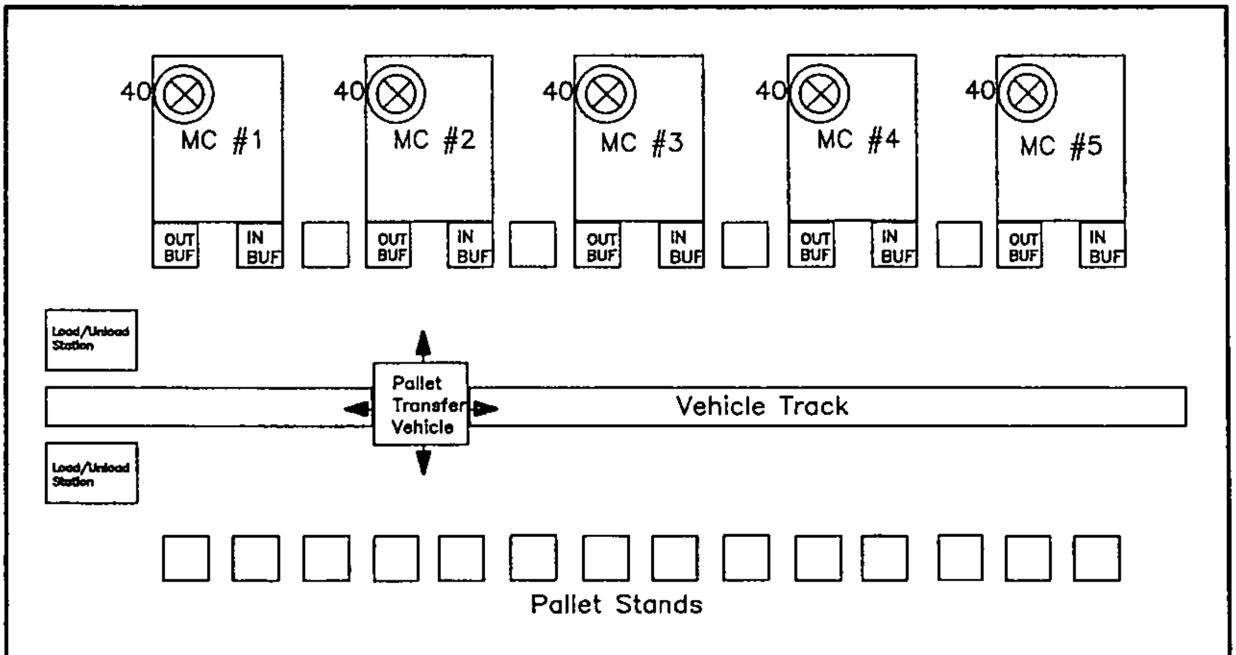


Figure 16.3	*The Extended Cell - Part Flow*	LUT - FMS **Research Group**

Part Type	Quantity/Day
1	6
2	6
3	4
4	3
5	1
6	6
7	1
8	1
9	4
10	3
11	1
12	5
13	19
14	8
15	3
16	5
17	3
18	5
19	6
20	11

10 Pallets for All Part Types

Figure 16.4 — *The Extended Cell Part/Pallet Information* — **LUT-FMS** Research Group

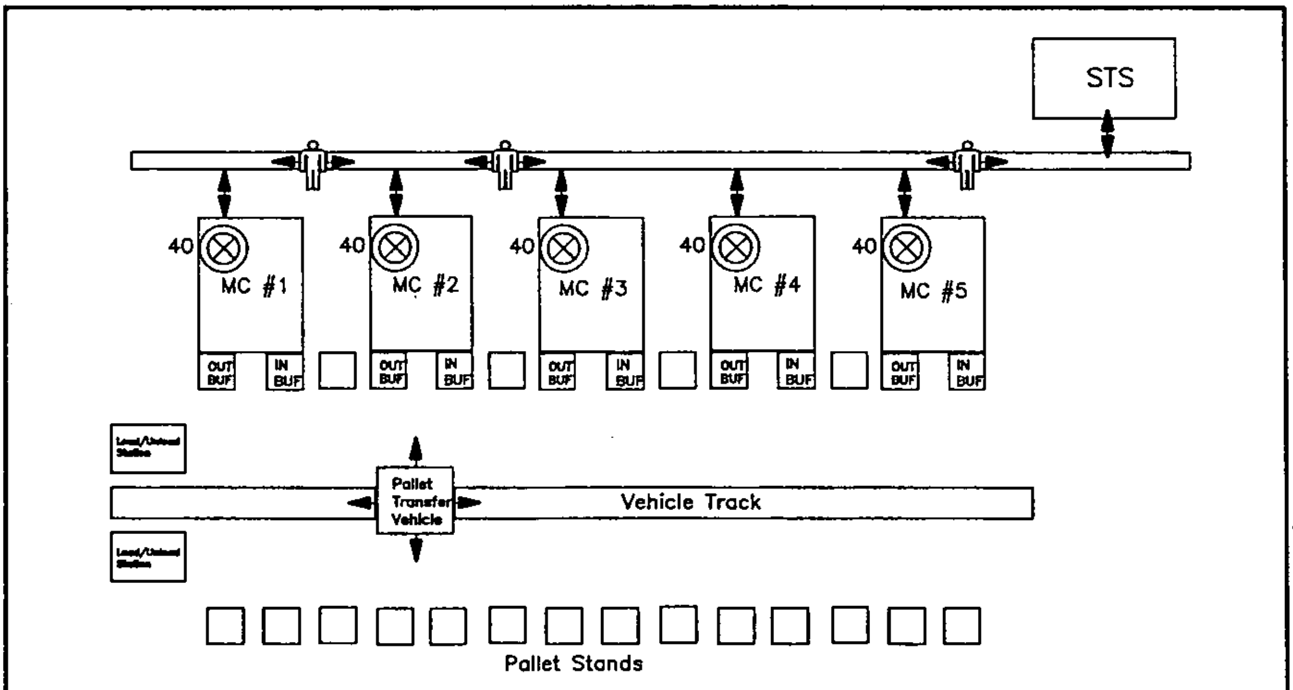


Figure 16.5 — *The Extended Cell - Part & Tool Flow* — LUT - FMS **Research Group**

Rule 1 of PalletConflictSet.

FirstInFirstOut:

```
IF palletselected—(PickLowObj self
                   'GetFinishTime
                   :Pallets)
THEN :PalletSelected—palletselected;
```

Figure 16.6	*Run 1 of the Extended Cell - Pallet Prior. Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

ShortestQueueLength:

```
IF nextstation—(PickLowObj self
                'GetQueueLength
                :NextStations)
THEN :NextStationSelected—nextstation;
```

Figure 16.7	*Run 1 of the Extended Cell - Stn. Prior. Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

EarliestAvailable:

```
IF station←(PickLowObj self
                'GetAvailableTime
                :NextStations)
THEN :NextStationSelected←station;
```

Figure 16.8	*Run 2 of the Extended Cell - Station Prio. Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

LeastWorkLoad:

```
IF station←(PickLowObj self
                'GetWorkLoad
                :NextStations)
THEN :NextStationSelected←station;
```

Figure 16.9	*Run 3 of the Extended Cell - Station Prio. Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

MostSpareSpaces:

```
IF station-(PickLowObj self
                'GetSpareSpaces
                :NextStations)
THEN :NextStationSelected-station;
```

Figure 16.10	*Run 7 of the Extended Cell - Station Prio. Rule*	**LUT-FMS Research Group**

Rule 1 of PalletConflictSet.

ShortestRemainingProcessingTime:

```
IF palletselected-(PickLowObj self
                'GetRemainingProcessingTime
                :Pallets)
THEN :PalletSelected-palletselected;
```

Figure 16.11	*Run 10 of the Extended Cell - Pallet Prio. Rule*	**LUT-FMS Research Group**

Rule 1 of ToolTransporterConflictSet.

SpecifiedPriority:

```
IF transporter-( PickHiObj    self
                 'GetPriority
                 :ToolTransporters)
THEN :ToolTransporterSelected-transporter;
```

Figure 16.12	*Run 15 of the Extended Cell - Tool Transp. Rule*	**LUT-FMS Research Group**

Rule 1 of ToolingConflictSet.

FirstInFirstOut:

```
IF pallet-(PickLowObj self
                 'GetFinishTime
                 :Pallets)
THEN :PalletSelected-pallet;
```

Figure 16.13	*Run 15 of the Extended Cell - Palt. Tooling Rule*	**LUT-FMS Research Group**

Rule 1 of NextStationConflictSet.

FewestToolsUnavailable:

```
IF nextstation-(PickLowObj self
                 'GetUnavailableTools
                 :NextStations)
THEN :NextStationSelected-nextstation;
```

Figure 16.14	*Run 17 of the Extended Cell - Stn. Prior. Rule*	**LUT-FMS Research Group**

Planning Horizon: 24 Hours (1440 mins)

FMC: - 5 machines with infinite part buffers

- 1 rail guided vehicle
- 2 load/unload stations with infinite part buffers
- 10 pallets
- 101 parts

Figure 16.15	*K. B. Modelling Level 1 of the Extended Cell - Cell Parameters*	LUT-FMS Research Group

Planning Horizon: 24 Hours (1440 mins)

FMC: - 5 machines with dual pallet exchange buffer

- 1 rail guided vehicle
- 2 load/unload stations
- 10 pallets
- 101 parts

Operation Assignment: part by part

Tool Life Management:

- permissible life 90%
- tool life as specified
- machine rationalisation

Tool Assignment: all required tool are available in each PTS

Figure 16.16	*K. B. Modelling Level 2 of the Extended Cell - Cell Parameters*	LUT-FMS Research Group

Planning Horizon: 24 Hours (1440 mins)

Tool Management: Workpiece-Oriented

Tool Issue: Differential Kitting

FMC: - 5 machines with dual pallet exchange buffer & 40-tool PTS

- 1 rail guided vehicle
- 2 load/unload stations
- 10 pallets
- 101 parts
- mutually exclusive part & tool flow
- STS of unlimited capacity
- 3 tool transporters

Operation Assignment: part by part

Tool Life Management:

- permissible life 90%
- tool life as specified
- cell rationalisation

Tool Assignment: all required tools are available in STS

Figure 16.17	*K. B. Modelling Level 3 of the Extended Cell - Cell Parameters*	LUT-FMS Research Group

Chapter 17
RESULTS OF THE KNOWLEDGE BASED MODELLING OF THE EXTENDED CELL

17.1 Introduction

In this chapter, a summary and discussion of the results obtained from the modelling of the extended cell is given. The emphasis is to assess the influenec of alternative decision rules on the performance of the cell and to identify the approriate control strategies which can be used in the operation of the cell.

17.2 Initial Comments

As mentioned in chapters 15 and 16, the purpose of this case study is to bring the modelling system un der a more substantial test. Major aspects to be assessed in clude the modelling of different decision rules and the prediction of the performance of tool flow in the cell. This is based on the results obtained from the modelling runs planned in the previous chapter.

Since each run represents the operation of the extended cell under the control of one combination of decision rules with the same set of data input, the significance of decision rules with regard to the cell performance can be expected to be adequately demonstrated through this case study.

17.3 Summary of Results Obtained

17.3.1 Initial Comments

A summary of the results follows, based on the volume of output generated from the computer modelling runs under the conditions described above. The detailed listing of these results, however, is not included in the thesis as a result of the considerable volume of the results obtained. The emphasis is placed on interpreting the results against the decision rule and data input to the modelling system. The abbreviations to be used in the figures are listed in Figure 17.1.

17.3.2 Output of Level 1

A summary of the results obtained from the level 1 modelling runs is given in Figures 17.2 to 17.4. It was found that in all six runs, the total time to complete the specified production requirements (or make span as shown in the figures) is over the planning horizon (1440 minutes). This implies that the production requirements might have been over-specified, or the cell does not have the sufficient capacity to meet these requirements.

The total part throughput rate is between 30 and 32 parts per shift, and the average part lead time is from 134.9 to 141.3 minutes. From Figure 17.2 it can be seen that the use of different station selection rules have resulted in the assignment of differing workloads to the machines, and thus the utilization of these machines varies with the selection of varying rules. With regard to the balancing of workload across machines, it can be found that Run 3 and Run 6 produced the best results because of the use of the *Least Workload* rule. The use of the *Shortest Queue Length* rule (Runs 1 and 4) has resulted in the most unbalancing of workload assignment to the five machines.

Since only one AGV was used and an average transfer time was assumed, the load running time and the load/unload time was exactly the same for all six runs. However, the empty running time varies with differnet runs. The reason for this was that the sequence of the allocation of the transporter to pallets could vary in different runs.

The utilization of the two load/unload stations under the same station selection rule was found to be similar for both batch and kit production. Again the *Least Workload* rule (Runs 3 and 6) produced the most balanced workload assignment to the two stations. The load/unload time of these two stations was found to be zero. This was because the station exchange time had been specified as zero on order to make the modelling realistic.

From Figure 17.3 it can be seen that the use of different station selection rules also caused the part lead times to be differing in different runs. The reason for this was that the selection of different stations could cause parts to spend different times on waiting for processing. The parts kit lead time for Runs 4 to 6is shown in Figure 17.4.

It is apparent that Run 6 produced the best results in terms of the kit lead time. For among the five kits, only the average lead time of kit 5 in Run 6 was sligtly longer that in Run 5, whereas the lead time of all the other kits was the shortest in Run 6.

17.3.3 Output of Level 2

The results of the modelling runs at level 2 are summarized in Figure 17.5 through to 17.8. It is apparent that the results for kit release are better than the corresponding results under batch release. This is shown explicitly with respect to the make span, the total throughput and the average part lead time.

Again the workload assignment to the five machines varied from run to run, but the *Least Workload* rule together with the *FIFO* pallet priority rule (Runs 9 and 13) generated the best results. Notably in the case of batch production, some of the machines have a stationary time which means that the pallet could not be unloaded because of the occupation of the output buffer by other pallets. This is one of the reasons that the overall performance of batch release was worse than that of kit release.

The transporter performance is very similar for all eight runs. The best balancing of the utilization of the two load/unload stations was achieved again by the *Least Workload* rule (Runs 9 and 13). From Figure 17.6 it can be seen that the part lead times varied significantly with different runs. This is also true with the parts kit lead times (Figure 17.7). From kit 1 to kit 5, the best average kit lead time was produced by Runs 13, 14, 11, 12 and 11 again respectively.

As shown in Figure 17.8, the cell minimum tool requirement was 100 tool types and 239 tools in total. The cell maximum tool requirement, however, varied with different runs. The machine PTS tool requirement for each modelling run was between 71 and 110.

17.3.4 Output of Level 3

A summary of the outputs obtained from the level 3 modelling of the extended cell is given in Figures 17.9 to 17.12. For batch production, the results were very similar for the three runs (Runs 15, 16 and 17) with regard to the overall cell performance.

However, Run 20 produced obviously the best results for kit prodcution. For the five machines, the stationary time for all runs was zero, thus it is not shown in Figure 17.9, but each machine spent considerable time on waiting for tools (between 138.0 and 270.0 minutes). The load running time and the load/unload time of the work transporter for all runs was the same, with slight variation on the empty running time. The stationary time of the two load/unload stationas was between 73.9 and 102.3 minutes.

The average part lead time for the twenty parts under different runs is shown in Figure 17.10. As for the kit lead time (Figure 17.11), the better results were obtained from Run 19. For the lead time of kitts 3, 4 and 5 was the shortest un fer this run, and that of kitts 1 and 2 was not the longest.

As shown in Figure 17.12, the cell tool requirement for kit production was smaller than that of batch production, and the use of *Fewest Tools Unavailable* rule produced the best results under both release methods. The number of worn tools was between 8 and 11. The changes of position tools on each machine were fairly considerable, with the minimum being 28 tools and the maximum 110 tools. This implies that a considerable number of tools were shared across different machines or parts. The final contents of the STS were between 169 and 297 tools under all six runs.

Since three men were used for tool load/unload and transfer, the utilization of these men was fairly low. It is also apparent that the load/unload time of a tool transporter is much longer than its tr4avelling time. This is because each tool needs 0.76 minutes to be loaded or unloaded, while the transfer time between tools stores for a whole differential kit is 0.8 minutes. Besides the selection of the *Specified Priority* tool transporter selection rule has resulted in the most use of man 1 and the least use of man 3.

17.4 Discussion of Results

17.4.1 Overview of the Discussion

Based on the above general summary of the results obtained from each level of modelling, this section concentrates on the implication of these results with respect to the structure and control strategies of the extended cell. Major cell parameters to be covered include the number of machines, machine buffer capacity, number of load/unload stations, number of work and tool transporters, machine PTS capacity, and cell tool requirement. Discussion on the cell control strategies is maiinly concerned with the relative merits of batch or kit production and the influence of the decision rules on the performance of the cell.

17.4.2 Discussion of Level-1 Results

17.4.2.1 No. of Machines

As mentioned above, the total throughput time for all six runs at level 1 was over the planning horizon. Given the production requirements, there are three possible physical limiting factors which can cause this lateness, i.e., the number of machines, transporters or load/unload stations. From the performance figures of the machines, transporters and load/unload stations (Figure 17.2) it can be concluded that since the five machines have a very high utilization (between 84.88 and 109.01% over the 1440 minutes), one more machine is needed in order to complete the production requirements within the planning horizon.

17.4.2.2 Machine Local Buffer Capacity

As shown in Figure 17.2, the maximum queue length of the five machines is between 2 and 4. Therefore a dual-type pallet exchange buffer, or a two or four position rotational buffer will minimise the stationary time of the machines, i.e. the time that pallets spend on waiting at the machines in order to be unloaded into the local buffers of the machines. This conclusion can be verified again through the results of the level 2 and level 3 modelling in the following sections.

17.4.2.3 Number of Load/Unload Stations

Since the utilization of the load/unload stations was not considerably high (bwtween 55.56 and 83.33%), the use of two stations in the cell should be considered as a feasible solution. The maximum queue length of the stations was 4 or 5. This does not mean that the two load/unload stations need a four or five position local buffer. For from the station operation schedules it can be found that this long queue only occured at the start of the modelling as a result of the specification of zero minutes for the palletisation process of each part. Again this can also be shown through the results of levels 2 and 3.

17.4.2.4 No. of Transporters

From the performance of the AGV (Figure 17.2) it can be concluded that although only one AGV is used in the cell to perform all the work transfer activities, there was very little delay caused by the situation where pallets spent a lot of time on queueing for a transporter. The reason for this is that the average transfer time between stations (0.8 minutes) is far shorter than the machining times of the components.

17.4.2.5 Number of Pallets

It is evident that both too many and too few pallets in the cell can cause decrease in the performance of the cell. Thus there is a point where the number of pallets to be used in the cell can bring the best performance.

Experiments using the level 1 with varying number of pallets in the cell have been carried out. Description of these experiments and the discussion of the results obtained can be found in Appendix VII.3.

17.4.2.6 Decision Rules

As shown in Figure 17.16, under batch production environment the best results are obtained with regard to the make span, the total throughput rate and the average part lead time. However, in Run 3 the workload is mostly balanced across the five machines. In the case of Run 1, the workload has been made mostly unbalanced. Thus, if the performance criterion is to achieve the shortest make span and part lead times, and the highest throughput rate at the same time, then the Earliest Available station selection rule should be selected. However if the workload assigned to the five machines is intended to be adequately balanced, the *Least Workload* rule should be used. In some cases, all the above criteria may have to be considered [100], and therefore either of these two rules can be used depending on the weight placed on each criterion.

In the case of kit production (Figure 17.17), the use of the *Least Workload* station selection rule (Run 6) provided the best results with regard to all five performance criteria. Thus there is no doubt that this rule should be applied.

17.4.3 Discussion of Level-2 Results

17.4.3.1 Machine Utilization

As shown in Figure 17.5, the machine utilization at level 2 was also very high. This was because ten pallets were employed in the cell to provide the five machines with workpieces. Therefore while a machine was processing a component, there was quite often another one waiting in the input buffer, causing very short machine idle time.

However, since the make spans for all eight runs exceeded the planning horizon (1440 minutes), there is a significant demand for another machine so as to complete the production requirements within the planning horizon. This conclusion is consistent with the conclusion made in the discussion of the level 1 results.

17.4.3.2 Temporary Storages

As mentioned in the previous chapter, temporary storages were not considered in the experiments presented in this chapter. For there were only ten pallets used in the cell, and the five machines and the two load/unload stations can provide 12 positions. As a result, no blockage should occur even if the 18 pallet stands were not used.

To assess the influence of these pallet stands on the performance of work flow, experiments which consider the use of temporary storages can be planned. For a detailed discussion of the modelling of temporary storages as an example, refer back to the study of the three machine cell.

17.4.3.3 Load/Unload Stations

Since the utilization of the load/unload stations was fairly low, two stations are therefore adequate for the palletisation and depalletisation activities over the planning horizon.

17.4.3.4 Work Transporter

Similar to the level 1 modelling, the utilization of the AGV at this level was also very low because of the short transfer time assumed in comparison with the long processing times of the components. For all the eight runs, the utilization of the AGV was between 28.19 and 29.47%. This far lower than the utilization of the five machines (typically around 98.29%).

17.4.3.5 Tool Management

As shown in Figure 17.8, the minimum cell tool requirement was 239 and the maximum was below 550. Thus if a magazine of less than 48-tool capacity is to be used on each machine (which means the total number of tools that can be accommodated by the PTSs is 240), then a tool flow management system must be developed no matter it is automated or mannual. For there must be some tools provided in addition to the tools in the PTSs so as to complete the production requirements.

On the other extreme, if the capacity of the magazine to be used is larger than 110 (such as 120) and the required tools are initially assigned to the specific magazines at the start of the production program, then there is no need for an additional tool provision system because the machine tool magazines can supply enough tools for the machining of the specified part spectrum.

17.4.3.6 Decision Rules

As shown in Figure 17.18, under batch production, the overall cell performance is very similar for all the four runs, but the machine workload balancing across the five machines is significantly different. Run 1 produced the most unbalancing among the four runs, and Run 3 the most balancing. This is because in Run 3 the *Least Workload* station selection rule was used in conjunction with the *FIFO* pallet priority rule.

It is apparent that the use of the *Shortest Remaining Processing Time* rule for pallet selection together with the *Least Workload* rule for station selection did not produce the better results, in stead the workload assignment balancing was disturbed by this pallet priority rule. Therefore, under batch production environmentthe *Least Workload* station selection rule and the *FIFO* pallet priority rule should be applied.

In the case of kit production (Figure 17.19), it is obvious that Run 13 produced the best results with regard to all the overall performance criteria. Again, in Run 14 the use of the *Shortest Remaining Processing Time* rule did not produce the better results than Run 13 which used the *FIFO* pallet priority rule. The worst results were produced by Run 11 which employed the *Most Spare Spaces* rule and the *FIFO* rule. As a result, the *FIFO* pallet priority rule and the *Least Workload* station selection rule should be selected.

17.4.4 Discussion of Level-3 Results

17.4.4.1 Machine Utilization

Since the provision of tools reduces the work flow rate in the cell, the utilization of the five machines was lower than that obtained from the other two levels. However, if the tool wait time of a machine is also included in its busy time, then the utilization of a machine at this level was between 99.46 and 118.29%. Therefore if the production requirements are to be completed within the production horizon, there is definitely a need for at least one more machine to be added to the cell. Refering back to the conclusions made for the other two levels, the addition of more machines at this level is more critical because of the provision of tools to the machines.

17.4.4.2 Load/Unload Stations and Work Transporters

As shown in Figure 17.9, the performance of the two load/unload stations and the AGV at this level is very similar to that of the other two levels. Thus these are not citical elements in this cell.

17.4.4.3 Tool Transporters

From Figure 17.12 it can be seen that the utilization of the three men is between 5.78 and 67.85%. Besides, among the three men man 1 was always utilized the most and man 3 the least for all six runs. The reason for this was that the *Specified Priority* rule was used for the selection of a man for the tool transfer. Thus it is apparent that the number of men required can be cut down to two, without significantly affecting the performance of the cell, if the manning level is to be reduced.

17.4.4.4 Machine PTS Capacity

As shown in Figure 17.12 the changes of position tools for a machine is between 28 and 110. Thus if 20 parts were assigned to a machine, the number of tool changes because of the change of the component is between 1.4 and 5.5 on the average. This implies that the demand for tool transporters is fairly frequent, though the utilization of the tool transporters is not high.

Therefore it can be concluded that the specification of the machine PTS capacity as 40 did cause comprehensive tool flow between tool stores and forty is an appropriate magazine capacity in the case of the extended cell when a tool provision system is available for the transfer of tools between tool stores.

17.4.4.5 Cell Tool Requirements

From Figure 17.12 it can be found that the perceived cell tool requirement varies with different runs. It ranges from 369 tools to 497 tools. Therefore, depending on the selection of decision rules, the cell should be provided with a different set of tools. The sister tooling prediction for each of the runs is shown in Figure 17.13. As can be expected the sister tools required for a particular tool type can also differ significantly for different runs. This is mainly because of that in the modelling, tool requirements are generated according to the workpieces scheduled to the machines, but the work flow patterns were seriously influenced by the decision rules selected. As a result, different tool requirements can be generated by the runs under different decision rules.

17.4.4.6 Tool Usage Frequency and Tool Life Management

A sub-set of eight tools are selected and shown in Figures 17.14 and 17.15 with regard to their usage frequency and life utilization. Each of the eight tool numbers under the six runs correspond to the same tool type, leading to the fact that each tool number for the different runs can be considered as the same tool.

From Figure 17.14 it can be seen that these eight tools were used the same times under batch production (Runs 15, 16 and 17). Under kit production (Runs 18, 19 and 20), however, the difference in tool usage frequency is considerable. This implies that the use of different station selection rules had a significant influence on the use of these tools in kit production but not in batch production. Similarly the life utilization of these eight tools was the same for Runs 15, 16 and 17, but considerably different in the cases of Runs 18, 19 and 20 (Figure 17.15).

In addition, since most of the tools were under-utilized subject to the specified 90% maximum percentage tool life utilization value, the tool inventory can be further reduced. This can be achieved by using a smaller tool magazine so that tools can be used more often within the production program.

17.4.4.7 Decision Rules

As shown in Figure 17.20, the overall cell performance is very similar for the three runs under batch production condition, but the machine workload balancing is fairly different. In Run 15 the workload is mostly balanced, where the *Most Spare Spaces* station selection rule was used. From Figure 17.21 it can be found that the cell tool requirements and the total position tool changes of Runs 15 and 17 are smaller than those of Run 16. Therefore, the *Most Spare Spaces* rule should be selected for batch production at level 3. The reason that the *Fewest Tools Unavailable* rule (Run 17) did not perform better than the *Most Spare Spaces* rule (Run 15) can be that the former rule did not affect the work flow much under batch release environment. For when there are several machines available, the tools available in each PTS can be very similar if the workpieces introduced to the machines so far are from the same batch.

In the case of kit production (Figure 17.22), the *Fewest Tools Unavailable* rule for station selection (Run 20) produced the best results with regard to the make span, the total throughput rate, the average part lead times and the machine workload balancing though the average kit lead time of this run is 6.43 minutes longer than that of Run 19. What's more, the cell tool requirement and total position tool changes are the smallest in Run 20 (Figure 17.23). Thus, so long as the most important performance criterion is not the kit lead time, the *Fewest Tools Unavailable* rule should be used for kit production.

One of the major reasons that this rule so greatly influenced the cell performance should be that under kit production, the tools present in a PTS can be significantly different from those in the other PTSs. Therefore the use of this rule did reduce significantly the tool changes and tool provisions due to the change of components. Meanwhile, because of this rule, the tools available in the PTSs can be more frequently used, leading to the considerable reduction in the cell tool requirement. Besides, fewer tool changes can reduce the time that the machine spend on waiting for tools, and therefore the overall cell performance is improved.

17.5 Concluding Remarks

17.5.1 Scope of the Conclusions

From the above discussion it can be concluded that depending on the entering of different combinations of decision rules, the performance of the modelled cell can be significantly differing. In the following, the relative merits of the alternative decision rules are discussed. In particular, the entering of other rules with regard to the operation of the extended cell is indicated, with some examples illustrated.

17.5.2 Part Release Rules

It is obvious that the results of batch production are greatly different from those of the kit production. Therefore the part release rule plays a very important role in the management of work flow. Broadly speaking, the performance of kit release is better than that of batch release. Therefore to further improve the performance of the cell, rules shold be developed which enable the release of different components in sequence.

17.5.3 Station Selection Rules

The station selection rule also affects the performance of the cell significantly. Obviously, this is not true if a fixed route is defined for a component. However once alternative stations are assigned to the operations of the components, this rule dynamically selects a station according to certain criteria if several are available. It is apparent that this rule not only dictates the flow pattrens of workpieces but also influences the workload assignment to the stations. As mentioned before, this rule is applied after the station availability rule is used (see Chapter 8). Therefore, relying on the representation of the conditions under which the stations can be considered available, the station selection rules can be either simple or complex.

In the case of pure work flow, the *Least Workload* rule had proven to be broadly better than the others. However, if both work and tool flow was considered in the modelling, the *Most Spare Spaces* rule performed well when the release of components was based on batches. When the components released were mixed, the *Fewest Tools Unavailable* rule was the best.

17.5.4 Pallet Priority Rules

Although only one alternative pallet priority rule, the *Shortest Remaining Processing Time* rule, was tested in the case study, it can be concluded that this rule also affects the performance of the cell. In the case of the extended cell, the *FIFO* rule performed better than the *Shortest Remaining Processing Time* rule. Other rules, such as the *Longest Remaining Processing Time* rule and the *Shortest Processing Time* rule [257] can be conveniently developed.

17.5.5 Pallet Tooling Rules

In this study, only one pallet tooling rule was developed and tested, i.e. the *FIFO* rule. However, other rules, such as the *Fewest Tools Required* rule, can be easily expressed and brought under test. The *Fewest Tools Required* rule selects a pallet which enables the shortest use of a tool transporter because of the short total tool load/unload time involved. This rule can especially be useful when the demand for tool transporters is critical.

17.5.6 Tool Transporter Selection Rules

In the above presented case study, again, only one tool transporter selection rule, the *Specified Priority* rule, was tested. To balance the utilization of the tool transporters, the *Shortest Cummulative Transport Time* rule can be developed. Similarly, to minimise the travelling time, the *Nearest Transporter* rule can be developed, which selects the tool transporter that is the closest to the STS.

MAC. – Machining Time

LUD. – Load/Unload Time

MQU. – Maximum Queue Length

LRU. – Load Running Time

ERU. – Empty Running Time

FIX. – Fixturing Time

CUT. – Cutting Time

TCH. – Tool Change Time

STA. – Stationary Time

TYP. – Tool Types

TOL. – Tools

FTL. – Tool Waiting Time

FCO. – Final Contents

CWT. – Changes of Worn Tools

CPT. – Changes of Position Tools

ICO. – Initial Contents

Figure 17.1	*Abbreviations Used in Following Figures*	LUT-FMS Research Group

Production Methods			Batch Production			Kit Production		
Run No.			1	2	3	4	5	6
Make Span (mins)			1564.5	1518.8	1541.6	1594.0	1534.5	1513.5
Total Throughput (part/shift)			30	31	31	30	31	32
Average Part Lead Time(mins)			139.7	134.9	137.8	141.3	136.4	137.2
Mach. Perform.	MC1	MAC.	1483.4	1408.4	1399.6	1521.7	1418.7	1347.4
		LUD.	46.0	40.0	40.0	48.0	44.0	44.0
		MQU.	2	2	3	2	3	3
	MC2	MAC.	1313.3	1347.1	1384.1	1186.3	1297.4	1349.5
		LUD.	36.0	38.0	42.0	36.0	36.0	38.0
		MQU.	2	2	3	2	2	3
	MC3	MAC.	1318.8	1341.0	1367.3	1357.8	1404.7	1382.6
		LUD.	36.0	44.0	38.0	40.0	38.0	38.0
		MQU.	2	3	3	2	3	3
	MC4	MAC.	1357.0	1428.1	1379.4	1448.2	1356.5	1412.4
		LUD.	42.0	40.0	42.0	38.0	40.0	42.0
		MQU.	2	3	3	2	3	3
	MC5	MAC.	1462.1	1409.9	1404.1	1420.6	1457.2	1442.6
		LUD.	42.0	40.0	40.0	40.0	44.0	40.0
		MQU.	2	3	3	2	4	4
Transp. Perform.	AGV	LRU.	161.6	161.6	161.6	161.6	161.6	161.6
		ERU.	142.4	132.8	142.4	143.2	141.6	139.2
		LUD.	109.1	109.1	109.1	109.1	109.1	109.1
Load/ Unload Perform.	LU1	FIX.	1220.0	1100.0	1020.0	1200.0	1120.0	1020.0
		LUD.	0.0	0.0	0.0	0.0	0.0	0.0
		MQU.	4	4	4	4	4	4
	LU2	FIX.	800.0	920.0	1000.0	820.0	900.0	1000.0
		LUD.	0.0	0.0	0.0	0.0	0.0	0.0
		MQU.	5	5	5	5	5	5

Figure 17.2	*K.B.Modelling Level 1 of the Extended Cell - Summary of Outputs*	LUT-FMS Research Group

Production Methods		Batch Production			Kit Production		
Run No.		1	2	3	4	5	6
Average Part Lead Time (mins)	p1	99.55	99.55	105.68	137.48	94.84	100.66
	p2	134.21	131.71	135.56	135.64	109.31	117.46
	p3	92.26	90.38	88.31	152.57	132.14	120.25
	p4	91.31	80.12	103.92	131.73	106.69	140.11
	p5	141.21	131.77	130.49	130.49	139.37	148.20
	p6	180.71	169.38	179.20	161.24	170.07	181.36
	p7	183.13	275.67	162.52	199.40	137.86	175.92
	p8	223.09	215.80	218.72	203.05	220.19	203.82
	p9	155.89	136.21	175.04	150.14	148.94	148.21
	p10	140.18	111.45	118.45	152.84	142.10	87.99
	p11	148.21	162.50	178.30	200.03	189.08	193.59
	p12	194.17	156.73	158.60	175.91	166.64	171.25
	p13	218.19	225.03	219.91	172.71	179.34	179.44
	p14	125.44	108.61	122.40	84.66	99.82	126.61
	p15	90.02	95.45	84.75	94.16	129.49	98.65
	p16	120.40	102.42	110.49	130.35	111.66	134.92
	p17	88.00	99.57	94.62	120.38	120.51	80.56
	p18	116.82	97.76	94.98	134.26	124.64	108.82
	p19	88.57	93.09	95.59	110.85	121.72	101.08
	p20	79.22	80.00	80.87	126.59	115.87	121.39

Figure 17.3	*K.B.Modelling Level 1 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Method			Kit Production		
Run No.			4	5	6
Parts Kit Lead Time (mins)	kit #1	Average	232.31	237.81	204.64
		Maximum	304.32	292.28	229.60
		Minimum	163.36	205.38	149.72
		Mean	233.84	248.83	189.66
	kit #2	Average	259.40	272.17	256.60
		Maximum	259.40	272.17	256.60
		Minimum	259.40	272.17	256.60
		Mean	259.40	272.17	256.60
	kit #3	Average	227.57	215.36	211.83
		Maximum	247.46	224.88	220.40
		Minimum	208.78	205.19	204.39
		Mean	228.12	215.04	212.40
	kit #4	Average	238.39	235.87	232.99
		Maximum	267.49	297.55	252.15
		Minimum	213.67	212.15	214.85
		Mean	240.58	254.85	233.50
	kit #5	Average	231.69	204.22	220.36
		Maximum	254.38	234.69	250.79
		Minimum	209.79	164.39	165.44
		Mean	232.09	199.54	208.12

Figure 17.4	*K.B.Modelling Level 1 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Methods			Batch Production				Kit Production			
Run No.			7	8	9	10	11	12	13	14
Make Span (mins)			1599.5	1603.5	1596.0	1599.5	1513.0	1498.9	1491.0	1492.5
Total Throughput (part/shift)			30	30	30	30	32	32	32	32
Average Part Lead Time(mins)			152.6	153.5	152.3	152.8	143.0	140.6	139.7	140.7
Mach. Perform.	MC1	CUT.	1292.1	1317.8	1213.7	1299.4	1228.9	1242.3	1210.0	1251.0
		TCH.	164.9	164.9	149.7	166.4	171.8	155.8	149.0	168.7
		LUD.	42.0	42.0	38.0	42.0	40.0	44.0	40.0	44.0
		STA.	9.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MC2	CUT.	1255.0	1200.2	1236.6	1247.6	1216.5	1245.0	1235.7	1228.6
		TCH.	160.4	157.3	152.0	158.8	152.0	140.6	149.7	133.0
		LUD.	40.0	38.0	42.0	40.0	40.0	36.0	38.0	34.0
		STA.	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MC3	CUT.	1175.6	1020.8	1253.2	1214.2	1270.7	1212.8	1232.5	1259.3
		TCH.	145.9	120.1	150.5	149.0	153.5	158.8	141.4	160.4
		LUD.	40.0	34.0	40.0	42.0	42.0	38.0	40.0	40.0
		STA.	5.6	0.0	0.0	11.1	0.0	0.0	0.0	0.0
	MC4	CUT.	1249.5	1316.7	1231.2	1210.8	1255.5	1202.7	1241.6	1189.1
		TCH.	145.2	158.1	153.5	142.1	151.2	148.2	154.3	154.3
		LUD.	40.0	46.0	42.0	38.0	40.0	42.0	42.0	44.0
		STA.	7.0	5.1	0.0	7.0	0.0	0.0	0.0	0.0
	MC5	CUT.	1204.0	1320.6	1241.3	1204.0	1204.4	1273.3	1256.3	1247.9
		TCH.	142.1	158.1	152.8	142.1	130.0	155.0	164.2	142.1
		LUD.	40.0	42.0	40.0	40.0	40.0	42.0	42.0	40.0
		STA.	0.0	0.0	6.3	6.2	0.0	0.0	0.0	0.0
Transp. Perform.	AGV	LRU.	161.6	161.6	161.6	161.6	161.6	161.6	161.6	161.6
		ERU	145.6	142.4	152.0	152.0	135.2	148.0	148.0	153.6
		LUD.	109.1	109.1	109.1	109.1	109.1	109.1	109.1	109.1
Load/ Unload Perform.	LU1	FIX.	1100.0	1080.0	1020.0	1100.0	1060.0	1060.0	1020.0	1020.0
		LUD.	29.7	29.2	27.5	29.7	28.4	28.6	27.5	28.6
		STA.	52.8	56.0	53.6	52.8	56.5	55.2	58.1	53.5
	LU2	FIX.	920.0	940.0	1000.0	920.0	960.0	960.0	1000.0	1000.0
		LUD.	24.8	25.4	27.0	24.8	26.2	25.9	27.0	25.9
		STA.	56.9	57.9	56.3	56.1	67.3	65.2	59.8	57.5

Figure 17.5

K.B.Modelling Level 2 of the Extended Cell - Summary of Outputs

LUT-FMS Research Group

Production Methods		Batch Production				Kit Production			
Run No.		7	8	9	10	11	12	13	14
Average Part Lead Time (mins)	p1	99.84	99.97	99.84	99.84	153.02	117.35	109.11	109.56
	p2	140.85	144.72	141.12	141.12	150.11	122.39	122.59	124.57
	p3	126.87	127.74	126.00	121.20	149.36	113.50	137.96	144.03
	p4	112.77	104.15	115.00	121.40	132.38	130.39	120.31	149.85
	p5	167.82	162.26	131.29	167.02	136.48	151.26	152.06	187.16
	p6	181.33	210.92	180.84	180.04	166.33	189.72	180.66	192.86
	p7	162.52	192.05	139.04	163.32	97.69	153.03	153.03	148.44
	p8	211.28	131.88	206.48	215.82	147.76	205.87	205.87	191.61
	p9	163.82	191.44	157.60	158.55	127.37	142.13	122.69	115.95
	p10	115.66	121.10	137.17	124.29	140.32	130.74	142.98	124.12
	p11	167.89	156.26	166.02	167.89	250.20	211.67	211.67	197.17
	p12	166.49	156.00	168.50	166.49	181.82	170.70	158.76	175.17
	p13	227.57	224.44	226.42	225.62	185.14	187.27	189.79	184.05
	p14	152.21	152.93	153.49	158.16	125.24	99.51	107.69	102.99
	p15	120.87	118.73	116.98	121.40	80.95	122.51	111.89	123.30
	p16	133.47	119.29	123.73	120.76	130.05	119.88	127.17	122.37
	p17	108.31	116.90	117.25	124.43	98.29	127.11	128.17	103.80
	p18	125.61	124.58	121.40	117.82	127.35	115.77	114.40	137.43
	p19	116.95	117.21	121.40	121.00	106.34	98.98	101.78	109.60
	p20	115.30	115.84	117.51	117.36	107.26	124.35	116.73	111.41

Figure 17.6	*K.B.Modelling Level 2 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Method			Kit Production			
Run No.			11	12	13	14
Parts Kit Lead Time (mins)	kit #1	Average	238.95	222.83	210.81	226.51
		Maximum	280.96	263.91	233.84	247.89
		Minimum	139.93	177.23	177.23	216.55
		Mean	210.45	220.57	205.54	232.22
	kit #2	Average	253.68	253.63	253.63	251.26
		Maximum	253.68	253.63	253.63	251.26
		Minimum	253.68	253.63	253.63	251.26
		Mean	253.68	253.63	253.63	251.26
	kit #3	Average	183.10	233.95	216.21	202.94
		Maximum	200.88	265.70	259.76	214.15
		Minimum	168.53	192.98	197.00	190.21
		Mean	184.71	229.34	228.38	202.18
	kit #4	Average	242.99	199.53	209.93	221.25
		Maximum	278.41	222.41	234.64	271.12
		Minimum	213.66	181.19	196.51	196.67
		Mean	246.04	201.80	215.58	233.90
	kit #5	Average	211.95	218.91	218.62	221.75
		Maximum	229.6	238.59	244.39	238.14
		Minimum	202.61	198.64	202.25	199.29
		Mean	216.11	218.62	223.32	218.72

Figure 17.7	*K.B.Modelling Level 2 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Methods			Batch Production				Kit Production			
Run No.			7	8	9	10	11	12	13	14
Cell Minimum Tool Requirement		TYP.	100	100	100	100	100	100	100	100
		TOL.	239	239	239	239	239	239	239	239
Cell Maximum Tool Requirement		TYP.	100	100	100	100	100	100	100	100
		TOL.	500	478	500	506	470	458	459	443
Machine PTS Tool Require.	MC1	TYP.	76	77	88	82	58	86	74	75
		TOL.	93	95	102	99	80	98	95	91
	MC2	TYP.	81	83	81	81	81	76	71	57
		TOL.	98	99	97	98	99	99	87	80
	MC3	TYP.	95	59	80	95	78	75	79	76
		TOL.	103	71	94	104	92	88	92	93
	MC4	TYP.	91	94	93	.91	86	73	79	76
		TOL.	106	110	105	105	101	87	94	91
	MC5	TYP.	83	84	85	83	81	62	72	72
		TOL.	100	103	102	100	98	86	91	88

Figure 17.8	*K.B.Modelling Level 2 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS **Research Group**

Production Methods			Batch Production			Kit Production		
Run No.			15	16	17	18	19	20
Make Span (mins)			1735.4	1735.7	1742.9	1733.3	1764.6	1663.5
Total Throughput (part/shift)			27	27	27	27	27	29
Average Part Lead Time(mins)			165.5	165.9	166.1	161.4	161.9	157.1
Mach. Perform.	MC1	CUT.	1314.3	1255.1	1314.3	1289.9	1291.9	1212.7
		TCH.	165.7	155.0	165.7	154.3	156.6	151.2
		LUD.	44.0	42.0	44.0	40.0	38.0	42.0
		FTL.	179.4	197.7	179.4	216.1	155.2	198.6
	MC2	CUT.	1267.9	1127.9	1311.7	1321.7	1219.4	1254.8
		TCH.	157.3	144.4	163.4	160.4	139.1	146.7
		LUD.	42.0	34.0	42.0	38.0	40.0	44.0
		FTL.	179.4	157.3	181.0	186.8	196.2	184.5
	MC3	CUT.	1240.2	1247.6	1276.8	1201.3	1163.5	1221.3
		TCH.	145.2	146.7	152.8	148.2	150.5	155.8
		LUD.	40.0	40.0	44.0	40.0	42.0	46.0
		FTL.	217.7	202.3	185.5	200.7	250.3	201.8
	MC4	CUT.	1188.8	1289.4	1201.8	1182.4	1232.4	1206.9
		TCH.	146.7	161.1	142.9	151.2	151.2	160.4
		LUD.	40.0	42.0	38.0	40.0	40.0	38.0
		FTL.	208.7	193.5	219.0	248.5	270.0	160.3
	MC5	CUT.	1164.8	1256.1	1071.5	1180.7	1268.9	1280.4
		TCH.	143.6	151.2	133.8	144.4	161.1	144.4
		LUD.	36.0	44.0	34.0	44.0	42.0	32.0
		FTL.	182.3	226.5	192.9	232.9	259.7	138.0
Transp. Perform.	AGV	LRU.	161.6	161.6	161.6	161.6	161.6	161.6
		ERU	147.2	155.2	149.6	152.0	157.6	150.4
		LUD.	109.1	109.1	109.1	109.1	109.1	109.1
Load/ Unload Perform.	LU1	FIX.	1100.0	1080.0	1100.0	1080.0	1180.0	1100.0
		LUD.	29.7	29.2	29.7	28.9	31.3	29.7
		STA.	92.0	85.1	88.0	75.4	77.3	73.9
	LU2	FIX.	920.0	940.0	920.0	940.0	840.0	920.0
		LUD.	24.8	25.4	24.8	25.7	23.2	24.8
		STA.	88.1	102.3	90.0	76.4	82.6	78.9

Figure 17.9	*K.B.Modelling Level 3 of the Extended Cell - Summary of Outputs*	LUT-FMS Research Group

Production Methods		Batch Production			Kit Production		
Run No.		15	16	17	18	19	20
Average Part Lead Time (mins)	p1	121.33	121.33	121.33	125.23	134.19	122.68
	p2	148.02	148.29	148.15	140.88	158.62	148.72
	p3	128.34	126.40	129.31	139.61	194.31	173.43
	p4	115.89	114.73	117.82	162.05	128.22	115.00
	p5	203.10	174.12	203.10	182.17	192.11	195.18
	p6	220.02	213.08	220.24	193.09	200.09	182.92
	p7	195.76	194.97	194.34	105.42	192.07	200.01
	p8	257.20	251.78	279.09	268.51	262.32	238.27
	p9	178.95	192.82	180.48	155.97	185.35	141.53
	p10	150.89	139.59	137.00	143.54	184.01	176.34
	p11	177.06	190.83	177.78	196.34	196.70	209.35
	p12	201.79	197.21	200.61	202.39	183.64	219.95
	p13	243.39	240.96	243.39	206.47	196.85	211.36
	p14	159.27	164.95	161.18	128.94	134.44	104.55
	p15	119.27	120.51	120.07	168.57	170.64	162.95
	p16	131.69	135.97	133.53	150.20	147.21	137.46
	p17	127.50	147.07	116.10	120.98	117.84	119.78
	p18	115.35	124.44	136.30	142.53	132.49	113.45
	p19	123.17	122.37	121.05	148.71	135.54	139.36
	p20	117.52	116.85	116.48	137.07	116.75	116.02

Figure 17.10	*K.B.Modelling Level 3 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Method			Kit Production		
Run No.			18	19	20
Parts Kit Lead Time (mins)	kit #1	Average	249.31	253.06	254.09
		Maximum	276.69	302.24	285.32
		Minimum	177.81	157.81	177.81
		Mean	227.25	230.03	231.57
	kit #2	Average	339.57	329.20	305.05
		Maximum	339.57	329.30	305.05
		Minimum	339.57	329.30	305.05
		Mean	339.57	329.30	305.05
	kit #3	Average	252.79	225.93	242.30
		Maximum	273.86	244.57	264.04
		Minimum	240.89	208.33	203.24
		Mean	257.38	226.45	233.64
	kit #4	Average	259.99	255.23	277.40
		Maximum	315.93	292.35	304.27
		Minimum	190.04	227.83	231.77
		Mean	252.99	260.09	268.02
	kit #5	Average	256.51	245.92	262.64
		Maximum	277.10	266.88	277.35
		Minimum	246.04	224.51	254.10
		Mean	261.57	245.70	265.74

Figure 17.11	*K.B.Modelling Level 3 of the Extended Cell - Summary of Outputs (continued)*	LUT-FMS Research Group

Production Methods			Batch Production			Kit Production		
Run No.			15	16	17	18	19	20
Cell Tool Requirement		TYP.	100	100	100	100	100	100
		TOL.	486	497	482	381	384	369
No. of Worn Tools			8	8	8	9	11	8
Machine PTS Perform.	MC1	LUD.	81.3	88.2	81.3	97.3	68.4	89.7
		FCO.	40	40	40	40	40	40
		CWT.	1	2	1	4	4	2
		CPT.	66	74	66	84	46	76
	MC2	LUD.	81.3	71.4	81.3	83.6	88.9	80.6
		FCO.	40	40	40	40	40	40
		CWT.	2	1	1	2	2	0
		CPT.	65	53	66	68	75	66
	MC3	LUD.	88.2	92.0	84.4	92.0	114.8	89.7
		FCO.	40	40	40	40	40	40
		CWT.	2	2	2	1	1	1
		CPT.	74	79	69	80	110	77
	MC4	LUD.	87.4	79.8	88.2	106.4	115.5	63.1
		FCO.	40	40	40	40	40	40
		CWT.	2	1	2	2	2	3
		CPT.	73	64	74	98	110	40
	MC5	LUD.	76.8	92.7	75.2	95.0	112.5	53.2
		FCO.	40	40	40	40	40	40
		CWT.	1	2	2	0	2	2
		CPT.	60	80	57	85	106	28
Cell STS Perform.	STS	LUD.	677.9	696.2	668.8	796.5	848.2	600.4
		ICO.	0	0	0	0	0	0
		FCO.	286	297	282	181	184	169
Tool Transp. Perform.	MAN1	LRU.	77.6	60.8	80.0	65.6	83.2	77.6
		ERU.	9.6	8.8	10.4	12.0	10.4	16.0
		LUD.	640.7	530.5	652.1	706.0	800.3	580.6
	MAN2	LRU.	32.8	44.0	32.0	49.6	50.4	40.0
		ERU.	4.8	6.4	4.0	5.6	3.2	5.6
		LUD.	294.9	394.4	317.7	452.2	430.9	321.5
	MAN3	LRU.	11.2	21.6	8.0	17.6	17.6	6.4
		ERU.	2.4	1.6	2.4	4.0	4.0	2.4
		LUD.	157.3	195.3	109.4	112.5	117.0	74.5

Figure 17.12

K.B.Modelling Level 3 of the Extended Cell - Summary of Outputs (continued)

LUT-FMS Research Group

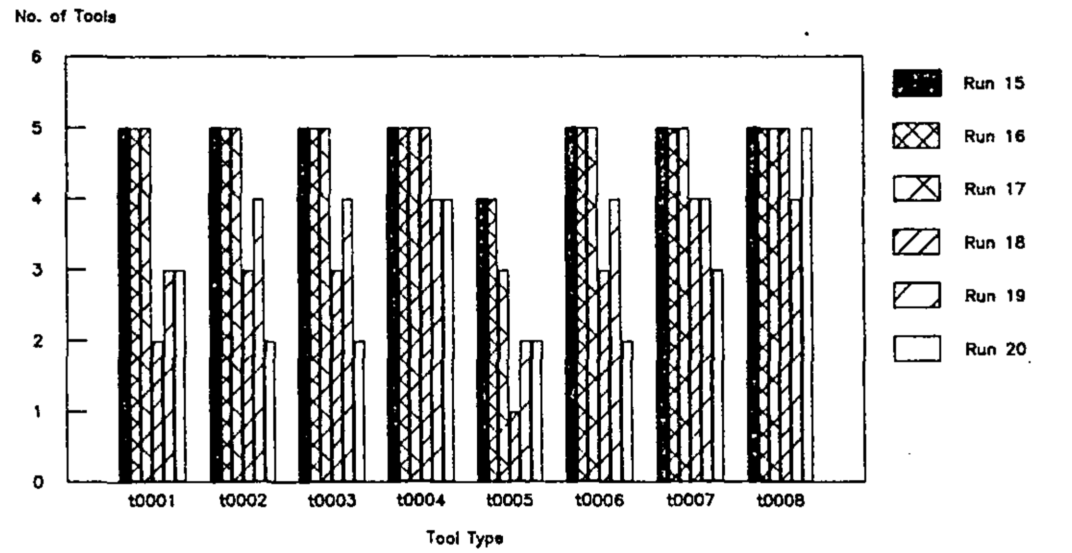


Figure 17.13	*K.B.Modelling Level 3 of the Extended Cell - Sister Tooling Prediction*	LUT - FMS Research Group

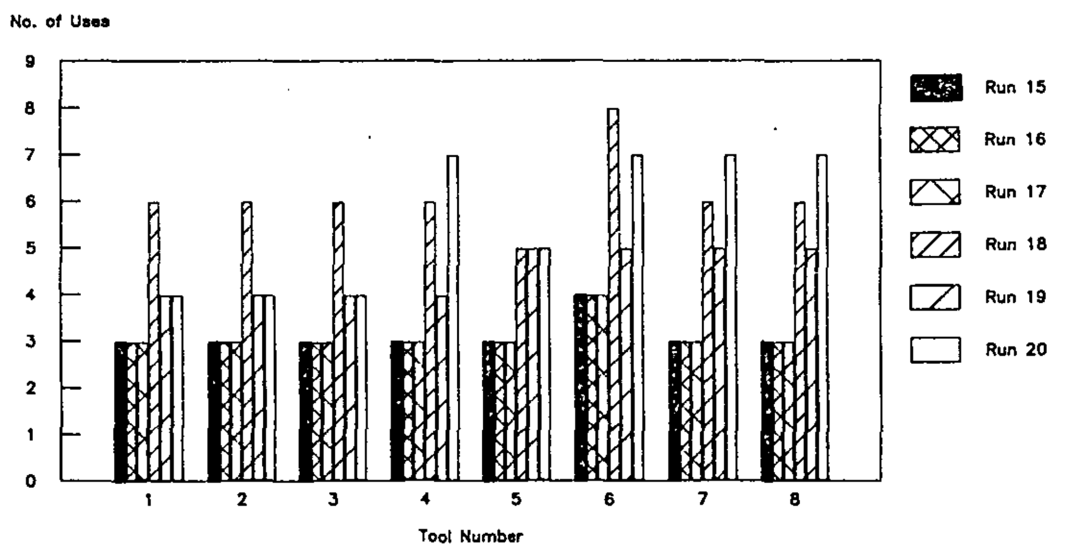


Figure 17.14	*K.B.Modelling Level 3 of the Extended Cell - Tool Usage Frequency*	LUT - FMS Research Group

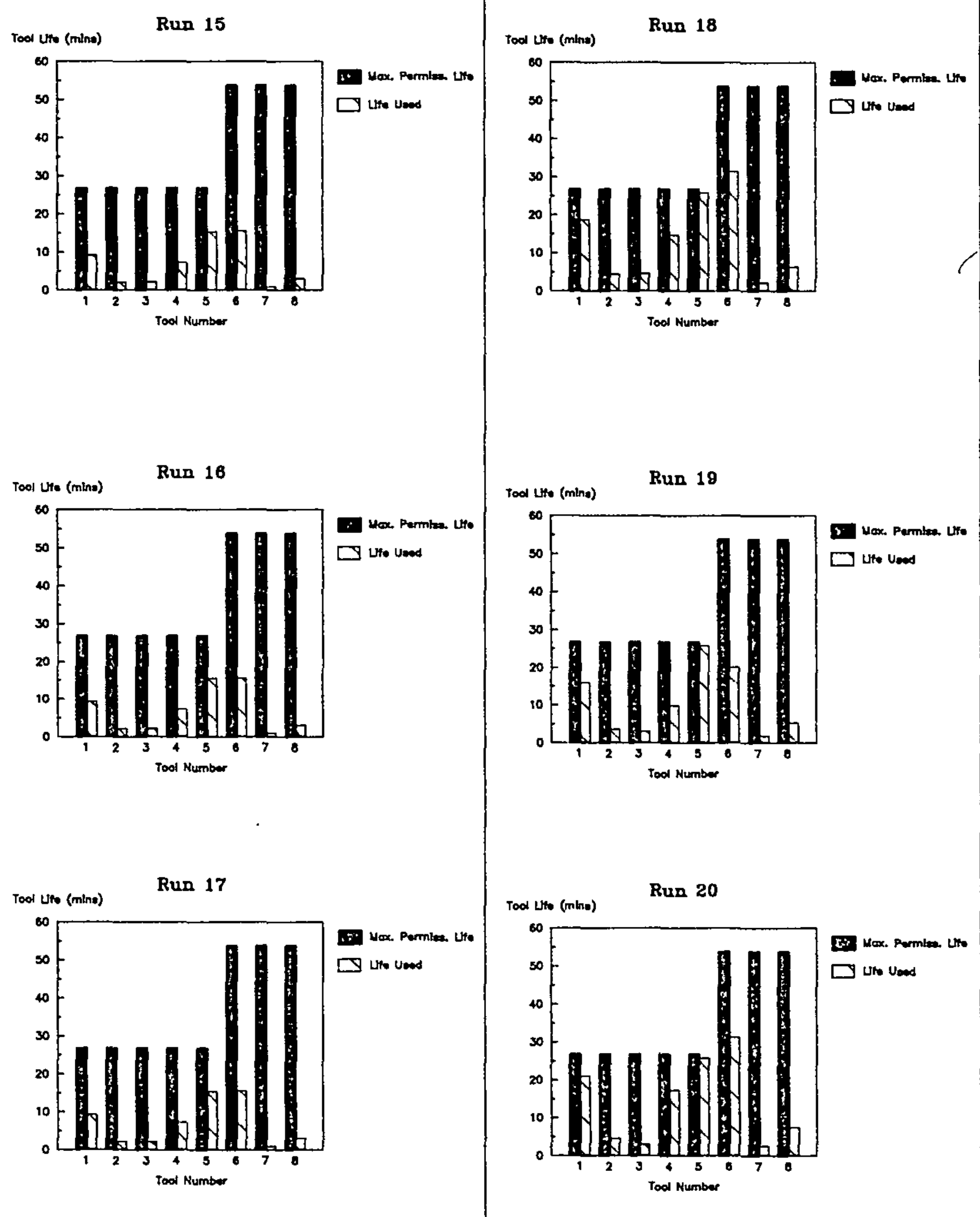


Figure 17.15	*K.B.Modelling Level 3 of the Extended Cell - Tool Life Utilization*	LUT-FMS Research Group

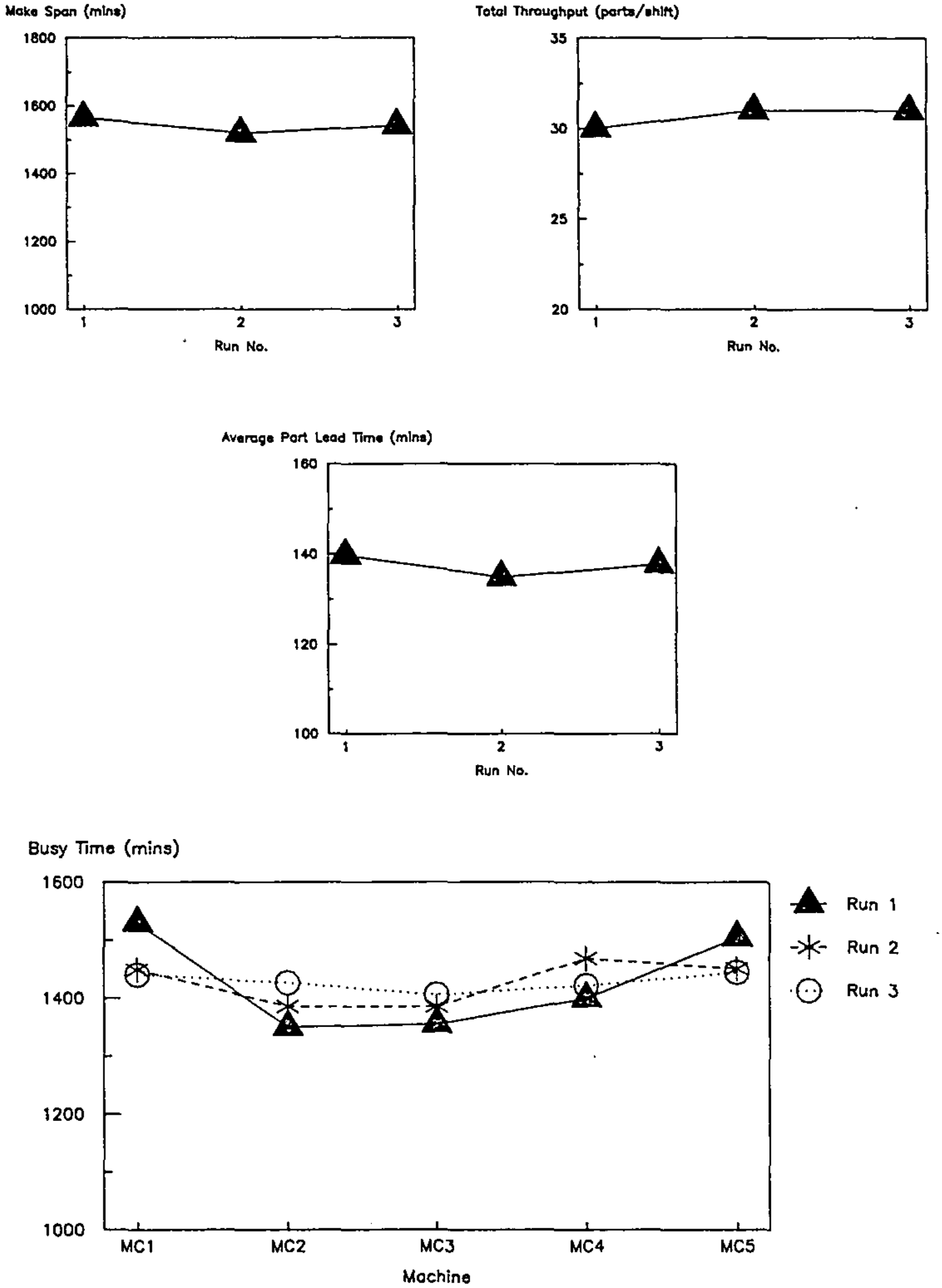


Figure 17.16	*Comparison of the Results under Different Decision Rules for Batch Production - Level 1*	LUT-FMS Research Group

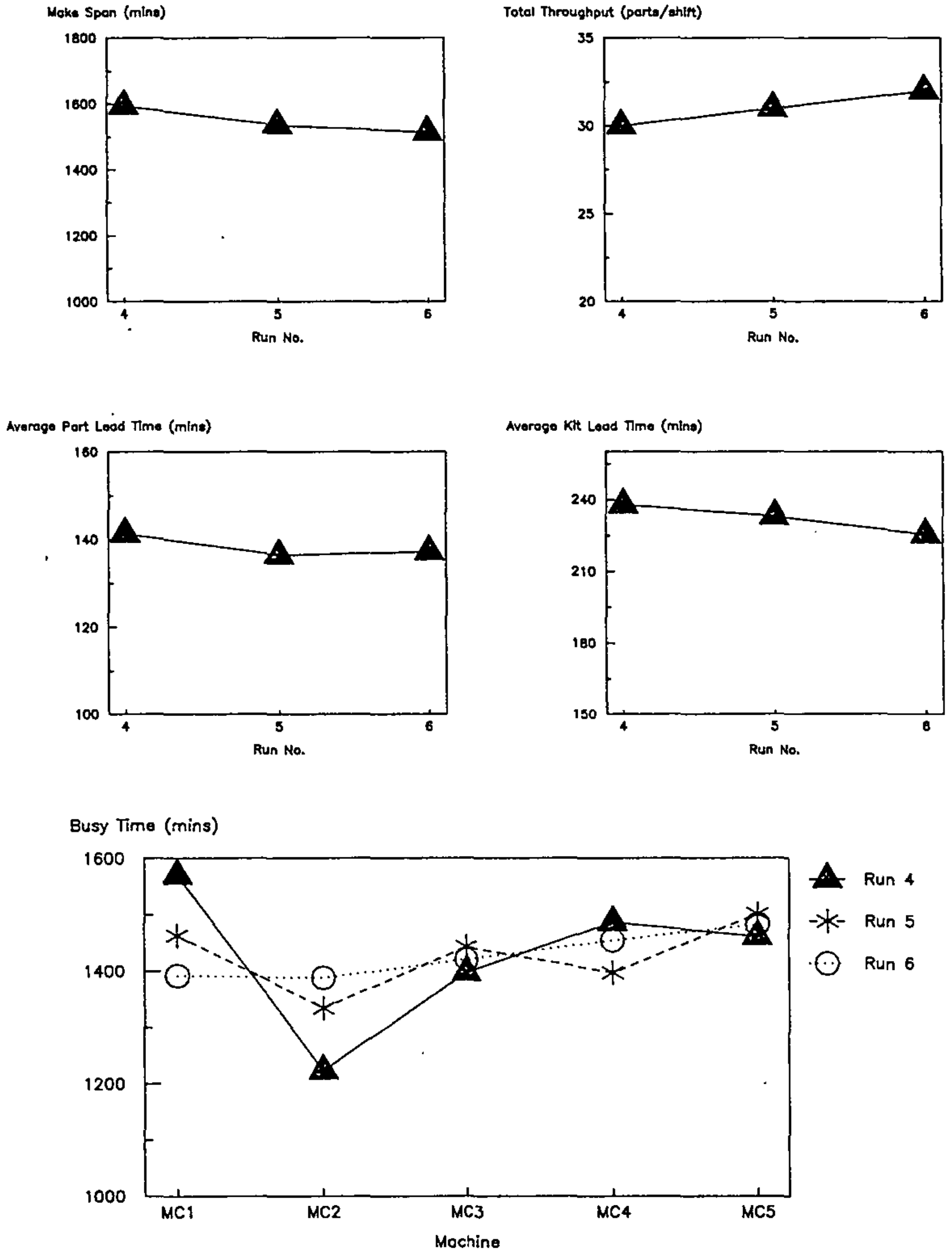


Figure 17.17	*Comparison of the Results under Different Decision Rules for Kit Production - Level 1*	LUT-FMS Research Group

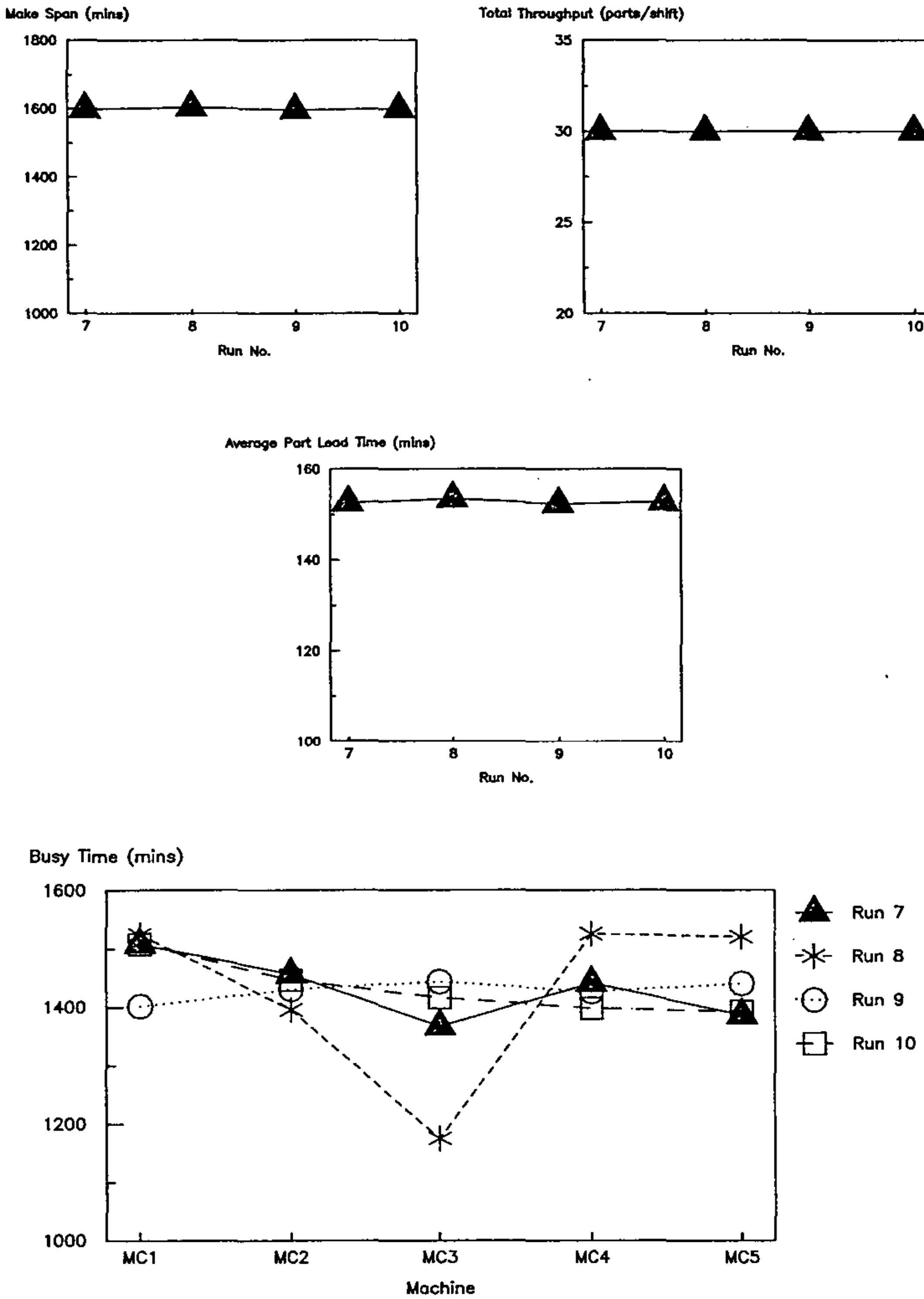


Figure 17.18	*Comparison of the Results under Different Decision Rules for Batch Production - Level 2*	LUT-FMS **Research Group**

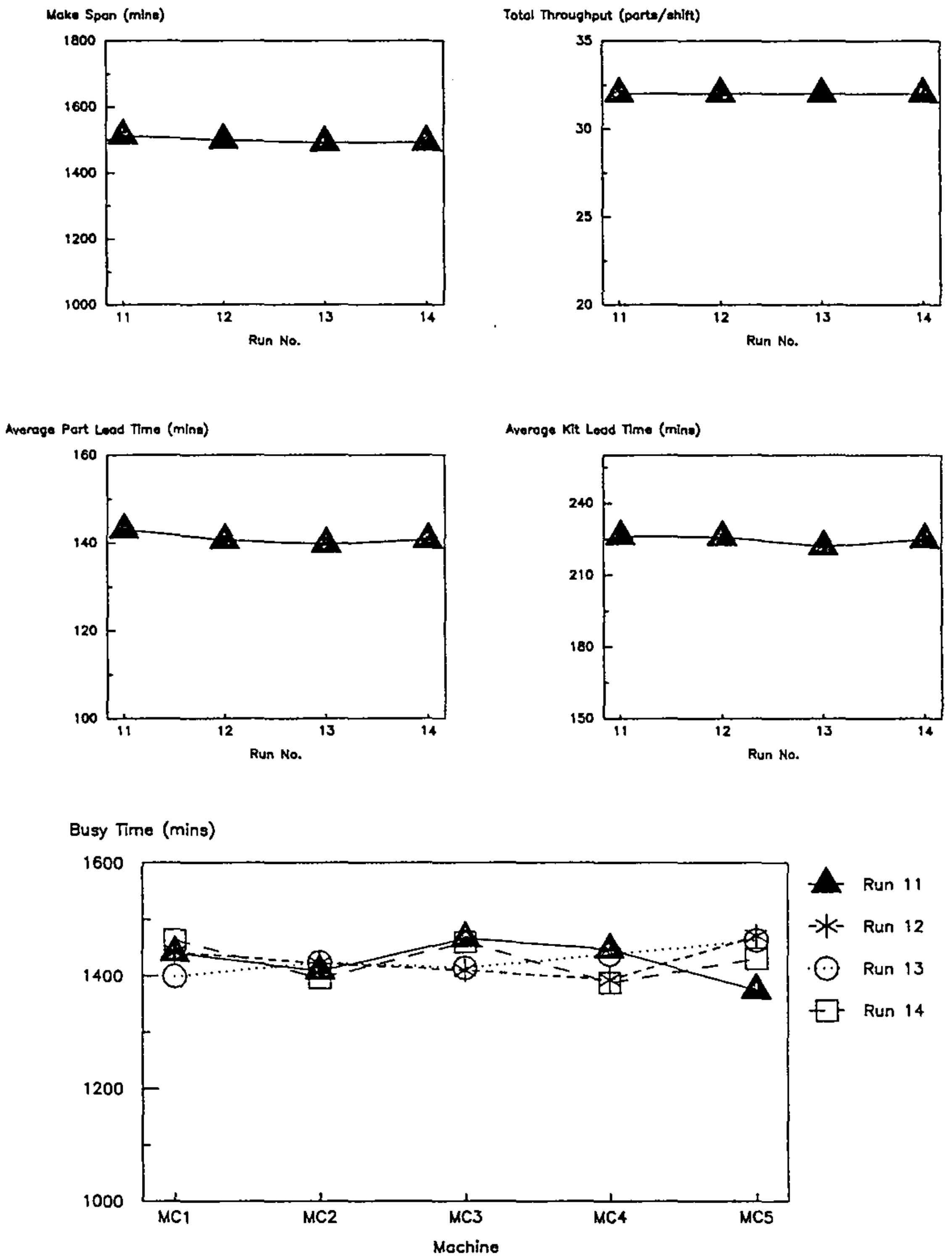


Figure 17.19	*Comparison of the Results under Different Decision Rules for Kit Production - Level 2*	LUT-FMS Research Group

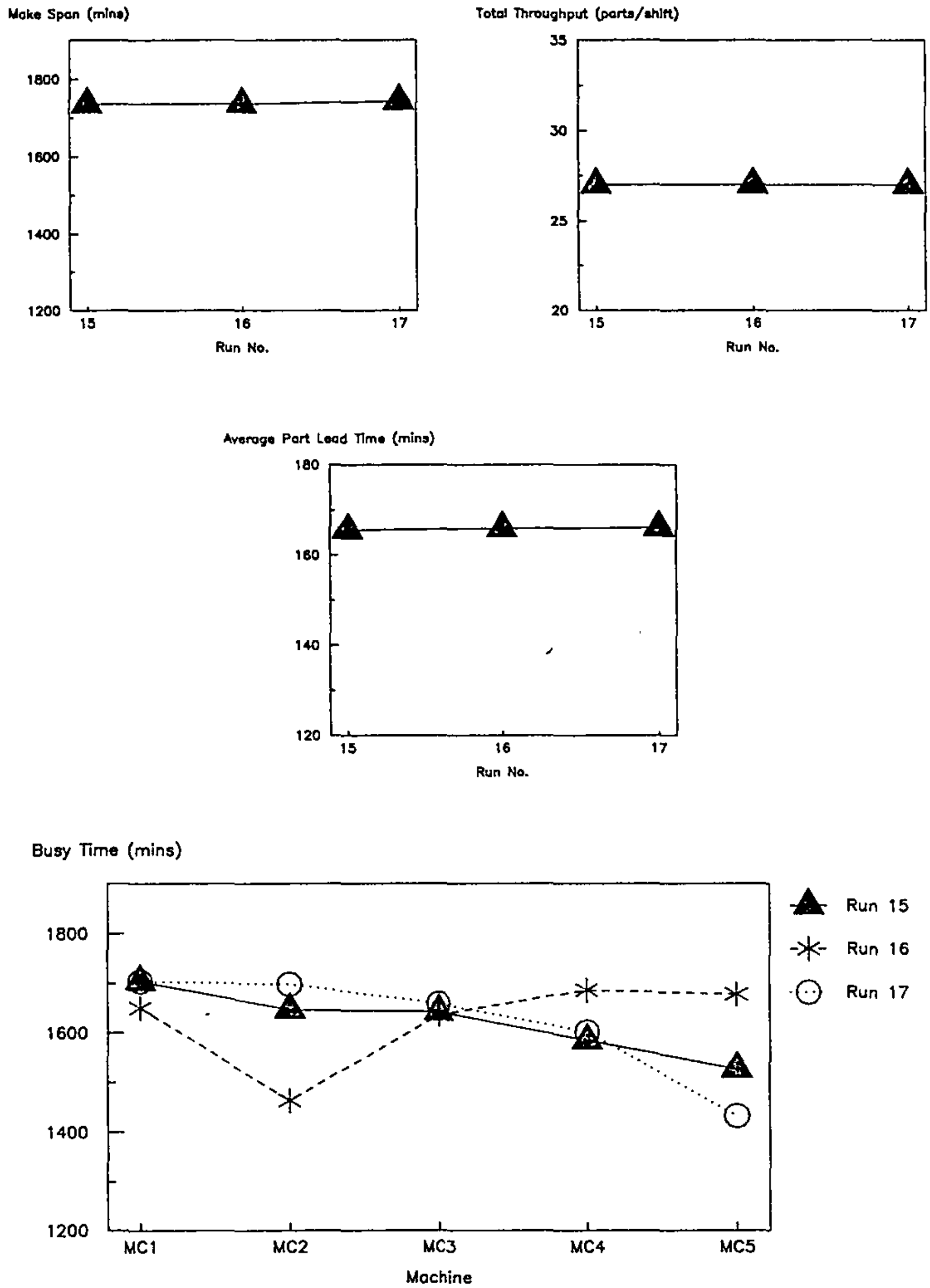
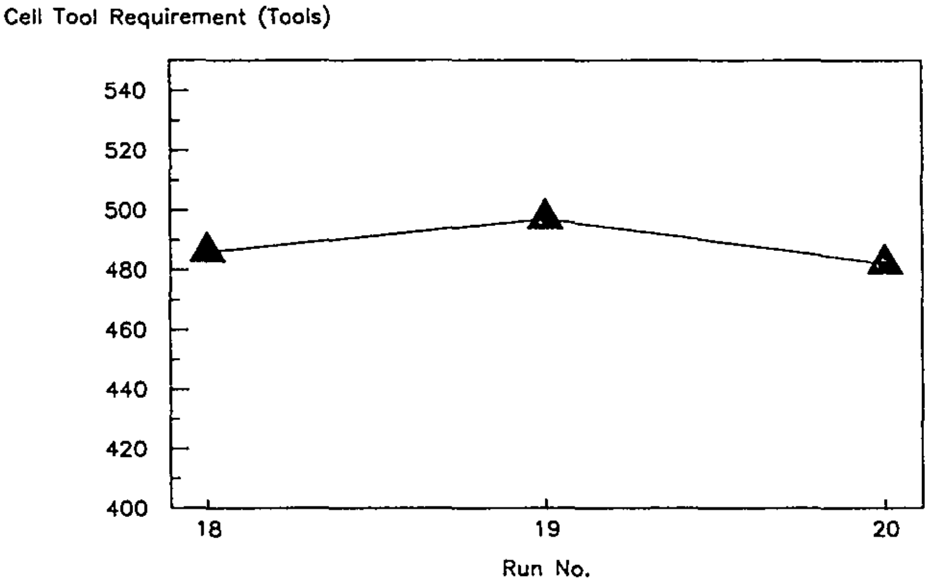


Figure 17.20	*Comparison of the Results under Different Decision Rules for Batch Production - Level 3*	LUT-FMS Research Group



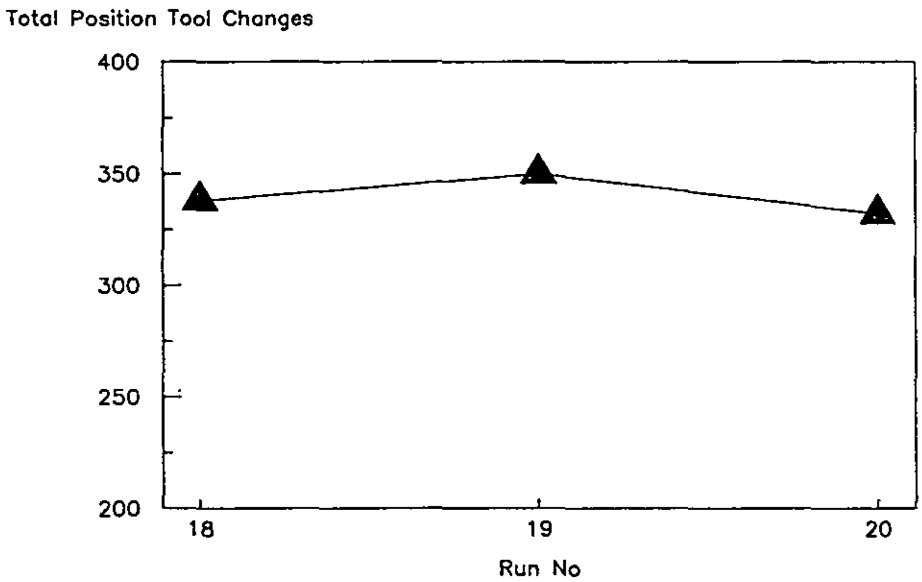


Figure 17.21	*Comparison of the Results under Different Decision Rules for Batch Production - Level 3 (continued)*	LUT-FMS Research Group

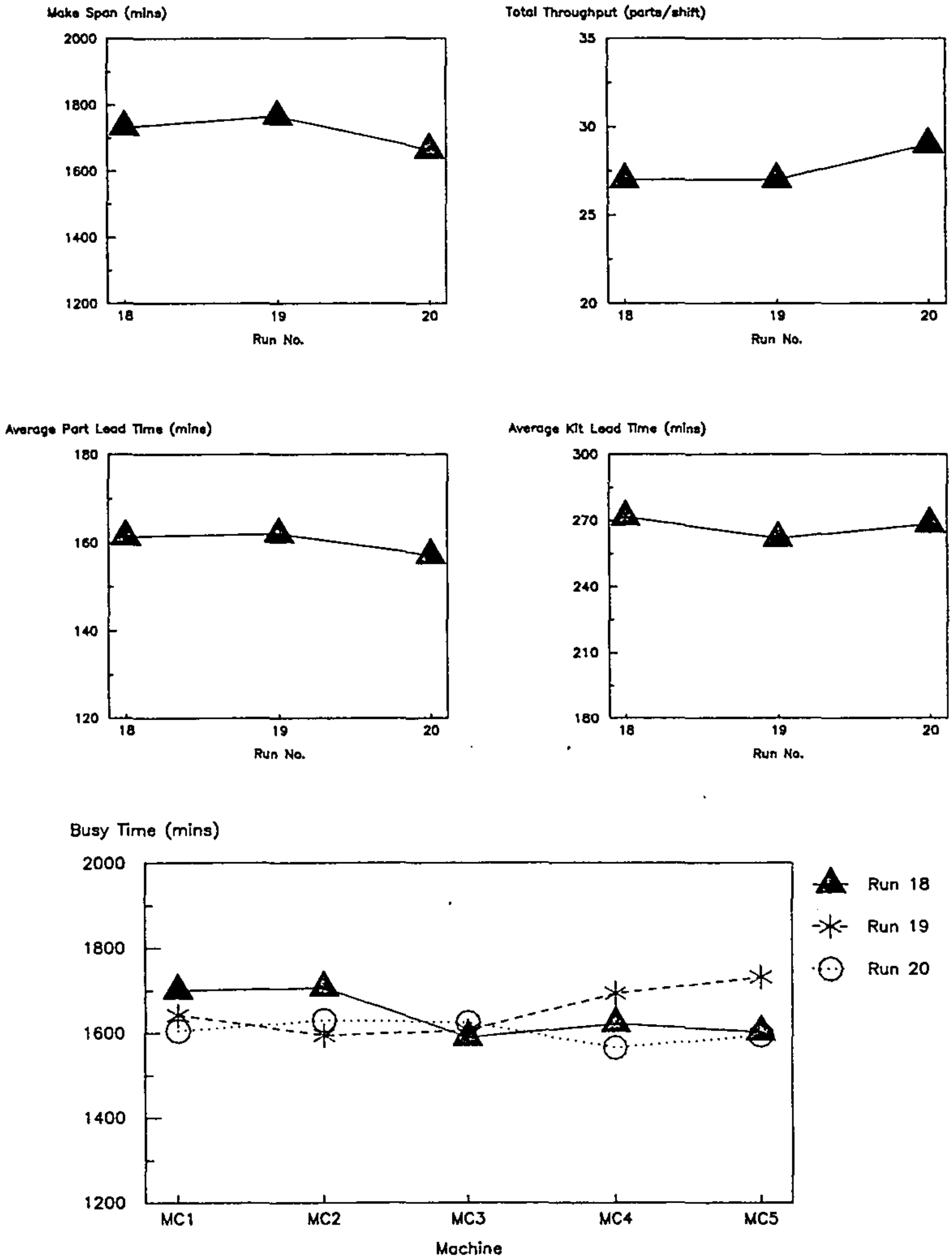
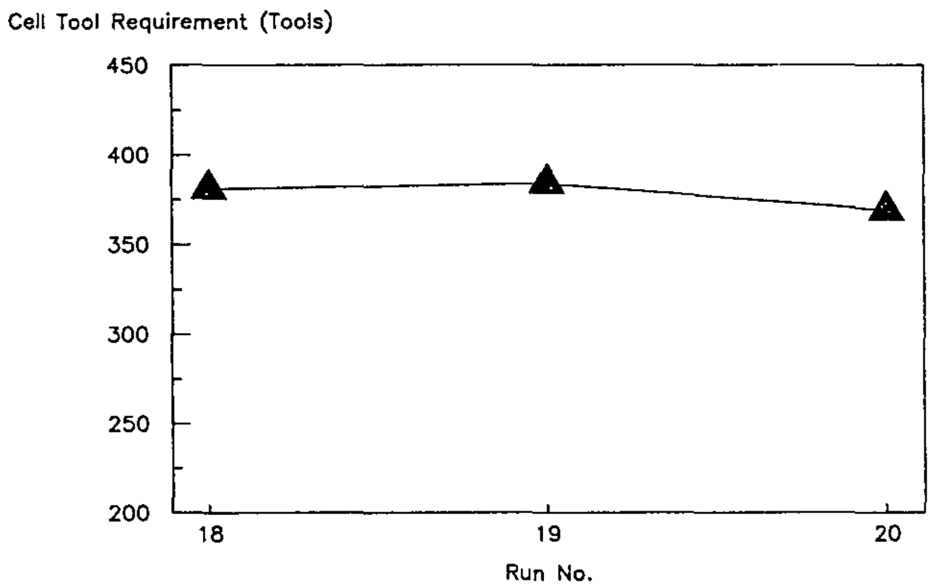


Figure 17.22	*Comparison of the Results under Different Decision Rules for Kit Production - Level 3*	LUT-FMS Research Group



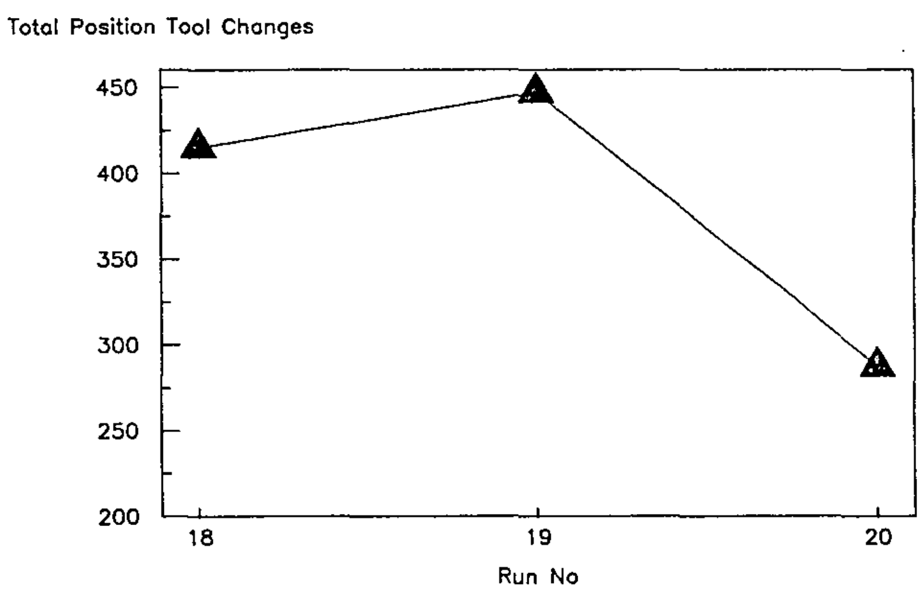


Figure 17.23	*Comparison of the Results under Different Decision Rules for Kit Production - Level 3 (continued)*	LUT-FMS Research Group

Chapter 18
A CRITICAL COMPARISON OF THE KNOWLEDGE BASED MODELLING, THE EMULATOR BASED STUDY AND THE TOOL FLOW MODELLING

18.1 Introduction

In this chapter, an evaluation of the AI based modelling method is conducted. First the results obtained from the knowledge based modelling, the Emulator based study, and the tool flow modelling of the extended cell, are compared. Then the comparison is extended to include other criteria which are identified according to the requirements of manufacturing system modelling. This will naturally lead to the concluding discussion on the potential value of the knowledge based modelling system to be presented in the next chapter.

18.2 Planned Experiments for Comparison

18.2.1 Initial Comments

Since the operation of level 2 of the knowledge based modelling system is very similar to that of the Emulator [43], the results from these two studies can be compared. For the tool flow modelling [84], as it does not consider any work flow, its operation is not completely compatible with that of the level 3 modelling. However, the cell tool requirments generated from the tool flow modelling and the performance of the tool transporters can be compared with those obtained from the level 3 modelling.

It is appparent that the part process data for all studies is exactly the same, but the cell parameters used in these studies can be slightly different. For example, a time matrix has been used in the Emulator to model the travelling of the AGV between stations, whereas in the level 2 and level 3 modelling, an average transfer time is assumed. As a result, it can be expected that the results provided form these studies can be affected by these assumptions.

Like the comparison of the three levels of modelling in Chapters 12 to 15, appropriate operational strategies have to be specified for each study so that the operation of the extended cell model within each study can be made mostly compatible. In the following , these strategies are to be discussed for each of the experiments.

18.2.2 The Knowledge Based Modelling Level 2 and the Emulator

Two experiments are to be performed with regard to the comparison bwtween the knowledge based modelling level 2 and the Emulator, one for batch production and the other for kit production. In both of these cases, parts are released according to the order number and the *FIFO* pallet priority rule is to be used. As five alternative machines are assigned to each machining operation, and two load/unload stations to the palletisation and depalletisation operations, the station selection rule must be carefully developed. In the Emulator, each time when more than one machines are available, the one which has the most spare spaces is selected. Therefore, the *Most Spare Spaces* rule should be selected in the level 2 modelling.

As the Emulator does not consider temporary storages at the stage that these experiments are planned, the 18 pallet stands are not to be modelled in the level 2 study either. Therefore the pallets will be transferred only between the load/unload stations and the machine stations. As a result, the results as obtained from Runs 7 and 11 can be used for the comparison.

18.2.3 The Knowledge Based Modelling Level 3 and the Tool Flow Modelling System

As descibed in [84], load/unload stations, work transporters, temporary part storages and the work flow are all not considered in the tool flow modelling system. Instead, a machining list, which is either generated from a work scheduler or input manually, has to be specified for each machine. According to this machining list, a machine tool is selected first for the processing of each statge of all the parts' work lists. Then the cutting tools are selected and assigned to the machines, with a schedule for the transfer of cutting tools being determined. Finally a transporter device is selected, among the candidate devices, to carry the selection of cutting tools scheduled for delivery to the machines. Thus this tool flow modelling method is of a algorithmic nature.

In the level 3 modelling, however, work and tool flow in a dynamic and integrated way, i.e. work flow can be influenced by tool availability and tool flow be determined by work scheduling. Therefore, in order to make the operation of level 3 more compatible with that of the tool flow modelling, the following strategies are to be considered:

(1) The station operation schedules generated in the level 3 modelling is to be used as the machining list for the tool flow modelling.

(2) The influence of tool availability on work flow is to be eliminated fo rthe level 3 modelling. Therefore the operation of the level 3 can be seen as tools being pulled by workpieces, and this, in a sense, can be considered as similar to the logic of the tool flow modelling method. This is achieved by developing station selection rules which do not take into consideration the tools available in the machine PTSs. An example of such a rule can be the *Most Spare Spaces* rule which determines the priority of the machines according to the spare spaces available at the machines.

(3) It is obvious that the level 3 modelling and the tool flow modelling should use the same tool issue strategy. At this stage, there is only one option in the knwoledge based modelling system, i.e. the differential kitting strategy. According to [84], the differential kitting strategy is also available in the Tool Flow Modelling System.

However, because of the complex part spectrum, the associated machining actvities and certain assumptions made in the model, the use of this strategy had lead to computer stack overflow. Therefore another strategy, the *Single Tools* startegy [84], is to be used. With this strategy, the number of tools loaded and transferred from the STS is dependent upon the capabilities and capacity of the tool transporters and the PTS. The tools are loaded subject to the machining list of the machine and not constrained by part type or part set assignement. As a result, tool sharing among several batches of parts is more progressive and tool changing is minimised at the expense of a large local tool inventory.

Due to the use of different tool issue strategies, the results generated from the two studies can be expected to be differing, but interpretation will be given in the text with regard to this difference.

(4) As mentioned before, the machine tool magazine capacity of the extended cell has been specified as 40 so as to allow for considerable tool sharing across different components or machines. The results obtained from the level 3 modelling as discussed in the previous chapter has verified this consideration.

For the tool flow modelling, however, because of the use of the *Single Tools* issue strategy, the specification of a 40-tool magazine capacity is no longer appropriate. The reason for this is that the single tools issue strategy usually finds its application in high volume/mid-variety production, and is particularly suited to machines with a large PTS capacity. Thererfore, in order to obtain realistic results from the tool flow modelling, a magazine capacity of 120 tools is to be specified for each machine in the tool flow modelling.

(5) The tool assignment strategy is such that no tools are initially assigned to the machine tool magazines. This strategy is to be used in both studies.

From the above discussion it can be concluded that the results of Runs 15, 16, 18 or 19 can be used for this comparison. Because of the run time of the tool flow modelling, only one run is planned for the tool flow modelling, and the results of Run 15 is selected for the comparison.

18.3 Content of Results Obtained

18.3.1 Initial Comments

As the results of the knowledge based modelling for the above planned experiments have been summarized and discussed in the previous chapter, only the results from the Emulator and the Tool Flow Modelling System are to be presented in the following. Detailed listing of these results is given in Appendix VII.4 and Appendix VII.5. The emphasis is placed on the content of the results which can be provided by each method.

18.3.2 Results of the Emulator Based Study

The outputs which can be provided by the Emulator typically include part lead times, machine performance figures, work transporter performance figures, load/unload station performance figures and machine work lists. In addition, an approximate tooling requirement for each machine can be computed according to the workpieces scheduled to the machines. This is similar to the maximum tool requirements planning conducted at the level 2 of the knowledge based modelling system, but the tool life management within the Emulator is much more aggregate. More importantly, the Emulator can provide an animated graphics display with regard to the operation of the model. Figures 18.1 and 18.2 summarize the results obtained from the Emulator based study of the extended cell.

18.3.3 Results of the Tool Flow Modelling

Unlike the normal statistics based outputs obtained form simulations, the algorithmic approach allows the Tool Flow Modelling System to record, manipulate and output considerable amounts of user specific data on the operation of tooling systems within an overall manufacturing system [84]. Typical ouputs consist of final promary tool store contents, final tooling details, machining history and cell performance, machine activites, cell tool transfer, tool transporter contents and schedule, and cell tool summary and tool status report. For a summary of the results obtained from the tool flow modelling of the extended cell, see Figure 18.3.

18.4 Comparison of Results

18.4.1 Scope of the Comparison

This section focuses on the comparison of the results obtained from the knowledge based modelling, the Emulator based study and the tool flow modelling, which have been summarized in the previous Chapter or section. To demonstrate the insight as provided by each study with regard to the behaviour of the cell, the comparison can be conducted along three dimensions. Firstly the overall cell performance is compared. Then a comparison is made with respect to the performance of work flow within the cell, which is mainly between the level 2 modelling and the Emulator. After that, the level 3 modelling is compared with the tool flow modelling with regard to the tool flow performance.

18.4.2 Overall Cell Performance

18.4.2.1 Total Throughput Time

As shown in Figures 18.4 and 18.5, under both batch and kit production, the total throughput time of the level 2 modelling was very close to that of the Emulator based study. This is because the operation of the level 2 is similar to that of the Emulator and compatible operational rules were chosen for the study in both cases.

As for the level 3 modelling and the tool flow modelling, there was a fairly considerable difference in the total throughput time. As mentioned in the previous chapter, the tool flow model does not consider any work flow, and therefore the total throughput time of the tool flow modelling should have been shorter if the work transfer within the cell were more significant than the tool transfer. Hence, it can be concluded that the total time required for tool transfer in the cell was much longer than that required for work transfer. This also implies that the overall cell performance was dominated by tool flow. This can be shown again in the later discussion on the performance of the work and tool transporters.

18.4.2.2 Machine Performance Times

The machine performance time is defined as the time period between the start and the end of the use of a machine. By comparing the performance times of a machine obtained from different modelling methods, the aggregate use of the machine within each model can be demonstrated.

From Figures 18.7 and 18.8 it can be seen that the performance time of each machine for the level 2 modelling and the Emulator was very close to each other under both batch and kit production conditions. In particular, the total performance time of the five machines for the level 2 and the Emulator was approximately the same. The implication of this is that the overall use of the five machines within these two studis was considerably similar.

However, except machine 1, the performance time of the other four machines for the tool flow modelling was longer than that of the level 3 modelling (Figure 18.9). The implication for this difference is that the efficiency of the use of the five machines within the tool flow modelling is lower than that within the level 3 modelling, i.e. in general the idle time of these machines in the tool flow modelling was longer. The reason for this should be the use of diffrent tool issue strategies in the two studies. In the level 3 modelling a machine could start machining once the tools for the first component had arrived, whereas in the tool flow modelling, the machine could only start machining after the tools for certain batches of components had been transferred from the STS. In fact, this is also another reason that the total throughput time of the tool flow modelling was longer than that of the level 3 modelling.

18.4.3 Work Flow

18.4.3.1 Part Lead Times

As shown in Figure 18.10, the average part lead time for each part type under batch prodcution was very close for the level 2 modelling and the Emulator based study. This because in the case batch prodcution, identical components were released into the cell successively, and therefore the work assignment to alternative stations did not significantly affect the work flow performance. However, in the case of kit prodcution, since different components were released one after another, the assignment of a component to a different station could result in a totally diffrent work flow pattren. Thus, as depicted in Figure 18.11, the average part lead time varies considerably.

18.4.3.2 Machine Station Performance

As shown in Figure 18.12, under batch production condition, the utilization of each machine was slightly diffrent for the level 2 and the Emulator. Machines 1, 4 and 5 were utilized more at level 2, whereas in the Emulator machines 2 and 3 were used more. This difference in machine utilization was smaller in the case of kit production (Figure 18.13).

In section 18.2, it was claimed that the use of the *Most Spare Spaces* rule for station selection should make the operation of the level 2 model compatible with that of the Emulator. However, as have been discussed in the early chapters, an average transfer time was entered for the level 2 modelling. Thus it happened that the decision with regard to the selection of a station was made at a different time in the Emulator based study. The effect of this was that the same component have been sent to different stations in the two models, resulting in the variation on the use of the machines.

18.4.3.3 Load/Unload Station Performance

From figures 18.4 and 18.5 it can be seen that each load/unload station was utilized alomost exactly the same in the level 2 and the Emulator. The reason for this was that the same duration was specified for the re-palletisation activity for different components and that the same number of components were sent to a particular station.

The only major difference between these two studies with regard to the load/unload station performance was that the stationary time in the case of level 2 was longer than that in the case of the Emulator for both stations. This means that in the Emulator, a palletised component was more quickly transferred away from the load/unload station. One of the reasons could be that the AGV spent more time on travelling in the level 2 modelling, and this can be shown in the next section on the performance of the work transporter.

18.4.3.4 Work Transporter Performance

Because of the assumption of the average transfer time within the level 2 modelling, the total travelling time for the AGV at level 2 was longer than that in the Emulator (Figure 18.16). This is consistent with the argument made in the previous section.

18.4.3.5 Machine Work List

The workpieces scheduled to machine 1 in both the level 2 modelling and the Emulator based study are summarized in Figure 18.17. As can be expected, the machine work list for the level 2 and the Emulator in the case of batch production was more close to each other than in the case of kit production. The explanation given in section 18.4.3 also applies here.

In fact, the difference between the level 2 the Emulator with respect to the machine work list under kit production is so considerable that the two lists produced resemble very little. This furthur stresses the influence of the modelling assumptions in conjunction with the decision rules on the cell performance.

18.4.4 Tool Flow

18.4.4.1 Cell Tool Requirements

As illustrated in Figure 18.18, the total cell tool requirement for the level 3 and the tool flow modelling was fairly close to each other, with the tool flow model used 7 more tools. This is because both methods used the work-oriented tool management strategy, and the tool flow modelling used the work list generated from the level 3 modelling. The slight difference should be seen as being brought by the use of different tool issue strategies.

18.4.4.2 PTS Tool Utilization

Since the single tools issue strategy, as applied in the tool flow model, generated the tool requirements by taking into account several batches of components accoring to the capacities of the tool magazine and the tool transporter, the number of tools used on each machine in the tool flow model was different from that in the level 3 (Figure 18.19). In the tool flow modelling, the capacity of each PTS and the tool transporter was specified as 120, and hence a tool could be transported to the machine just once and be used across several components.

In the level 3 modelling, a 40-tool magazine was used on each machine, and therefore a tool could be transported to the same machine several times though it could also be used across anumber of components. Further more, as the newly provision of the same tool from the STS was considered as another tool, then the PTS tool utilization in the level 3 modelling was higher than that in the tool flow modelling.

18.4.4.3 Cell Tool Wait Time

As shown in Figure 18.20, because of the use of the different tool issue strategies, each machine spent more time on waiting for tools in the tool flow modelling. This is because in the tool flow modelling a machine could not start machining until the tools required for several batches of components were transferred to the PTS, while in the level 3 the machine could start its machining process as soon as the tools for the first component had arrived. From the machine utilization point of view, the level 3 used the machines more efficiently.

Notably, from machines 1 to 5, more and more time was required for tool wait in the tool flow modelling. The reason for this was that in this model, the STS had obly one service position and the priority for tool provision for these five machines was ordered decreasingly. As a result, since machine 5 had the lowest priority, the transporters could only transfer tools for this machine after the other machines had obtained the required tools.

18.4.4.4 Tool Transporter Performance

From Figure 18.21 it can be found that the utilization of the three tool transporters (men) for the level 3 and the tool flow model was considerably different. This is mainly due to the fact that the performance of the tool transporters was significantly affected by the tool issue strategy. In the tool flow model, man 2 and man 3 were both used twice, while man 1 was only used once for the trasnfer of tools to machine 3, and therefore the utilization of man 1 was lower than that of the other two.

In the case of the level 3 modelling, however, tool transporters were used more frequently because a differential kit of tools must be transported for each component, and the use of the *Specified Priority* rule favoured the selection of man 1.

18.4.4.5 Sister Tooling Prediction

Again, since the tool flow model used the machining list generated from the level 3 modelling, the sister tools required for each tool type were also very similar. A depiction of the sister tools for a sub-set of eight tool types is shown in Figure 18.22, with the predictions being exactly the same.

18.4.4.6 Tool Life Utilisation

As shown in Figure 18.23, since the same tool number in the level 3 and the tool flow model could be different tool types, the life utilization for a tool can not be compared side by side for the two studies. However, a more close examination on the status of these tools should lead to the conclusion that most of the tools were under-utilized as well in the tool flow model.

18.4.4.7 Tool Usage Frequency

Due to the same fact as given above, a comparison with regard to the tool usage frequency can only demonstrate the aggregate behaviour of the tools within the two models (Figure 18.24).

18.4.5 Conclusions

Drawn on the above comparison of the results obtained from the three modelling studies, the following conclusions can be reached:

(1) When pure work flow is condidered in the modelling, the performance figures obtained from the level 2 and the Emulator are very similar to each other.

(2) If alternative stations are assigned to an operation of a part, the resulted work flow pattern at level 2 can significantly be different from that of the Emulator because of the transfer times used in each method, but the aggregate performance figures can still be close as a result of the use of similar decision rules.

(3) The performance of the extended cell is dominated by the tool flow in the cell if dual flow is considered.

(4) The tool requirements of the extended cell are mainly determined by the machine work lists.

(5) From the comparison of the results obtained from the level 3 modelling and

the tool flow modelling of the extended cell, the influence of work and/or tool flow on the system performance can take three basic forms. The first is the case where work transfer between stations is more significant than tool flow between tool stores. This happens when the cell work transport system is a bottleneck, or when the number of tools required is not considerable or most tools are always available in the magazines. In this case, the cell performance is dictated by the work flow within the cell.

On the other extreme, there can be cases where tool flow is much more significant than work flow. A typical example if the extended cell in which the time a component spends on travelling between stations is much shorter than the time it spends on waiting for tools. In this case, the cell performance is dominated by the tool flow within the cell. The third form takes in between these two extremes, where work and tool flow are equally important and the cell performance relies on both of them.

(6) Use of different tool issue strategies can also affect the generation of the cell tool requirements, but the major influence is on the performance of the machines and tool transporters.

18.5 Comparison of Modelling Methods

18.5.1 Criteria for the Comparison

To compare the AI based modelling method, the Emulation method and the tool flow modelling method, the following criteria are identified according to the requirements of manufacturing system modelling (see Chapter 4) [130a]:

(1) Skill Requirments,
(2) Time Scales for Model Building,
(3) Level of Detail Achievable
(4) Problem Areas Tackled
(5) Capacity to Provide Solutions
(6) Flexibility to Modelling of Alternative Systems
(7) Capability to Model User-Specific Operational Rules
(8) Transparency of Modelling Process
(9) User Friendliness and Visual Appeal
(10) Confidence Level

(11) Computational Performance

(12) Hardware and Software Constraints.

In the following sections, the three modelling methods are to be compared with regard to each of these criteria.

18.5.2 Skill Requirements

As reported in [43], the Emulator has been designed as a suit of data driven software which can be used by manufacturing design engineers. Thus it can be classified as a generalised simulator which consists of a validated model that the user configures to his own particular requirements by his own input data. Although algorithmic in nature, the tool flow modelling method is also data driven [84], where more specialised data input has to be entered in order to set up the model for particular applications. Hence it can be concluded that to model a manufacturing system, the above two methods require no programming experience.

In contrast to these data driven methods, the AI based method is of a knowledge driven nature, i.e. both data and rules need to be entered in order to design a model. Although the entering of data requires no programming skill, the expression of operational rules do need the designer to have limited LOOPS knowledge [53]. Of course, the user can also design a model by selecting rules, and in this case no programming is required.

18.5.3 Time Scales for Model Building

Based on the experience gained in the case studies, the time scale for building amodel using the knowledge based modelling system is fairly short. For the window and menu facilities provided by the Xerox Workstation significantly reduce the response time of the computer on communicating with the system. On the other hand, the interactive soft-key approach for data input as used by the other two methods is entirely based on the software written by the researchers using programming languages like PASCAL. Therefore the communication time between the user and the system can be slightly longer.

On the whole, since the Emulator and the tool flow modelling system have been designed to be data-driven, the time requied for model building is fairlt short . In the case of the knowledge based modelling system, a model can be constructed hierarchically, and therefore the time required for data input at an aggregate level can be significantly short. Since the selection of decision rules is menu driven, this mode of rule entry is also a fast process. See the study of the three machine cell for a more close description. However, if the user intends to express new rules, the total time for model building will certainly be longer than the time required for the Emulator and the tool flow modelling system.

18.5.4 Level of Detail Achievable

As described in [43] [244], the Emulator has been developed with a great deal of manufacturing operation detail incorporated. For example, modelling with the Emulator requires a plant layout with detailed information on transport and workpiece handing. Recent enhancement of the Emulator has considered manual operations in a machining cell, such as the manual involment in machine load/unload, machine setup, transporter movement, and repalletising and refixturing.

By ignoring the work flow in a cell, the tool flow modelling system concentrates in great detail the tool flow activities at the single machine level, multi-machine level or multi-cell level. Typical elements that are modelled include the tool transfer network, centralised and/or decentralised tool storage facilities, automated tool exchange mechanisms and the tool refurbishment facility.

In contrast to the above two methods, the knwoledge based modelling system contains limited details and each level has its associated assumptions though additional details like dual work and tool flow can be modelled. This is simply because the main objective of this thesis has been to explore the design and construction of the system structure, rather than the operation details that should be considered. Nevertheless, thanks to the knowledge system structure of the modelling system, other modelling details can be conveniently included. This is in contrast to the other two methods, where the extension to other modelling details is a fairly difficult task because more effort is required to understand the existing structure and computer code.

18.5.5 Problem Areas Tackled

By selecting the appropriate modelling levels, the knowledge based modelling system can be used at various stages during the design of a manufacturing system. For an existing manufacturing system, models are usually required to help to improve the performance of the system. This can be achieved with the knowledge based modelling system by expressing and trying out different operational rules, and therefore better control strategies can always be defined and brought into use.

The Emulator and the tool flow modelling system, however, can be seen as two effective tools for the detailed study of a manufacturing system at the design stage. For the operation of the system, the user can only select and try out the limited built-in strategies and rules.

18.5.6 Capacity to Provide Solutions

The capacity of a modelling method with regard to the provision of solutions is concerned with the scale of the systems that can be modelled by the method. Basically, this is dependent on the hardware used by the method and the modelling details contained. In the case of the Emulator, the size of the system modelled can be considerably large because of the use of the INTEL SYS 310 computer which allows the parallel processing of different tasks, with each appropriate task being allocated to a separate single board computer [43].

The limitation of the tool flow modelling system may be more considerable based on the experience gained in the comparative case study. Typically the system entities that can be modelled on an IBM PC computer include 10 machines with upto 120 tool capacity, 6000 tools in the installation and 4 tool transporters with upto 120 tool capacity [84].

As for the knowledge based modelling system, since the computer program works together with the specific application data in the virtual memory [302], there can be problems. With the incorporation of more and more modelling details, the computer program gets increasingly larger, but the maximum system size modelled becomes smaller. This is rather conflicting with the intention of modelling more complex systems with larger computer programs.

18.5.7 Flexibility to Modelling of Alternative Sytems

The comparison of the three modelling methods with respect to the flexibility to the modelling of alternative systems is considered in terms of the structural configuration of alternative models when designing a machining cell.

In the case of the knowledge based modelling system, as each system element is represented as an object which can be conveniently added, deleted or modified without affecting the others using the model configurator and the data base browser, configuration of alternative models is greatly straightforward and convenient. Especially, as everying is running in the virtual memory, modification of an existing model and the running of the alternative model can be readily carried out.

As for the Emulator and the tool flow modelling system, the process associated with modifying an existing model is more rigid because of the soft-key approach used for data input. For instance, addition of a machine can involve many selections of the menu options on the screen in order to get to the desired data file, and to run the new model other set-up procedures are normally required.

18.5.8 Capability to Model User-Specific Decision Rules

It is obvious that the AI based method is superior to the other two methods with regard to the modelling of user-specific operational rules of a manufacturing system. In the case of the Emulator and the tool flow modelling system, the user can only try out the options provided, and there is no way that he can immediately express his own rules. In contrast, by guiding the user to the defined decision points, he can enter his own rules in the LOOPS rule language without affecting the other parts of the program.

18.5.9 Transparency of the Modelling Process

After a model is established, the user needs to know how the model behaves and the relationship between the computer code and the model behaviour. Because of the representation of the system elements as objects and their interactions as rules, and the capability to trace the application of particular rules in the knowledge based modelling system, the user can relate the status of the system elements, the activities happened with these elements and the rules applied to enable these activities. As a result, he can have a full handle on the behaviour of the model.

In contrast, since the system elements are represented in data files and their interactions in PASCAL procedures or algorithms in the Emulator and the tool flow modelling system, except the experienced users, it is very difficult for a modeller to understand the relationship between the computer code and the model behaviour. Thus debugging of the models with these two methods can be a very difficult task.

18.5.10 User Friendliness and Visual Appeal

Because of the use of the algorithmic approach, the tool flow modelling system can not display visually the flow of tools on the screen and no facilities are provided with regard to the operation of a model. This is one of the major disadvantages of this method.

On the other hand, the Emulator and the knowledge based modelling system both provide graphics facilities which can display the running of the model on the screen. The Emulatorcan animate the operation of the system elements. This is very useful in debugging and validation of the model by showing whether the results are logical and the model is behaving like the real system.

Although not implemented with animation yet, the knowledge based modelling system has used the split-screen technique to provide graphics outputs. The user can see the model running on the windows representing the status of system elements and can play with these windows to ensure, to his own satisfaction, that they represent the manufacturing system concerned and that the results, therefore, are trustworthy. In addition, the textual output facilities also help the user to understand the operation of the model.

18.5.11 Confidence Level

One of the most important requirements for any modelling methods is that the design performance provided should be as close to the actual performance as possible, i.e. a modelling tool should be able to provide results with adequate confidence.

Based on the comparison of the results of the comparative case study, it can be concluded that the three methods can all meet this requirement, but the confidence level can be different. Since the Emulator is designed to imitate a system in regard to all the variables which it is possible to measure and very few assumptions were made in the model, the results provided by the Emulator should be considered as fairly realistic.

Although in real systems, tools rarely flow without integrating with work flow, the tool flow modelling system mimics the activities of tool flow in great details. Therefore for those systems where the performance is dominated by tool provision, the tool flow modelling system is an effective tool for providing complete solutions for tool management.

The knowledge based modelling system, at this stage, has certain assumptions associated with each of the three levels. Thus the results are, to some extent, approximate in comparison with the actual performance of the modelled system. Further enhancement of the modelling system will eliminate those assumptions which can possibly cause approximations in the results, such as the average transfer time assumed.

18.5.12 Computational Performance

The computational performance of a modelling system depends mainly on two factors, the hardware and software environment, and the software implementation. For the three modelling methods, these factors are all distinguishing. Nevertheless, it can be concluded, based on the case study experience, that the knowledge based modelling system is fairly efficient with the symbolic manipulation Workstation and the contained details. Especially, the high abstraction of level 1 enables considerably fast modelling of a manufacturing system.

Although more details are contained, the Emulator is also an efficient tool because of the use of the powerful computing vehicle. The tool flow modelling system, however, is slightly slow based on the case study experience. This is mainly because of the use of the very limited computer. The implementaion of the algorithms within the model can also be influencing. It is expected that the use of the SUN Workstation will solve this problem.

18.5.13 Hardware and Software Constraints

As mentioned before, the Emulator is currently run on an INTEL SYS 310 computer and the tool flow modelling system on an IBM PC. With the wide availability of SUN Workstations, these two modelling systems are being transferred to the SUN Workstations. Therefore the application of the two modelling systems in industry should not be constrained by the computers required.

The knowledge based modelling system, however, has been developed on the Xerox 1186 Work station which has not been widely used in industry. In addition, the LOOPS software environment is needed in order to run the modelling system. Nevertheless, there has been clear evidence that with the maturity of these AI symbolics machines, application of these machines in industry is becomming increasingly wide. Furthermore, there has been development in running the LOOPS software on SUN Workstations. If this is completely achieved, the wide application of the AI based modelling approach can be expected.

	Required	Actual
Total Part Throughput:	40	40
Time Horizon (mins):	1440.00	1522.09

Machine Performance

Machine	MC1	MC2	MC3	MC4	MC5
Machining	1435.1	1465.1	1400.5	1380.3	1286.0
Load/Unload	40.0	44.0	42.0	38.0	38.0
Idle	105.9	71.9	138.4	162.7	257.0
Tools Used	96	97	98	102	99

AGV Performance

	AGV
Travelling	214.5
Load/Unload	113.1
Idle	1253.3

Load/Unload Performance

Station	LUS1	LUS2
Repalletising	1100.0	920.0
Load/Unload	29.7	24.8
Waiting	19.4	24.3
Idle	431.9	611.9

Man Performance

No. in Use	Time
0	306.0
1	529.9
2	745.1
3	0.0

Man No.	1	2	3
Repalletising	1100.0	920.0	0.0
Idle	481.0	661.0	0.0
Unavailable	0.0	0.0	1581.0

Figure 18.1	*Emulator – Part Flow for Batch Production – Summary of Outputs*	**LUT-FMS Research Group**

	Required	Actual
Total Part Throughput:	40	40
Time Horizon (mins):	1440.00	1504.41

Machine Performance

Machine	MC1	MC2	MC3	MC4	MC5
Machining	1406.5	1374.4	1396.5	1415.6	1341.5
Load/Unload	42.0	38.0	42.0	36.0	44.0
Idle	56.0	92.0	65.9	52.8	118.9
Tools Used	94	95	93	88	95

AGV Performance

	AGV
Travelling	213.7
Load/Unload	113.1
Idle	1177.6

Load/Unload Performance

Station	LUS1	LUS2
Repalletising	1060.0	960.0
Load/Unload	28.6	25.9
Waiting	22.1	25.7
Idle	393.7	492.8

Man Performance

No. in Use	Time
0	174.0
1	640.8
2	689.6
3	0.0

Man No.	1	2	3
Repalletising	1100.0	920.0	0.0
Idle	404.4	584.4	0.0
Unavailable	0.0	0.0	1504.4

Figure 18.2	*Emulator – Part Flow for Kit Production – Summary of Outputs*	**LUT-FMS Research Group**

Machine No.	1	2	3	4	5
No. of Tools on Machine	100	97	105	92	99
Tool No.'s Required	1 – 100	101 – 197	198 – 302	303 – 394	395 – 493
Start	0.1	0.9	1.7	2.5	3.3
Tool Wait at Start	161.0	232.2	318.7	470.2	551.3
Finish Last Activity	1667.9	1683.0	1728.4	1829.9	1881.9
Parts Assigned	1–22	23–43	44–63	64–83	84–101
AGV Used	3	2	1	3	2
Performance Time	1667.8	1682.1	1726.7	1827.4	1878.6

Total Tool Required: 493
Worn Tools: 5
Added Tools: 0

Figure 18.3	*Tool Flow Modelling – Summary of Outputs*	**LUT-FMS Research Group**

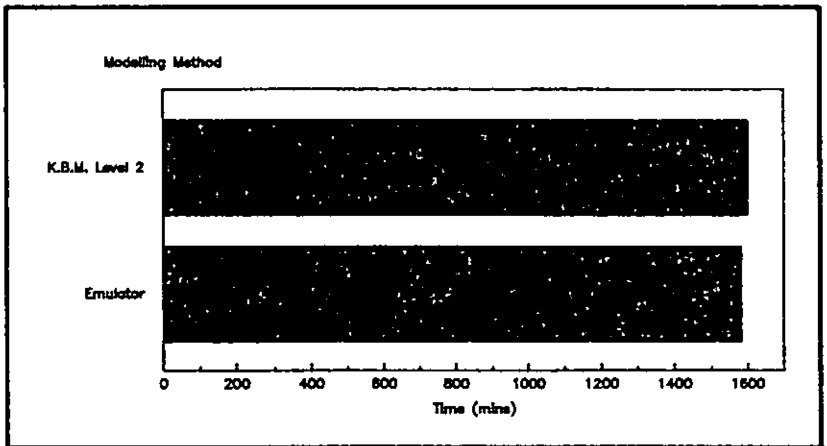


Figure 18.4 | *Comparison of K.B.M. Level 2 with Emulator for Batch Production - Total Throughput Time* | **LUT - FMS Research Group**

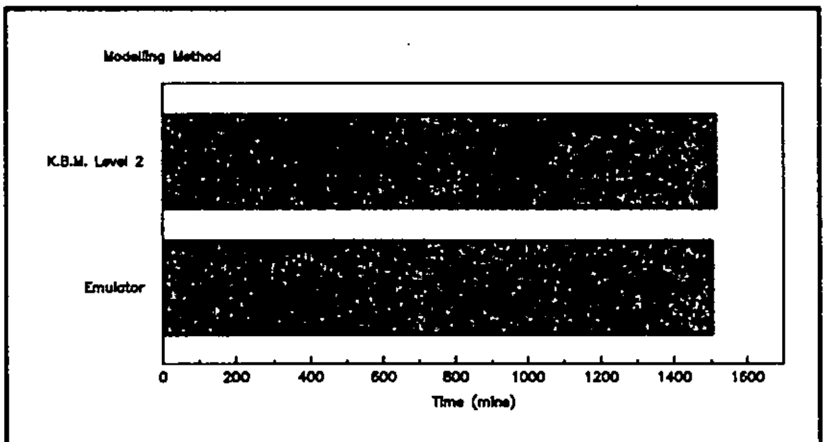


Figure 18.5 | *Comparison of K.B.M. Level 2 with Emulator for Kit Production - Total Throughput Time* | **LUT - FMS Research Group**

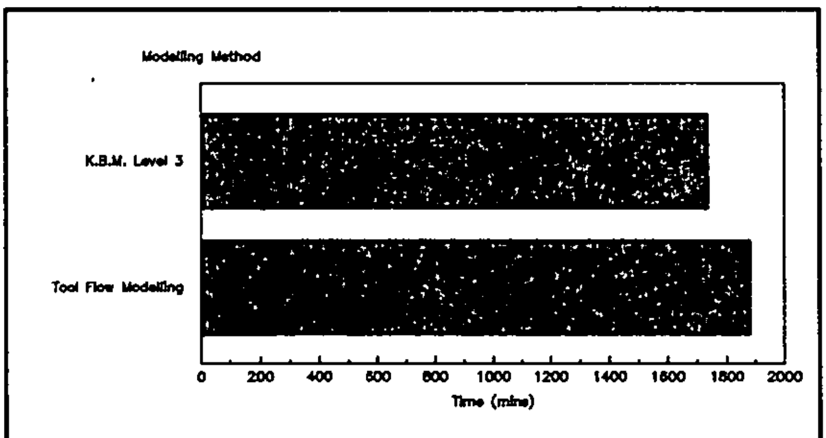


Figure 18.6 | *Comparison of K.B.M. Level 3 with Tool Flow Modelling - Total Throughput Time* | **LUT - FMS Research Group**

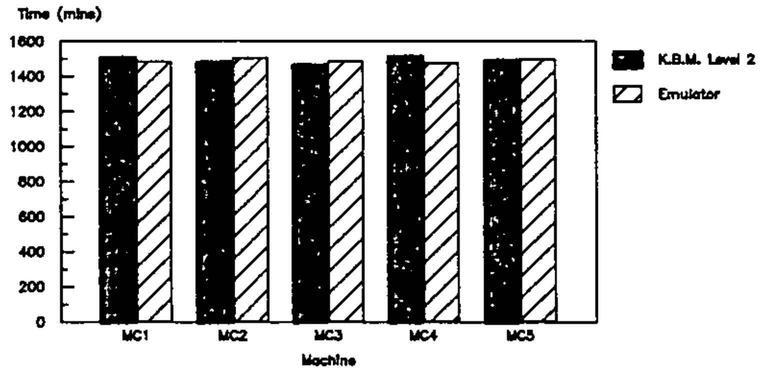


Figure 18.7	*Comparison of K.B.M. Level 2 with Emulator for Batch Production - Machine Performance Time*	**LUT - FMS** Research Group

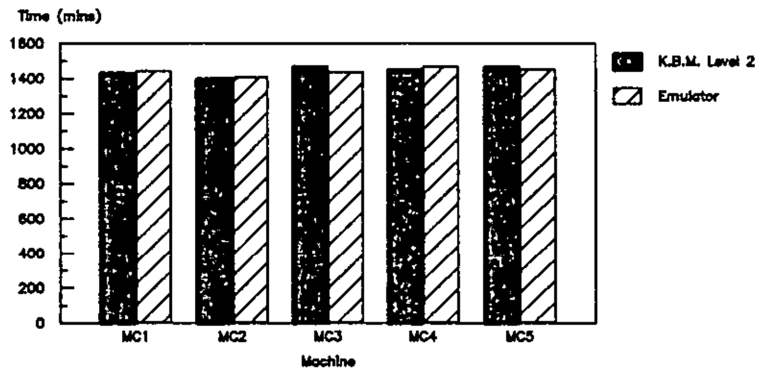


Figure 18.8	*Comparison of K.B.M. Level 2 with Emulator for Kit Production - Machine Performance Time*	**LUT - FMS** Research Group

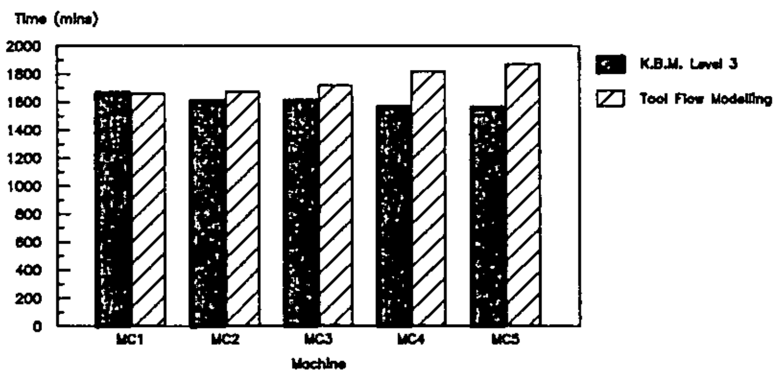


Figure 18.9	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - Machine Performance Time*	**LUT - FMS** Research Group

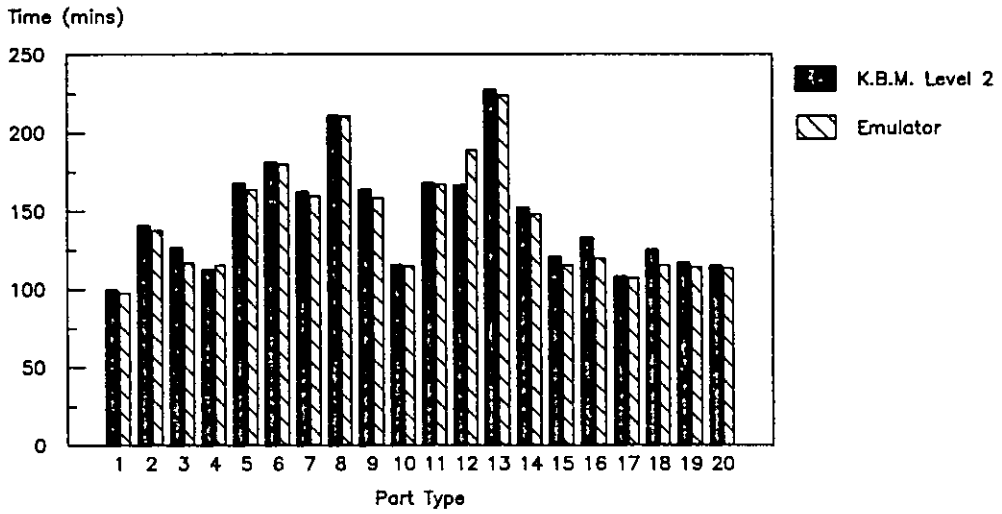


Figure 18.10	*Comparison of K.B.M. Level 2 with Emulator for Batch Production - Average Part Lead Times*	**LUT - FMS** Research Group

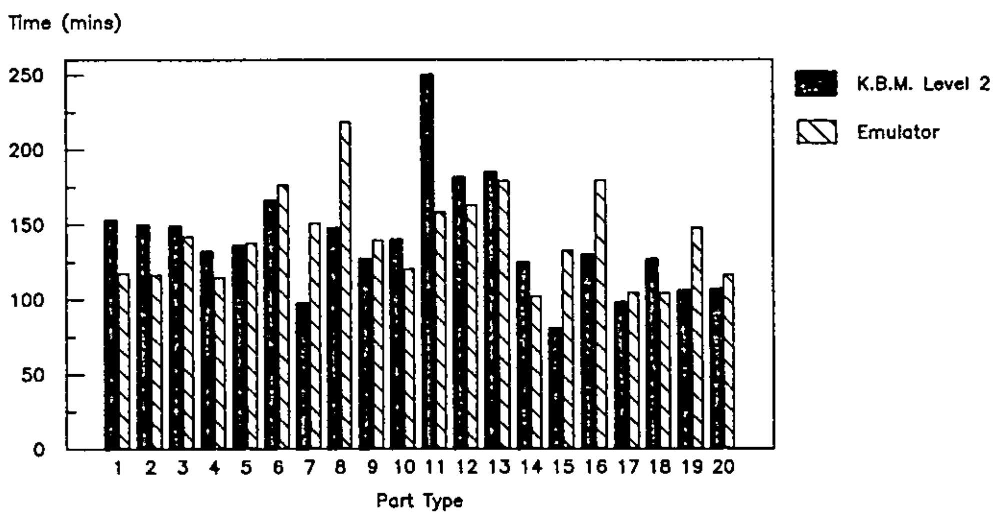


Figure 18.11	*Comparison of K.B.M. Level 2 with Emulator for Kit Production - Average Part Lead Times*	**LUT - FMS** Research Group

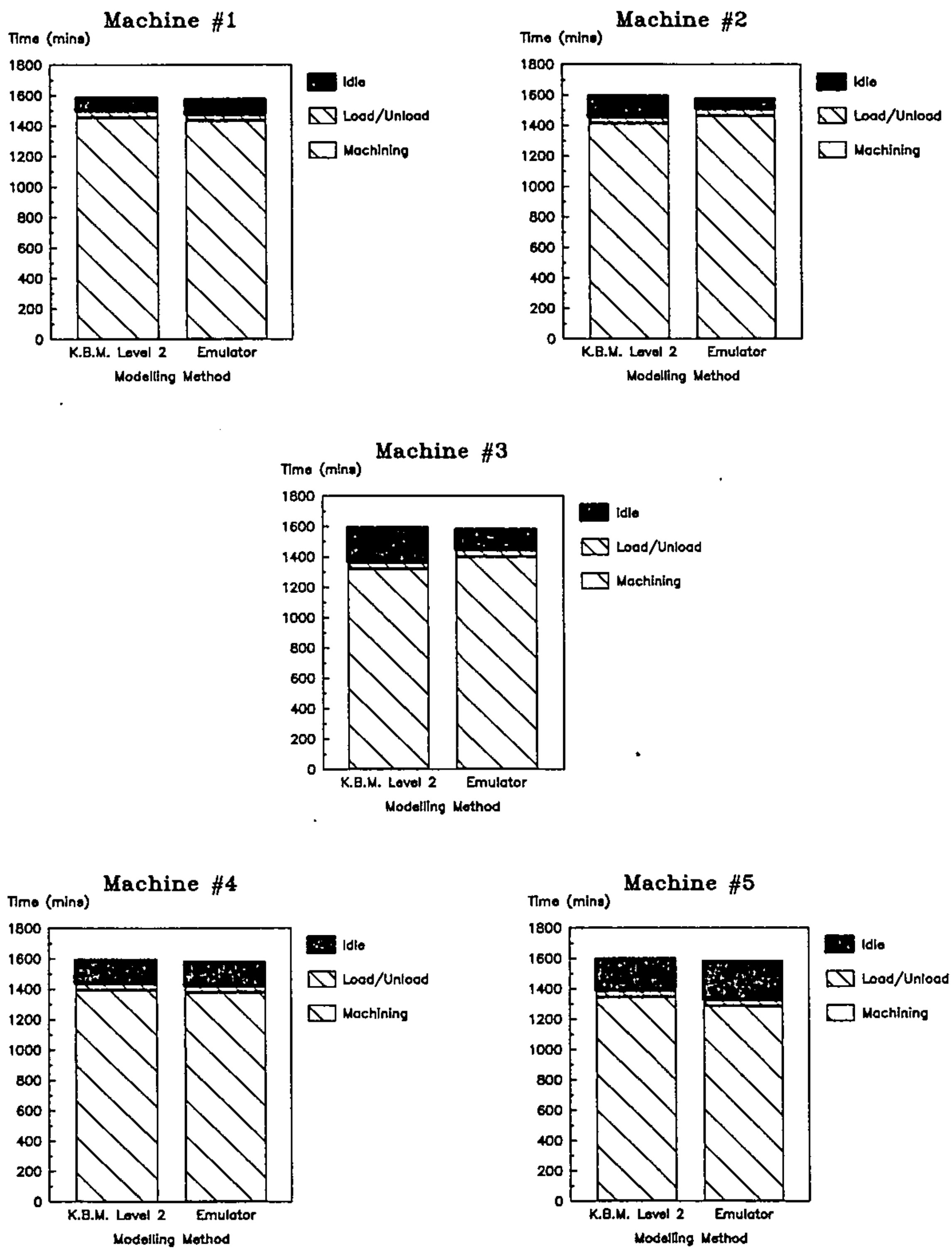


Figure 18.12	*Comparison of K.B.M. Level 2 with Emulator for Batch Production - Machine Performance*	LUT-FMS Research Group

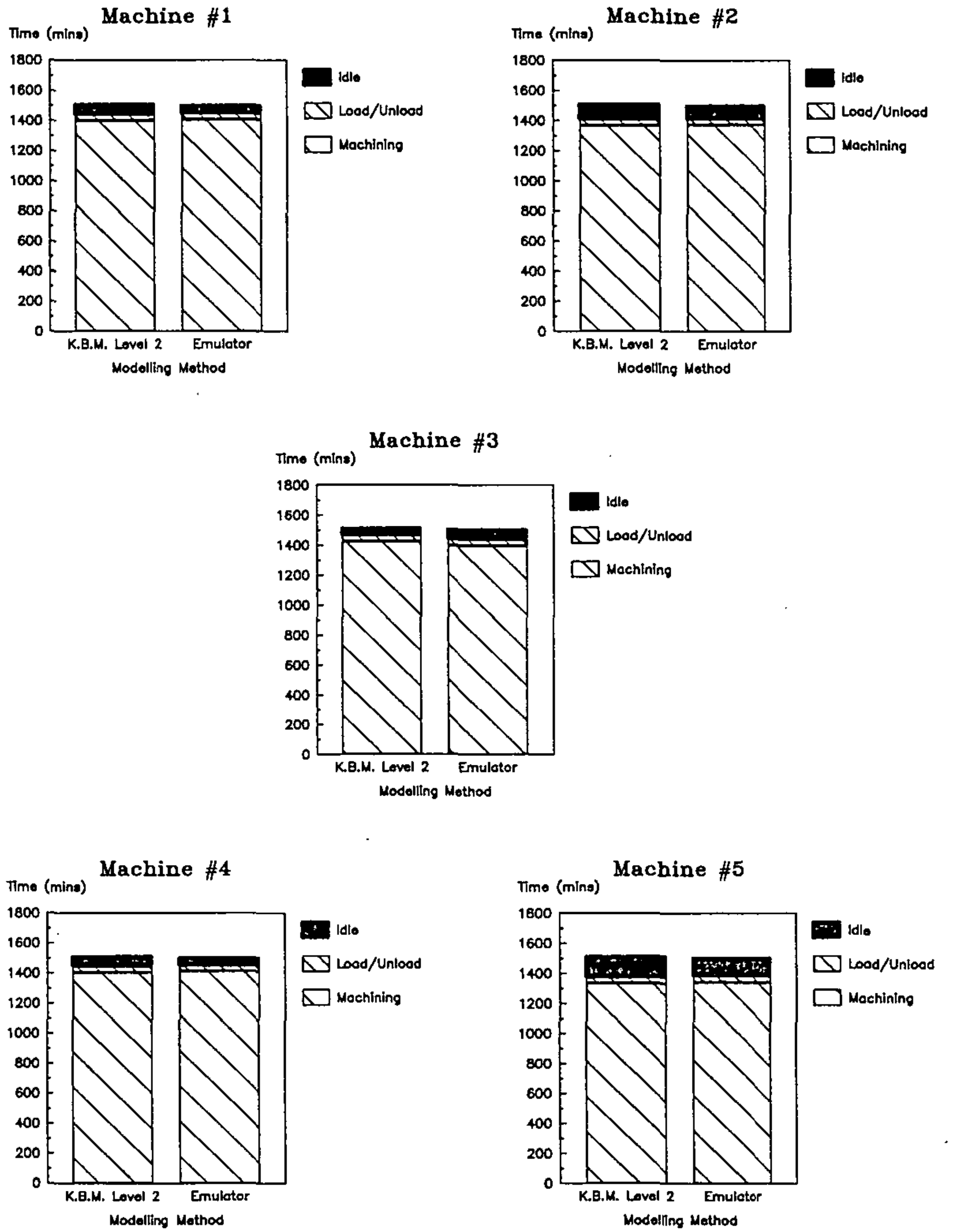


Figure 18.13	*Comparison of K.B.M. Level 2 with Emulator for Kit Production - Machine Performance*	LUT-FMS Research Group

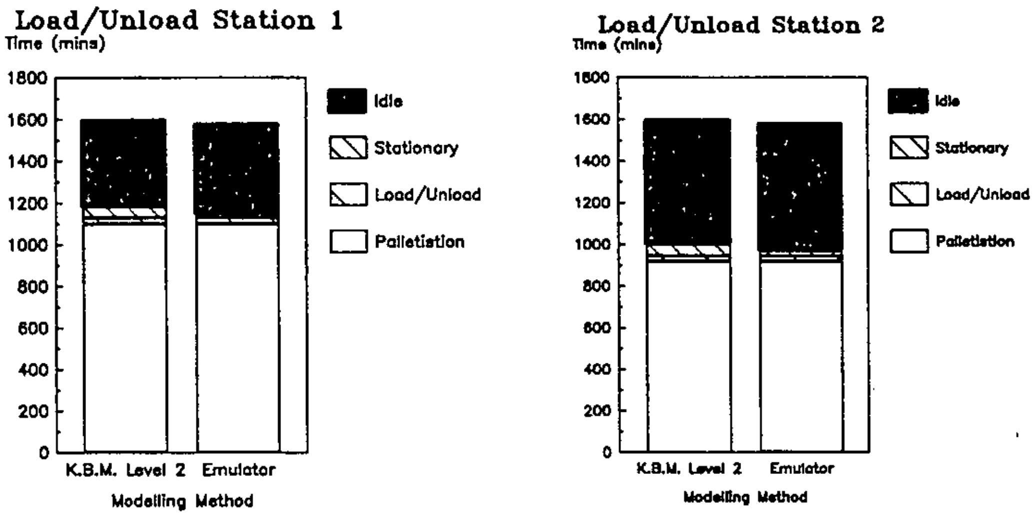


Figure 18.14	*Comparison of K.B.M. Level 2 with Emulator for Batch Production - Load/Unload Station Performance*	LUT - FMS **Research Group**

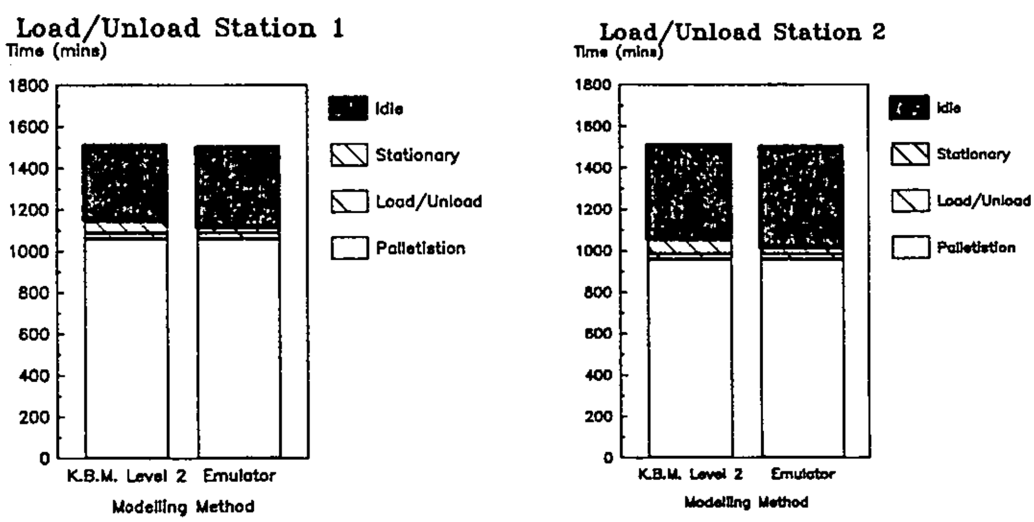


Figure 18.15	*Comparison of K.B.M. Level 2 with Emulator for Kit Production - Load/Unload Station Performance*	LUT - FMS **Research Group**

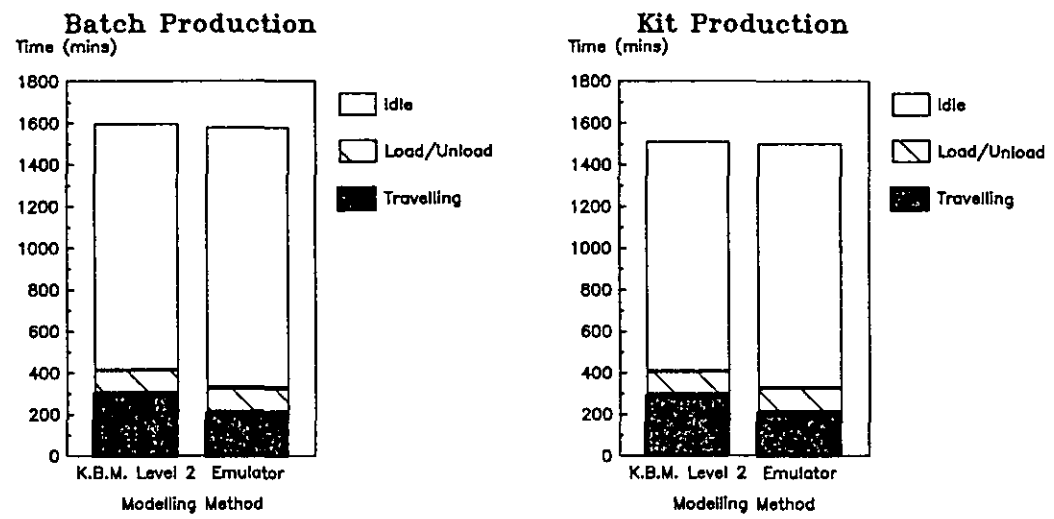


Figure 18.16	*Comparison of K.B.M. Level 2 with Emulator - Transporter Performance*	LUT - FMS Research Group

Part / Mach.		Job	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Batch Produc.	MC1	KB	1	1	2	3	5	6	9	12	13	13	13	13	14	14	16	16	17	19	19	20	20	
		EM	1	1	2	3	6	6	9	12	13	13	13	13	14	16	17	18	18	20	20	20		
Kit Produc.	MC1	KB	1	2	13	12	13	1	20	13	20	4	17	18	19	13	2	4	13	1	3	13	1	
		EM	1	2	13	16	13	10	9	1	19	12	4	14	6	19	20	15	13	1	13	18	2	

Figure 18.17	*Comparison of K.B.M. Level 2 with Emulator - Machine Work List (machine #1)*	LUT - FMS Research Group

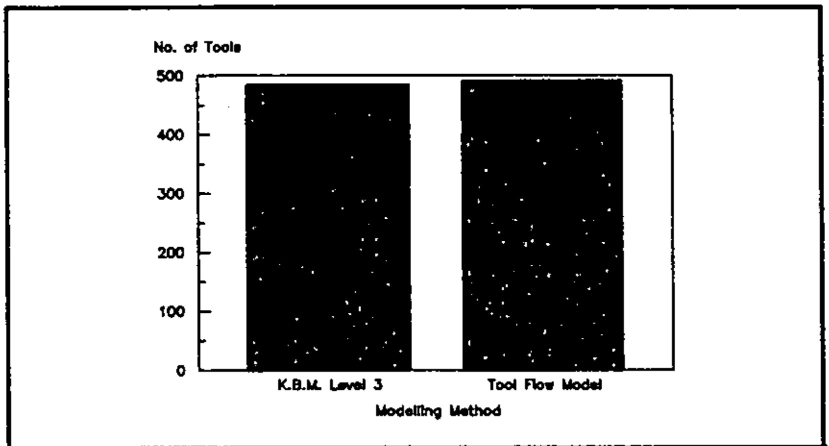


Figure 18.18	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - Cell Tool Requirement*	**LUT - FMS** Research Group

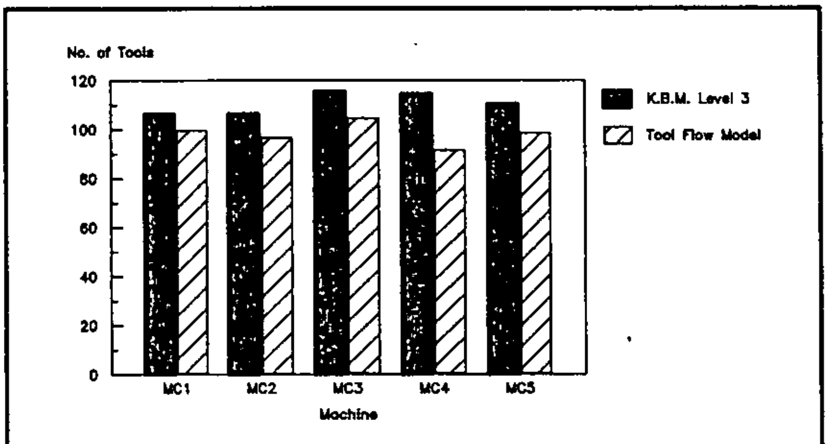


Figure 18.19	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - PTS Tool Utilization*	**LUT - FMS** Research Group

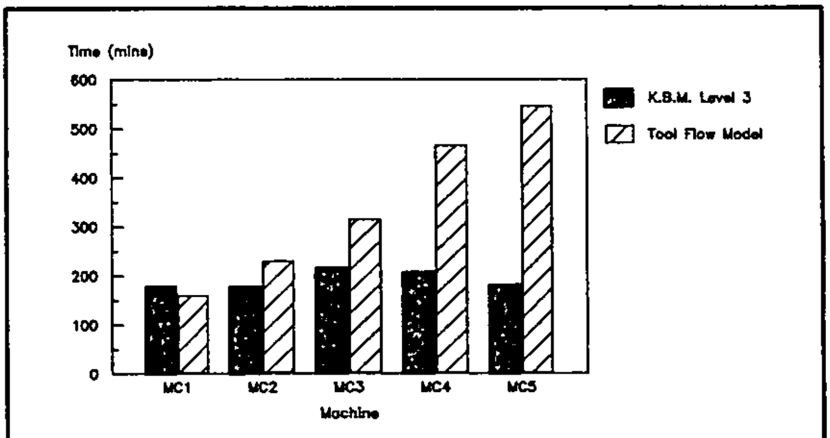


Figure 18.20	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - Cell Tool Wait Times*	**LUT - FMS** Research Group

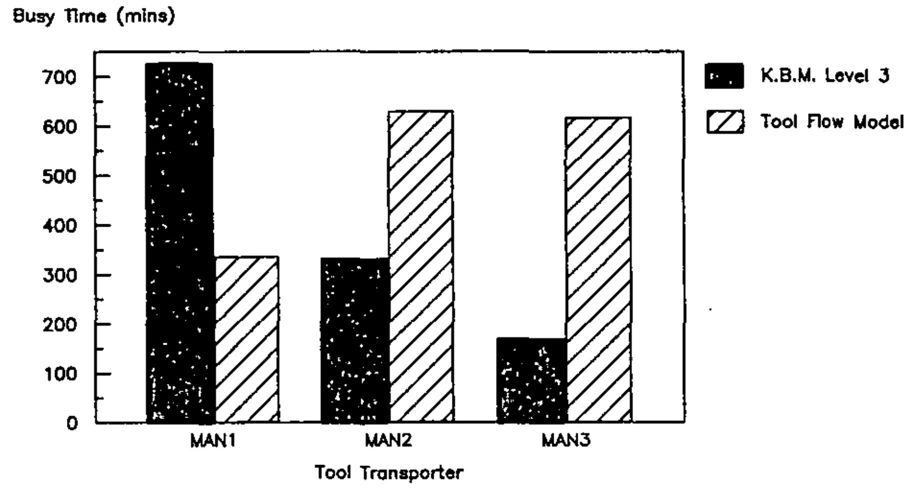


Figure 18.21

Comparison of K.B.M. Level 3 with Tool Flow Modelling - Tool Transporter Performance

LUT - FMS
Research Group

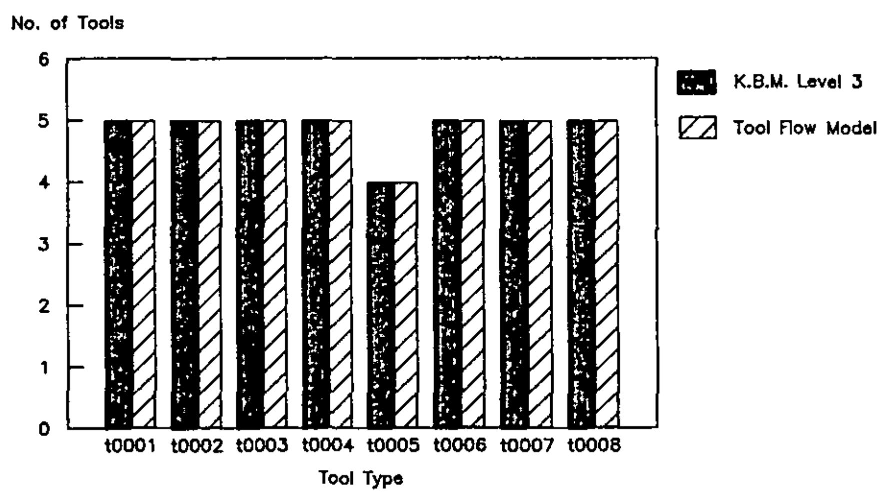


Figure 18.22

Comparison of K.B.M. Level 3 with Tool Flow Modelling - Sister Tool Prediction

LUT - FMS
Research Group

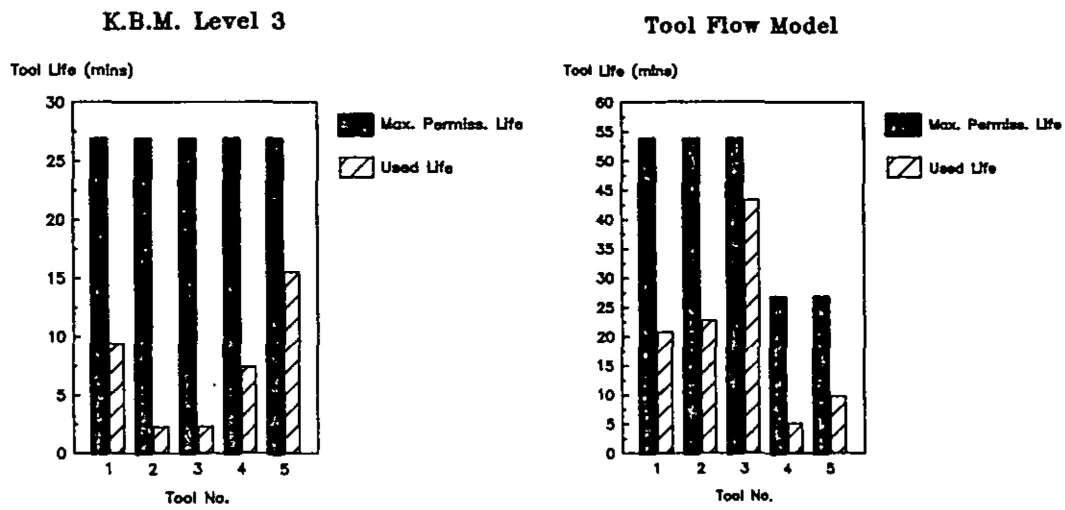


Figure 18.23	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - Tool Life Utilization*	LUT - FMS Research Group

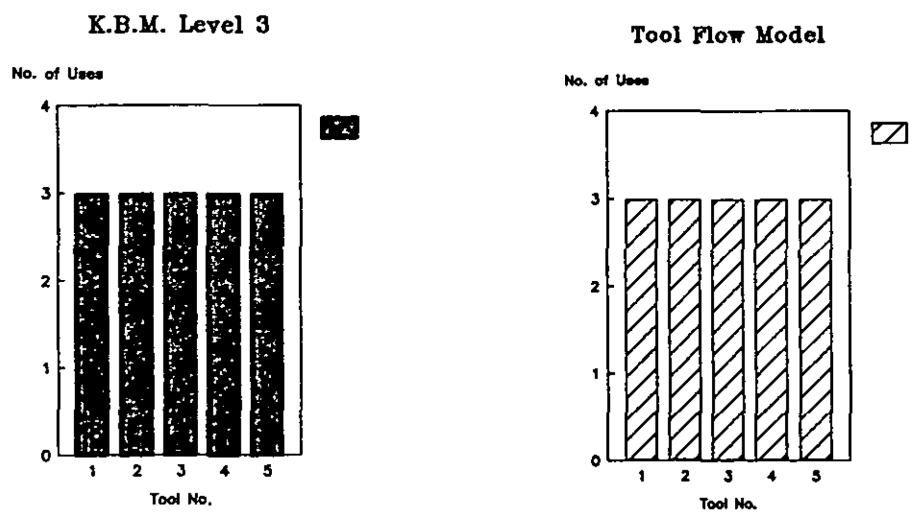


Figure 18.24	*Comparison of K.B.M. Level 3 with Tool Flow Modelling - Tool Usage Frequency*	LUT - FMS Research Group

Chapter 19
CONCLUDING DISCUSSION

19.1 Introduction

The discussion in this chapter centres on the major issues arising from the work reported in this thesis. In addition, the experience gained in the case studies is summarized, and the potential value of the modelling system are identified and discussed.

19.2 A Potentially Powerful Approach to Manufacturing System Modelling

The literature survey in Chapter 2 has indicated the emergence of a distinct trend towards the provision of powerful modelling tools for the design of advanced manufacturing systems. The study of various forms of flexible manufacturing facilities in Chapter 3 has illustrated their complex structure and unique control functions in contrast to conventional manufacturing technology. Recognition and discussion of the modelling challenge with regard to the design of these systems in Chapter 4 lead to the critical comparative assessment of the major currently available modelling methods, with conclusions being drawn that there is scope for improvement of these techniques. This has provided the impedus for the research into the application of modern AI techniques for the modelling these complex advanced manufacturing systems.

With the available Xerox Workstation and LOOPS package, a knowledge based multi-level approach has been developed in competition for the conventional modelling methods [165] [248] [254] [138] [84] [237] [93] [164] [43] [266]. The established modelling system has been designed to cover all the major aspects of the manufacturing system modelling process, Chapter 6. These include physical configuration of a model, collection and management of the associated data, entry of operational rules to govern the behaviour of the model, automatic setup of graphics and textual output facilities for model verification, and finally, although not implemented yet, automatic analysis of the results obtained from the modelling.

19.3 The Knowledge Based System Structure

As described in Chapter 7, the modelling system has been constructed using an integrated knowledge representation scheme within a knowledge based system structure. This is characterized by the separate design and logical integration of the inference engine, the working memory, the decision centre, the knowledge base and the data base, which are the typical elements of a knowledge system [234a] [44]. The resulting system structure then enables a more logic and adequate representation of a machining cell subject to the modelling requirements of the cell.

The inferencial process within the inference engine has been highly generalised by developing meta-rules so that it can also be used in the modelling of other discrete event systems, such as assembly cells. The construction of the working memory has made it possible for all the model related objects to be brought under a totally structured organization. Information retrieval and objects access, as functioned by the rules in the knowledge base, are then made flexible and efficient. The distinct design of the decision centre as a separate element in the working memory has enabled the structured generation and representation of various conflicts. This then allows the decision rules required in the operation of a machining cell to be more explicitly represented in the knowledge base, and thus alternative user-specific decision rules can be entered. This is differing from conventional knowledge systems where these conflicts tend to be handled by the inference engine or by the transformational rules. With conventional simulation techniques [209], the making of these decisions is usually embedded in the computer code leading to the difficulty for the user to enter his own rules.

The separation of the knowledge base from the inference engine allows the user with limited programming experience to be able to access and modify the existing rules or express his application-specific rules without affecting the functionality of the whole system. The design of the data base browser has separated the static data of a model from its dynamic data which is stored in the working memory. This then allows the user to flexibly specify and manage the static objects and process data associated with the model without worrying about the complex links between the dynamic objects. These dynamic objects are created automatically and stored in the working memory, which inherit the information from the corresponding static objects through their active value connections. This is in contrast to the technique used by STEM [32], where each object contains not only static information but also dynamic information, leading to the fact that the user has to be concerned with the links of the objects while building a model though it is of more generality.

19.4 The Four Level Control Strategy

The control within the knowledge based modelling system has been facilitated in a hierarchical way, with four basic levels being defined, i.e. the inference level, the decision level, the state transformation level and the state description level, Chapter 7. The inference level is the highest level, which is responsible for sequencing the actions and conflict resolutions which occur as the modelling proceeds. It controls not only the decision level but also the state transformation level. The decision level receives the conflicts arising at the transformation level, resolves these conflicts, detects and resolves further conflicts which can occur between the conflict-resolved actions, and finally sends the non-conflicting decisions to the working memory which are then used at the transformation level.

The transformation level is the set of statements which describe the actions that make up the model. These are the explicit rules about the interactions of the cell elements. The state description level is the set of rules used by the transformation level to model the details of the cell elemments. It consists of rules for accessing and changing those particular attributes of the cell elements, and for collecting statistics. It has been designed to consist of a number of sub-description levels, i.e. rules have been written which can be called by higher level rules.

The advantage of this hierarchical control strategy is that the functionality of different rules has been made explicit and the interactions between these rules have been clearly defined.

19.5 The Hierarchical Modelling Capability

In a machining cell, the flow of work or tools can be described as chains of actions associated with pallets and tool transporters, which include palletisation, load/unload station unloading; work transporter empty running, loading, load running and unloading; machine loading, tool change, cutting, machine unloading; load/unload station loading and depalletisation; tool transporter empty running, tool loading at STS, tool transfer to PTS, tool exchange at PTS, tool transfer to STS and tool unloading.

Depending on the modelling of the relationships between these actions, rules can be defined to model different combinations of these actions, thus creating the multiple levels of modelling detail, Chapter 8.

According to the decisions a manufacturing system designer has to make during the design process, three levels of modelling detail has been defined in this modelling system. At the first level, station is assumed to have an infinite local buffer so that no blockage could occur. Temporary storages and tool availability are ignored. Level 2 allows more system details to be modelled. Each station can have a specified buffer type and size. Part temporary storages, and tool requirments with regard to the whole cell or particular machines are considered. At level 3, in addtion to the features considered at level 2, tool flow within the cell can be modelled. Primary tool store, cell secondary tool store, tool transporters and tool flow strategies are all considered in the modelling.

Although only three levels are defined in the modelling system, it can be easily extended to include other levels, because the control of the rules at these multiple levels is integrated and the design of a new level can call to the description rules used by the other levels.

19.6 Data and Rule Driven Requirements

In a real manufacturing environment, the manufacturing knowledge comprises of both data and rules, but the latter tends to be frequently neglected by most of the existing modelling tools [43], leading to the inadequate representation of a manufacturing system with these tools. In the knowledge based modelling system, two levels of rules have been defined according to the manufacturing tasks performed in the operation of a machining cell, Chapter 8. Firstly decision rules are carefully defined which model the various conflicts arising from the sharing of resources by the components to be manufactured or from the assignment of alternative resources to the same components. Decision points have also been designed to allow the user to enter alternative rules. These decision rules include part release rules, load/unload station selection rules, next station selection rules, work transporter selection rules, pallet scheduling rules, pallet tooling rules, tool transporter selection rules. The second level of rules are the behavioural rules which include the transformation rules and description rules. Although expression of alternative transformation rules requires more LOOPS programming experience, entry of description rules to allow different behaviours can be facilitated by the user without affecting the logic of the defined modelling levels.

As a result of the consideration of different levels of manufacturing functions within each modelling level, three levels of data and rule input have been defined to allow the user to model a machining cell according to his detail needs, Chapter 8.

19.7 User-Friendly Interface

On of the main advantages of the modelling system described in this thesis is that the barrier between the user and the computer assisted modelling facility has been removed. This has been achieved by developing various facilities to allow the user with no or limited programming skills to quickly design, modify and experiment with a model by manipulating icons and menus, modifying structure parameters, and selecting and expressing cell operational rules. In addition, it provides visibility into structure, behaviour, and data collection for analysis through using enhanced graphics capabilities, Chapter 9.

Since the interactions among the system elements are represented using rules and rules can be traced during the running of a model, facilities have been developed to allow the relationships between the rules entered by the user and the immediate behaviour of the model to be made explicit. This transparent modelling process enables convenient and straightforward debugging and modifications of the model.

19.8 Output of Performance Figures

Corresponding to the three levels of input, three levels of default output can be provided from the modelling system, Chapter 10. For each level, three categories of outputs have been defined, i.e. the overall cell performance, the primary output and the secondary output. In addition, the user can also specify performance parameters he desires by entering rules which collect the required statistics of the system elements. This is extremely useful when additional aspects of the system need to be analyzed.

19.9 Case Study Experience

19.9.1 Selection of Modelling Levels

The study of the three machine cell, as presented in chapters 13, 14 and 15, has concluded that as a result of the influence of the operational assumptions made within each level, the performance of the cell provided by each level can vary, and the less the assumption, the more realistic the results. From levels 1 to 3, as more and more system operational details are considered, the performance provided by the modelling becomes decreasingly optimistic but increasingly realistic. Nevertheless, according to the different modelling needs, the three levels can be used at different stages during the design of a manufacturing system, Chapter 15.

19.9.2 Flexible Data Base Management

The experience gained through the case study work has shown that gathering of data using the data base browsers of the knowledge based modelling system is fairly flexible and efficient. The representation of each cell element as an instance object helps to collect the data associated with all attributes of the element, Chapter 14 and Chapter 16. In particular, the addition, deleting or editing of a cell element can be carried out with ease without affecting the others. Therefore the data base browsers provede a flexible environment for the design of and experiment with alternative models.

19.9.3 Entry of Alternative Decision Rules

From the study of the extended cell as given in chapters 16 and 17, entry of alternative decision rules into the modelling system for paticular applications is a fairly convenient process. For the user can express a new rule by referring to the structure of the existing rules, or by writting a entirely new rule which access the desired variables of the associated objects.

As indicated by the case studies, the combination of different decision rules can have a significant influence on the cell performance. In particular, the release rules, the pallet priority rules and the station selection rules seriously affect the work flow patterns and thus the performance of the modelled cell. Therefore, in order to model a cell realistically, these rules must be carefully chosen or developed.

19.9.4 Comparison with Other Modelling Methods

In comparison with the Emulator and the tool flow modelling system, the knowledge based modelling system can provide considerable outputs with regard to the major structural and operational aspects of a flexible machining facility. Particularly, the exploration of the modelling of dual work/tool flow has provided some important results in regard to the interactions between work and tool flow within the cell.

Based on the comparison of the results obtained from the three methods, it can be concluded that the operation of level 2 is compatible with that of the Emulator, and the modelling of tool flow at level 3 is similar to that of the tool flow modelling system, Chapter 18.

19.10 The Potential Value of the Modelling System

As a result of the well-structured organisation of the modelling knowledge extracted from the field of flexible machining, this modelling system can be conveniently extended along the following directions:

- More detailed modelling of complex behaviours,
- Extension to multi-cell system modelling,
- Analysis of results,
- Enhancement of the user interface.

19.10.1 More Detailed Modelling of Complex Behaviours

Since this work has established a framework for single-cell system modelling, modelling of complex behaviours at a more detailed level can easily be achieved by specialising the objects or behavioural rules which have been built into the modelling system.

For example, if the performance of a local buffer is going to be adequately modelled, a class called, say, Buffer can be specialised from Entities. The user is then required to edit the class by adding a few instance variables, such as StationaryTime, LoadingUnloadingTime and IdleTime, etc., After this , the user can create necessary instances and set them to the instance variable, such as Buffer, of specific stations. Finally, the user is required to specialise the rulesets, such as *LoadToStation, LoadOntoStation, LoadOntoTransporter and CheckBuffer,* which affect the behaviour of the buffers, by changing the *PalletsInBuffer* instance variable of a station to the Contents of the buffer of that station. Besides, rules for collecting the statistics of buffers need to be created and added to the above rulesets.

Other aspects of a system, which can be modelled in more detail, include among others the routing interference of transporters [43], the management of tool magazines [84] and other tool issue strategies [181]. All these aspects can readily be implemented within the established modelling system structure.

19.10.2 Extension to Multi-Cell System Modelling

As for the modelling of large-scale systems, such as multi- cell FMS [129], or large-scale DNC systems [28], efficiency can be of particular significance, since any detailed modelling may be inhibited by the long run times.

A suggested modelling method for these type of workstations is to highly abstract each cell by extracting the most important behavioural properties of the cell, and then to model the interactions between cells. Abstraction of the performance of the single cells can be achieved by running the established cell modelling system.

By doing this, the most significant performance aspects of a multi-cell system cam be evaluated. Depending on the level of abstraction of single cells, a multi-cell system can also be modelled and assessed over various levels of detail. An AI technique which might match this modelling requirement is the hierarchical planning method [274].

Although at its early stage, the modelling of multi-cell systems within the Emulator is on the way by a complementary project [1].

19.10.3 Analysis of Results

As an important issue in the modelling of manufacturing systems, analysis of the modelling results with respect to the input parameters and strategies has to be faclitated. An potentially useful technique in developing this analysis procedure is the AI learning technique [77] [111].

An important aspect of learning that can be explored, albeit in a rather limited sense, is to add some reasoning mechanism to the modelling system. Given the performance specifications, this reasoning mechanism analyses the results from a run, automatically adjust input parameters of the model, run the model again, and finally determine, in some sense, a 'best' design (Figure 6.14).

Modelling system coupled with this type of learning will then become a prescriptive tool or an intelligent automatic design tool. It can be foreseen that the expertise involved in this learning mechanism is considerable, and quite a lot has not been well-established. Two techniques are available which can be applied to implement this type of learning. One is perturbation analysis [132], and the other is parameter adjustment [111].

Another aspect of learning that can be implemented with practical results is rule composition. With such a mechanism, the modelling system automatically create a new rule that summarises the behaviour of two or more rules that are executed in sequence. By implementing rule composition in knowledge based modelling, the execution of the program will improve, and further, the analyst may discover possible redundancies in a model. This is particularly valuable for the modelling of large-scale systems.

19.10.4 Enhancement of the User Interface

As described in early chapters, three independent data base browsers are used corresponding to the three levels of modelling. In most cases, all or part of the information used at a particular level can be shared by the other levels. Thus, there is a need for developing facilities which can enable the trasfer of information among different levels. If this is achieved, the specification of manufacturing data with the modelling system will be more convenient.

At this stage, entry of rules requires limited knowledge of LOOPS. A possible enhancement can be that a menu, which contains all the objects accessible by the work space class of the rule, is attached to the LOOPS rule editor [53] so that the expression of the rule is under direct guidance.

Aother aspect that needs to be improved is that with more details to be incorporated in the modelling system, the animation of the operation of the model on the screen becomes necessary. For this will further help the user to understand the behaviour of the designed model or to verify the logical operation of the model.

At this stage, the presentation of modelling results is of a text file form. Further enhancement can include the provision of dispaly graphics, such as bar charts, line graphics, histograms, scatter diagrams, pie charts, etc. [242]. These facilities are helpful in analyzing and communicating results.

Chapter 20
CONCLUSIONS

20.1 Introduction

This chapter summarizes the specific conclusions which can be drawn from the research work and the case studies as described in early chapters.

20.2 The Knowledge System Framework

As a result of the separate design of the inference engine, the working memory, the knowledge base and the data base browser, the modelling system allows access and modification of any of these elements without affecting the others. This is considerably in contrast to conventional modelling tools, where specific knowledge tends to be integrated with control functions.

20.3 Methodology for Model Configuration

The use of icons for displaying a manufacturing system on the screen enables highly flexible and interactive definition of the physical structure of the system. In addition to the icons which have been created, the user can add new ones to the library by drawing the new icon and specifying the defining class.

20.4 The Data Base Browser

Since a manufacturing facility usually involve enormous data, and thus the number of objects created to represent the system elements and to store the associated data can be huge, the hierarchical organization of these objects, as used by the data base browser, has proven to be an effective method. This highly interactive menu driven software enables flexible and efficient management of machining cell data.

20.5 Methodology for Rule Entry

The provision of rule libraries and the interactive facilities for rule expression for defined decision points has allowed the user to conveniently enter his own desired decision rules in order to control the performance of the designed model. Besides, the organization of all the behavioural rules in the rule base has made it possible for the user to inspect or modify these rules in order to design alternative modelling levels.

20.6 Graphics Output Facilities

The split-screen technique, as used in the development of the graphics output facilities, has shown to be a useful verification tool by speeding the process of locating and removing errors in the model. In addition, attaching of monitors to specific cell elements allows certain performance attributes to be dynamically displayed.

20.7 The Textual Output Facility

The design of the textual output facility has enabled the trace of the rules applied on a specific instance object. This capability further helps the user in debugging a model by relating the computer code to specific behaviour of the object or cell element.

20.8 Influence of Modelling Assumptions on Results

The results of the case studies have shown that under certain conditions the assumptions made in a model can considerably influence the results obtained, and the less the assumption, the more realistic the performance provided. Nevertheless this influence can be reduced by developing special operational rules.

20.9 Application of Level 1

Fron the case study experience, the level 1 is well suited to the basic sizing of a manufacturing system in the early design stage. It is best used to help determine the number of equipments, the rough size of station local buffers and overall control strategies. Additionally, it can be used for long-term manufacturing performance forcasts.

20.10 Application of Level 2

After the basic size and operational strategies have been determined, the level 2 can be selected to study the detailed work flow behaviour of the system. In particular, the various types and sizes of local buffers and temporary storages can be considered. Besides, the tool requirements planning can be conducted in order to help choose the preliminary tool management strategies.

20.11 Application of Level 3

For those systems where tool flow management is a crucial iisue, level 3 should be selected in order to study the tool flow activities between tool stores and the associated tool provision strategies.

20.12 Influence of Work and/or Tool Flow on System Performance

From the comparison of the results obtained from the level 3 modelling and the modelling of the extended cell, it can be concluded that the influence of work and/or tool flow on the system performance can take three basic forms: i.e. system performance dictated by work flow, by tool flow, or by both work and tool flow.

20.13 Influence of Cell Parameters on System Performance

The experience gained in the case studies has shown that cell parameters, such the number of machines, the number of pallets, the PTS capacity and the temporary storages, can have a significant effect on the cell performance. Thus, they should be carefully specified when designing a manufacturing system.

20.14 Influence of Station Pooling on System Performance

As shown in both case studies, the grouping of stations and the assignment of parts to these gorups seriously affect the system performance. It influences not only the workload balancing across the stations in a group, but also the work flow patterns of the components.

20.15 Influence of Decision Rules on System Performance

From the study of the extended cell, the influence of different decision rules on the system performance is also of significance. In particular, the part release rules, the pallet priority rules and the station selection rules can bring condiderably different results subject to the change of these rules. Therefore, these rules have to be carefully developed when applied to particular industrial cases.

20.16 Use of the Modelling Output

By selective change of the input variables and rules, and study of the resulting model output, particular aspects of a flexible machining cell behaviour and the suitability of the cell configuration for specified production requirements can be investigated, with different operational strategies being evaluated.

20.17 Application of Xerox Workstation and LOOPS Package in Manufacturing System Modelling

The Xerox 1186 Workstation and the LOOPS knowledge engineering language, as used in the development of this work, have proven to be effective facilities for the modelling of manufacturing environment. The Xerox Workstation provides a very powerful programming environment for creating and debugging knowledge systems. The LOOPS package, on the other hand, has integrated four programming paradigms and enables representation of different types of manufacturing knowledge using appropriate programming schemes.

Chapter 21
FURTHER WORK

21.1 Introduction

It is recommended that further work be carried out in the following areas, based on the concluding discussion in Chapter 19 and the specific conclusions reached in Chapter 20.

21.2 More Detailed Modelling of Wok and Tool Flow

To minimise the difference between the design performance and the actual performance of a manufacturing facility, modelling of the routing interference of transporters has to be considered. To further intestigate the integration of work and tool flow, other tool issue strategies should be modelled, for example the kitting and single tool strategies. Besides, the single function and shared work/tool flow networks need to be modelled.

21.3 Extension to the Modelling of Assembly Operations

Although flexible machining has been the most essential feature of advanced manufacturing technology, recent development in flexible assembly calls for powerful modelling tools in order to assess its performance. Therefore, extension of the modelling system to include the modelling of assembly operations can be a very useful feature to industry. Because of the diversity of assembly installations, it is expected that the representations for these facilities should be more generic than those for machining.

21.4 Extension to Multi-Cell System Modelling

In reflection of the current trend in the development of flexible manufacturing facilities, modelling of multi-cell machining systems is a very useful extension. In particular, if the links between machining and assembly can be studied, the value of the modelling system to industry will be more significant.

21.5 Analysis of Results

As mentioned in the main text of this thesis, analysis of results is another vitally important issue in providing a complete solution to manufacturing system modelling. This analysis procedure can be faciltated by building a front-end to the existing modelling system usinf perturbation analysis or AI parameter adjustment learning techniques.

21.6 Enhancement of the User Interface

To make the modelling system more valuable to industry, there is scope for improving the user-friendliness of the existing interface. Major aspects of the improvement can include the animation of the flow of icons in the model configurator, development of facilities for guiding the user's rule entry, provision of static display graphics for presenting modelling results, and enhancement of the capability of the data base browser.

REFERENCES

1. Adams, J.
Loughborough University, Private Communication, 1989.

2. Abdin, M. F. and N. S. Mohamed
"The role of simulation in design of FMSs"
Proc. 8th Annual Conf. on Computers and Industrial Engineering, 1986, pp.372-376.

3. Addis, T. R.
Designing Knowledge-Based Systems
Kogan Page, London, 1985.

4. Adelsberger, H. H., et al.
"Rule based object oriented simulation systems"
Intelligent Simulation Environment (Luker, P. and H. Adelsberger, Eds.), Simulation Series, Vol.17, No.1, Jan. 1986, pp.107-112.

5. Afentakis, P.
"A model for layout design in FMS"
Flexible Manufacturing Systems: Methods and Studies (A.Kusiak, Ed.), Elsevier Science Publishers, North-Holland, 1986.

6. Afentakis, P., R.A.Millen and M.M.Solomon
"Layout design for flexible manufacturing systems: Models and strategies"
Proc. 2nd ORSA/TIMS Conf. on FMS : Operations Research Models and Applications (K.E.Stecke and R.Suri, Eds), Elsevier Science Publishers, North-Holland, 1986.

7. Alam, M. D. Gupta, S. I. Ahmad and A. Raouf
"Performance modelling and evaluation of flexible manufacturing systems using Semi-Markov approach"
Flexible Manufacturing (A. Raouf and S. I. Ahmad, Eds.), Elsevier, Amsterdam, 1985.

8. Alla, H. and P. Ladet
"Coloured Petri nets: a tool for modelling, validation and simulation of FMS"
Flexible Manufacturing Systems: Methods and Studies (A. Kusiak, Ed.),
North-Holland, 1986.

9. Allen, J.
"Maintaining knowledge about temporal intervals"
Communications of the ACM, Vol. 26, No. 11, 1983.

10. [redacted]
"Process planning problems? First see how with SEE-WHY"
The Production Engineer, Vol.60, No.12, 1981, pp.14-18

11. [redacted]
"Fanuc's robot production line"
Metalworking Engineering and Marketing, Sept. 1981, pp.166- 168.

12. [redacted]
"Lheon automatic's 'TIPROS'"
Metalworking Engineering and Marketing, Sept. 1981, pp.170- 172.

13. [redacted]
"Yamazaki machining works 'FMS'"
Metal Working Engineering and Marketing, July 1982, pp.94-98.

14. [redacted]
"How Lucas Bryce approaches flexible unmanned machining"
Production Engineer, Oct. 1982, pp.13-14.

15. [redacted]
KT's World of Advanced Manufacturing Technologies, 2nd Edition, Kearney and Trecker Corporation, Milwankee, WI, USA, 1983.

16. [redacted]
"Scamp is unveiled"
The FMS Magazine, Vol. 1, Jan. 1983, pp.76-80.

17. Anon.
"Rolls Royce installs multi-robot grinding"
Industrial Robot, Vol. 10, March 1983.

18. Anon.
"How computer simulation helped Citroen set up new Paris flexible manufacturing system"
The Production Engineer, May 1983, pp.18-19.

19. Anon.
"Output is tripled by 101 robots"
Metalworking Engineering and Marketing, July 1983, pp.86-88.

20. Anon.
"KTM - Practising what is preaches with Brighton FMS"
The Production Engineer, July/August 1983, pp.24-25.

21. Anon.
"A challenge to the equipment industrialisation"
Metalworking Engineering and Marketing, Nov. 1983, pp.70-74.

22. Anon.
"FA model plant leads world's advanced technology"
Metalworking Engineering and Marketing, Nov. 1983, pp.66-69.

23. Anon.
"Combined CNC lathe with automatic chuck jaw changer and robot from Mazak"
Metal Working Engineering and Marketing, Jan.1984, pp54-57.

24. Anon.
"What is FMS? - Basic patterns and ideas"
Metal Working Engineering and Marketing, Jan. 1984, pp.46-52.

25. Anon.
"Cincinnati Milacron's formula for FMS"

The FMS Magazine, Jan. 1985, pp.11-14.

26. Anon.
"Max combines flexibility and productivity"
The FMS Magazine, Jan. 1985, pp.26-28.

27. Anon.
"Discussion on AI guidance in modelling and simulation"
AI Applied to Simulation (E. Kerckhoffs, et al., Eds.), Simulation Series, Vol.18/1, 1985, pp.201-202.

28. Anon.
"Strong Yen boosts automated factory"
Automation, July 1987, pp.47-50.

29. Anstiss, P.
"A new concept in the control of manufacturing systems"
Proc. 5th FMS, 1986, pp.201-214.

30. A.P.
"Yamazaki ends FMS speculation"
Production Engineer, Jan. 1986, p.3.

31. Arnaud, O. and E. Bloche
"Evaluating FMS by simulation"
Proc. of 5th Int. Conf. on FMS, 1986, pp.513-534.

32. Artificial Intelligence Ltd.
STEM (Reference Manual), Artificial Intelligence Ltd., Watford, England, 1988.

33. Astrop, A.
"£6 million 'model' balance unit line"
Machinery and Production Engineering, Feb. 1984, pp.31-33.

34. Azadivar, F. and Y. Lee
"Optimum number of buffer spaces in flexible manufacturing systems"

Proc. 2nd ORSA/TIMS Conf. on FMS: Operations Research Models and Applications (K. E. Stecke and R. Suri, Eds.), 1986.

35. Barash, M. M. et al.
"The optimal planning of computerised manufacturing systems"
Report No.1, Purdue University, School of Industrial Engineering, Nov. 1975.

36. Bard, Y.
"Some extensions to multiclass queueing network analysis"
Performance of Computer Systems (M. Arato, Ed.), North- Holland, Amsterdam, 1979.

37. Barr, A. and E. Feigenbaum
The Handbook of Artificial Intelligence, Vol.1, Kaufman Publishers, Los Altos, CA, 1981.

38. Barr, A. and E. Feigenbaum
The Handbook of Artificial Intelligence, Vol.2, Kaufman Publishers, Los Altos, CA, 1982.

39. Baskett, F. et al.
"Open, closed and mixed networks of queues with different classes of customers".
J. of Assoc. of Computing Machinery, Vol.22, No.2, pp.248-260.

40. Bell, R.
"Flexible automation in batch manufacturing"
2nd Int. Metal Cutting Conf., Wuhan, China, 1985.

41. Bell, R. and N. Bilalis
"Loading and control strategies for an FMS for Rotational Parts"
Proc. 1st Int. Conf. on Flexible Manufacturing Systems
IFS Publications, Oct. 1982.

42. Bell, R. and B. De Souza
"Tool flow management in flexible machining installations"
Proc. of 2nd Int. Conf. on Computer-Aided Production Engineering,

Edinburgh, April 1986, pp.233-238.

43. Bell, R., E. A. Roberts, N. Shires, S. T. Newman and R. Khanolkar
"A system for the design and evaluation of highly automated batch manufacturing systems"
Institute of Mechanical Engineers, 1984.

44. Ben-Arieh, D.
"Knowledge based control system for automated production and assembly"
Modelling and Design of Flexible Manufacturing Systems (A. Kusiak, Ed.), Elsevier, Amsterdam, 1986.

45. Ben-Arieh, D.
"A knowledge based system for simulation and control of a CIM"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, Chicago, USA, 1986, pp.13-21. IES Pubs.

46. Bernemann, S., B. Hellingrath and J. Joemann
"How and where can AI contribute to simulation"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, 1986, IES Pubs.

47. Bessant, J. and W. Haywood
"Experiences with FMS in the UK"
Proc. of the UK Operations Management Association Conf., 1986, pp.3-17.

48. Bevans, J. P.
"First choose your FMS simulator"
American Machinist, May 1982, pp.143-145.

49. Bilalis, N.
The Design and Control of FMS for Rotational Parts
Ph.D Thesis, Loughborough University of Technology, 1983.

50. Bill, E. G.
Tool Management - Determination of the Tooling Requirement for Prismatic Parts, M.Sc. Thesis, Loughborough University, 1984.

51. Blackstone,JR, J. H., D. T. Phillips and G. L. Hogg
"A state of the art survey of dispatching rules for manufacturing job shop operations"
Int. J. Prod. Res., Vol.20, No.1, 1982, pp.27-45.

52. Bobbio, A. and G. S. Aira
"Modelling automated production systems by deterministic Petri nets"
SIM-3, 1987, pp.127-136.

53. Bobrow, D. G. and M. Stefik
The LOOPS Manual, Xerox Corporation, Dec. 1983.

54. Briggs, S.
The design and Analysis of Alternative Systems for the Production of Fittings at APV International Ltd., Department of Manufacturing Engineering, Loughborough University of Technology.

55. Browne, J. and K. Rathmill
"The use of simulation modelling as a design tool for FMS"
Proc. 2nd Int. Conf. on FMS, London, 1983, pp.197-214.

56. Browston, L., et al.
Programming Expert Systems in OPS5 - Introduction to Rule- Based Programming, Addison-Wesley, Reading, Mass., 1985.

57. Butcher, M.
"Advanced integrated manufacturing system (AIMS) for aero engine turbine and compressor discs"
Proc. of 5th FMS, 1986, pp.93-104.

58. Buzacott, J. A.
"'Optimal' operating rules for automated manufacturing systems"
IEEE Transactions on Automatic Control, Vol.AC-27, No.1, 1982. pp.80-86.

59. Buzacott, J. A. and J. G. Shanthikumar
"Models for understanding flexible manufacturing systems"
AIIE Transactions, Vol.12, No.4, December 1980.

60. Buzen, J. P.
"Computational algorithms for closed queueing networks with exponential servers"
Communication of ACM, Vol.16, No.9, pp.527-531.

61. CACI Ltd.
"A quick look at SIMFACTORY with animation"
Publicity Document, CACI Ltd., Surrey, England, 1987.

62. Canuto, E., G. Menga and G. Bruno
"Analysis of flexible manufacturing systems"
Efficiency of Manufacturing Systems (B. Wilson, C. Berg and D. French, Eds.), pp.189-201, Plenum Press.

63. Carrie, A.
"Work scheduling in FMS under tool availability constraints"
Nat. Conf. on Prod. Res., Nottingham, Sept. 1985.

64. Carrie, A.
"FMS simulation: Needs, experience and facilities"
Proc. 1st Int. Conf. on Simulation in Manufacturing, Stratford, March 1985.

65. Carrie, A.
Simulation of Manufacturing Systems, Wiley, 1988.

66. Carrie, A., E. Adhami, A. Stephens and I. Murdoch
"Introducing a flexible manufacturing system"
Int. J. Prod. Res., Vol.22, 1984.

67. Cavaille, J. B. and D. Dubois
"Heuristic methods based on mean value analysis for flexible manufacturing systems performance evaluation"
Proc. 21st IEEE Conf. on Decision and Control, Orlando, 1982, pp.1061-1065.

68. Chan, W. W. and K. Rathmill

"Digital simulation of a proposed manufacturing system"
Proc. 19th MTDR Conf., UMIST, 1978.

69. Chan, W. W. and K. Rathmill
"An integrated FMS design procedure"
Annals of Operations Research, 3, 1985, pp.207-237.

70. Chandy, K. M., V. Hertzog and L. Woo
"Approximate analysis of general queueing networks"
IBM Research Report RC 4931, July 1974.

71. The Charles Stark Draper Laboratory Inc.
Flexible Manufacturing Systems Handbook, Noyes Publications, 1984.

72. Charniak, E. and D. McDermott
Introduction to Artificial Intelligence
Addison-Wesley, USA, 1985.

73. Charniak, E., C. K. Riesbeck and D. McDermott
Artificial Intelligence Programming, Erlbaum, Hillsdale, NJ, 1980.

74. Clementson, A. J.
ECSL User's Manual, CLE-COM Ltd., Birmingham, 1982.

75. Clocksin, W. and C. Mellish
Programming in Prolog, Springer-Verlag, New York, 1981.

76. Cohen, G., D. Dubois, J. P. Quadrat and M. Viot
"A linear-system-theoretic view of discrete-event processes"
Proc. 22nd IEEE Conf. on Decision and Control, 1983.

77. Cohen, P. R. and E. Feigenbaum
The Handbook of Artificial Intelligence, Vol.3, Kaufman Publishers, Los Altos, CA, 1982.

78. Cowan, D.
"Howden has first retrofitted flexible machining cell"

The FMS Magazine, Jan. 1986, pp.21-23.

79. Crookes, J. G., D. W. Balmer, S. T. Chew and R. J. Paul
"A three-phase simulation system written in Pascal"
J. Opl. Res. Soc., Vol.37, No.6, 1986, pp.603-618.

80. Dallery, Y. and R. David
"A new approach based on operational analysis for flexible manufacturing systems performance evaluation"
Proc. 22nd IEEE Conf. on Decision and Control, 1983.

81. Dallery, Y. and D. D. Yao
"Modelling a system of flexible manufacturing cells"
Modelling and Design of Flexible Manufacturing Systems (Kusiak, A., Ed.), 1986, pp.289-300, Elsevier.

82. Davey, S. G.
"A flexible manufacturing system (FMS) for PCB production"
Proc. 5th FMS, 1986, pp.29-40, IFS Pubs, Stratford.

83. Davis, R.
"Representation as a basis for system construction and maintenance"
Pattern-Directed Inference Systems (D. A. Waterman and F. Hayes-Roth, Eds.), Academic Press, 1978.

84. De Souza, B.
The Management of Tool Fow in Highly Automated Batch Manufacturing Systems, Ph.D Thesis, Loughborough University, 1988.

85. Dee, Z., H. Co and R. Wysk
"SIM-Q: A simplified approach to simulation modelling of automated manufacturing systems"
Proc. 2nd ORSA/TIMS Conf. on FMS, 1986, Elsevier, Amsterdam.

86. Denning, P. J. and J. P. Buzen
"The operational analysis of queueing network models"
Computer Survey, Vol.10, No.3, 1978, pp.225-260.

87. Diehl, G. W. and R. Suri
"Efficient analysis of blocking in finite buffer queueing systems"
Proc. of Performance 83, 1982.

88. Diehl, G. W. and R. Suri
"Queueing models for efficient analysis of tool sharing and blocking in automated manufacturing systems"
TIMS/ORSA Meeting, Chicago, 1983.

89. Doukidis, G. I.
"An anthology on the homology of simulation with artificial intelligence"
J. Opl. Res. Soc., Vol. 38, No. 8, 1987, pp.701-712.

89a. Doulgeri, Z., et al.
"The scheduling of flexible manufacturing systems"
Annals of the CIRP, Vol. 38, No. 1, 1987, pp.343-346.

90. Doumeingts, G.
"Methodology to design computer integrated manufacturing and control of manufacturing units"
Methods and Tools for Computer Integrated Manufacturing (U. Rembold and R. Dillmann, Eds.), Karlsruhe, Germany, 1983.

91. Doumeingts, G. et al.
"Design of advanced manufacturing systems using AI techniques"
Proc. IXth ICPR, Cincinnati, 1987.

92. Dubois, D.
"A mathematical model of a flexible manufacturing system with limited in-process inventory"
European J. of Opl. Res., Vol.14, 1983, pp.66-78.

93. Dubois, D. and K. E. Stecke
"Using Petri nets to represent production process"
Proc. 22nd IEEE Conf. on Decision and Control, 1983, pp.1062- 1067.

94. Duda, R. O. and R. Reboh
"AI and decision making: The PROSPECTOR experience"
Artificial Intelligence Applications for Business, Ablex, NJ., 1984.

95. Dunn, P.
"British Aerospace's FMS assembles and tests PCBs"
The FMS Magazine, April 1986, pp.97-98.

96. Dupont-Gatelmand, C.
"Flexible manufacturing systems for gearboxes"
Proc. of 1st FMS, pp.453-461.

97. Edghill, J. S. and C. Cresswell
"FMS control strategy - A survey of the determining characteristics"
Proc. of 4th Int. Conf. on FMS, 1986, pp.305-315.

98. Edghill, J. S. and A. Davies
"Flexible manufacturing systems - the myth and the reality"
Int. J. of Advanced Manufacturing Technology, Vol.1, No.1, 1985.

99. Egbelu, P. and J. Tanchoco
"Characterisation of automatic guided vehicle dispatching rules"
Int. J. Prod. Res., Vol.22, No.3, 1984.

100. ElMaraghy, H. A.
"Simulation and graphical animation of advanced manufacturing systems"
J. of Manufacturing Systems, No.1, 1982, pp.53-63.

101. Elzas, M. S.
"Relations between artificial intelligence environments and modelling and simulation support systems"
Modelling and Simulation Methodology in the Artificial Intelligence Era (Elzas, M. S., et al., Eds.), 1986, pp.61-77.

102. Elzas, M. S.
"The kinship between artificial intelligence, modelling and simulation: An

appraisal"
Modelling and Simulation Methodology in the Artificial Intelligence Era (Elzas, M. S., et al., Eds.), 1986, pp.3-13.

103. Eversheim, W. , K. Erhes and H. Schmidt
"Review of German FMS reveals shortcommings"
The FMS Magazine, July 1986, p.159.

104. Eversheim, W. and E. Westkamper
"Computer - controlled manufacturing systems - Planning and organisation"
Proc. of 3rd North American Metalworking Research Conf., Dearborn, Michigan, 1975.

105. Faught, W.S.
"Application of AI in Engineering"
IEEE Computer, Vol.19, No.7, pp.17-27, July 1986.

106. Feigenbaum, E. A.
How to Build Expert Systems
1983.

107. Feltner, C. E. and S. A. Weiner
"Models, myths and mysteries in manufacturing"
Industrial Engineering, July 1985, pp.66-76.

108. Fikes, R. E., P. E. Hart and N. J. Nilsson
"Learning and executing generalised robot plans"
Artificial Intelligence, Vol.3, pp.251-288.

109. Fikes, R. E. and N. J. Nilsson
"STRIPS: A new approach to the application of theorem proving to problem solving"
Artificial Intelligence, Vol.2, pp.189-208.

110. Fishwick, P.
Hierarchical Reasoning: Simulating Complex Processes over Multiple Levels of Abstraction, Ph.D Thesis, Dept. of Computer and Information Sciences,

University of Pennsylvania, 1985.

111. Flitman, A. M. and R. D. Hurrion
"Linking discrete-event simulation models with expert systems"
J. Opl. Res. Soc., Vol.38, No.8, 1987, pp.723-733.

112. Ford, D. R. and B J. Schroer
"An expert manufacturing simulation system"
Simulation, Vol.48, No.5 1987, pp.193-200.

113. Fox, M. S.
"The intelligent management system: An overview"
Processes and Tools for Decision Support (H.G. Sol, Ed.), North-Holland Publishing Company, 1983, pp.105-130.

114. Futo, I. and J. Szeredi
"A discrete simulation system based on artificial intelligence techniques"
Discrete Simulation and Related Fields (I. Javor, Ed.) North-Holland, 1982, pp.135-150.

115. Gaines, B. R. and M. L. Shaw
"Knowledge engineering for an FMS advisory system"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, Chicago, 1986, IFS Pubs.

116. Gershwin, et al.
"Progress in mathematic methods for manufacturing systems"
8th NSF Grantees Conf., Stanford, NTIS Microfiche, Jan. 1981.

117. Goldberg, A. and D. Robson
Smalltalk-80: The language and its implementation, Addison- Wesley, Reading, Mass., 1983.

118. Gorden, W. J. and G. F. Newell
"Closed queueing networks with exponential servers"
Ops. Res., Vol.15, pp.254-265, (1967).

119. Graham, G. S.

"Queueing network models of computer system performance"
Computing Surveys, Vol.10, No.3, pp.219-224.

120. Greenwood, N. R.
Implementing Flexible Manufacturing Systems, MacMillan Education, 1988.

121. Groen, A. et al.
"The integration of simulation and knowledge based systems"
AI Applied to Simulation (E. Kerckhoffs, et al., Eds.), Simulation Series, Vol.18/1, 1985, pp.189-197.

122. Groover, M.P. and E. W. Zimmers
CAD/CAM, 1980, Prentice-Hall

123. Harmon, P. and D. King
Expert Systems, Artificial Intelligence in Business, Wiley, New York, 1985, p.218.

124. Hartley, J.
"Mammoth FMS"
The FMS Magazine, Vol.1, Oct. 1983, pp.277-281.

125. Hartley, J.
FMS at Work,
IFS Publications, 1984.

126. Hartley, J.
"Classic system achieves outstanding results"
The FMS Magazine, April 1986, pp.91-95.

127. Hartley, J.
"NKK's new FMS - potential in need of a plan"
The FMS Magazine, Vol.4, No.4, 1986, pp.207-209.

128. Hayes-Roth, F. and V. R. Lesser
"Focus of attention in the Hearsay-II system"
IJCAI-77, Cambridge, MA, 1977.

129. Henderson, M. J.
"Economical computer integrated manufacturing"
Proc. 5th FMS, 1986, pp.167-178.

130. Hildebrant, R. R.
"Scheduling flexible manufacturing systems using mean value analysis"
Proc. IEEE COnf. on Decision and Control, 1980, New Mexico.

130a. Hill, S. R. and M. A. Rogers
"Practical experience contasting conventional modelling and data driven visual interactive simulation techniques"
SIM-2, June 1986, Chicago, pp.207-220, IES Pubs.

131. Hills, P. R.
"An introduction to simulation using SIMULA"
Publication 5-Ss, Oslo, 1973.

132. Ho, Y. C. and X. Cao
"Perturbation analysis and optimisation of queueing networks"
J. Optimisation Theory and Applications, 1983, Vol.40, No.4, pp.559-582.

133. Ho, Y. C. and C. Cassendra
"A new approach to the analysis of discrete event dynamic systems"
Automatica, 1983, Vol.19, No.2, pp.149-167.

134. Ho, Y. C., M. A. Eyler and T. T. Chien
"A gradient technique for general buffer storage design in a serial production line"
Int. J. Prod. Res., Vol.17, No.6, 1979, pp.557-580.

135. Hollingum, J.
"Helping buyers maximise the benifits of FMS"
The FMS Magazine, Oct. 1986, pp.181-184.]

136. Hollocks, B. W.
"Simulation and the micro"

J. of Opl. Res. Soc., Vol.34, No.4, 1983, pp.331-343.

137. Howard, R. A.
Dynamic Probabilistic Systems, Vol.II: Semi-Markov and Decision Process, John Wiley, NY, 1971.

138. Hutchinson, G. K.
"The control of flexible manufacturing systems: Required information and algorithm structures"
IFAC-77, Japan, 1977.

139. Hutchinson, G. K. and J. Holland
"The economic value of flexible automation"
J. Manufacturing Systems, Vol.1 No.2, 1982.

140. Hutchinson, G. K. and J. J. Hughes
"A generalised model of flexible manufacturing systems"
Proc. of Multi-station, Digitally Controlled Manufacturing System Workshop, Wisconsin University, Jan. 1977, pp.89-114.

141. Istel Ltd.
"WITNESS - Product overview"
Publicity Document, Istel Ltd., Redditch, England, 1986.

142. Iwata, K. F., A. Murotsu and K. Yasuda
"Production scheduling of flexible manufacturing systems"
Annals of CIRP, Vol.31/1, 1982.

143. Iwata, K., F. Oba and K. Yasuda
"Simulation for design and operation of manufacturing systems"
Annals of CIRP, Vol.33/1, 1984.

144. Jackson, J. R.
"Networks of waiting lines"
Ops. Res., Vol.5, pp.518-521.

145. Jackson, J. R.

"Jobshop-like queueing systems"
Management Science, Vol.10, pp.131-142.

145a. Jiao, H.
Loughborough University of Technology, Private Communication, 1989.

146. Kay, J. M.
"Tooling for FMS - can simulation help?"
SIM-1, 1985 (IFS Pubs., Stratford.)

147. Kellock, B.
"Tool head change suit volume needs"
Machinery and Production Engineering, Sept. 1986, pp.122-125.

148. Kerckhoffs, E. J. H., et al.
"General considerations on AI applied to simulation"
AI Applied to Simulation (E. Kerckhoffs, et al., Eds.), Simulation Series, Vol.18/1, 1985, pp.ix-xii.

149. Kesselburg, C.
"Jig boring machines in production with flexible manufacturing systems"
Proc. 5th Int. Conf. on FMS, Nov. 1986, pp.327-336.

150. Khoshnevis, B. and A. Chen
"An expert simulation model builder"
Intelligent Simulation Environment (Luker, P. and H. Adelsberger, Eds.), Simulation Series, Vol.17, No.1, Jan. 1986, pp.129-132.

151. Kimemia, J. G. and S. B. Gershwin
"Multicommodity network flow optimisation in flexible manufacturing systems"
Report ESL-FR-834.2, Electronic Systems Lab., MIT, 1980.

152. Kimemia, J. G. and S. B. Gershwin
"An algorithm for the computer control of a flexible manufacturing system"
IEE Transactions, Vol.15, No.4, 1983.

153. Kimemia, J. and S. B. Gershwin

"Flow optimisation in flexible manufacturing systems"
Int. J. Prod. Res., Vol.23, No.1, 1985, pp.81-96.

154. Klahr, P. and W.S. Faught
"Knowledge-based simulation"
Proc. of 1st Annual National Conf. on Artificial Intelligence, Palo Alto, California, 1980, pp.181-183.

155. Klahr, P., W. S. Faught and G.R. Martins
"Rule-oriented simulation"
IEEE Cybernetics and Society, 1980, pp.350-354.

155a. Kochan, A.
"Gamble in FMS pays off"
The FMS Magazine, Jan. 1985, pp.51-53.

156. Kochan, A.
"Holset takes bold but unorthodox approach"
The FMS Magazine, July 1986, pp.33-135.

157. Kochan, A.
"East German makes impressive try at FMS"
The FMS Magazine, July 1986, pp.131-132.

158. Kochan, A.
"French interest centres on flexible cells at Paris"
The FMS Magazine, July 1986, pp.147-149.

159. Kochan, A.
"KTM uses DOI grant to build machine tools with FMS"
The FMS Magazine, April 1988, pp.158-160.

160. Kumar, K. R. and A. Vannelli
"Design of flexible production systems: Capacity balancing and subcontracting strategies"
Proc. 2nd ORSA/TIMS Conf. on FMS, 1986.

161. Kunz, J. C., T. P. Kehler and M. D. Williams
"Applications development using a hybrid AI development system"
The AI Magazine, Vol.5, No.3, 1984.

162. Kusiak, A. and J. P. Cyrus
"Routing and scheduling of automated guided vehicles"
Toward the Factory of the Future (H. J Bullinger and H. J. Warnecke, Eds.), 1985[redacted]

163. Lacy, K.
"Factory where man is a mere observer"
Machinery and Production Engineering, March 1982, pp.27-31.

164. Lenz, J. E.
"MAST: A simulation tool for designing computerised metalworking factories"
Simulation, Feb. 1983, pp.51-58.

165. Lenz, J. E.
"MAST: A simulation tool as advanced as the FMS it studies"
Proc. 1st Int. Conf. on Simulation in Manufacturing, 1985.

166. Lenz, J. E.
"FMS: What happens when you don't simulate"
Simulation (R. D. Hurrion, Ed.), IFS Publications, 1986.

167. Lenz, J. E.
"General theories of flexible integration"
Proc. 5th Int. Conf. on FMS, Nov. 1986, pp.255-264[redacted]

168. Lenz, J. E.
"Automatic evaluation of simulation output or 'The answer to why?'"
Proc. of SIM-3, 1987, pp.3-10[redacted]

169. Little, D.
"The integration of FMS with existing factory systems"
Proc. 5th Int. Conf. on FMS, 1986, pp.439-448.

170. Little, J. C.
"A proof of the queueing formula L = W"
Ops. Res., Vol.9, 1961, pp.383-387.

171. Looveren, A. J. V., L. F. Gelgers and L. N. V. Wassenhove
"A review of FMS planning models"
Modelling and Design of Flexible Manufacturing Systems (A. Kusiak, Ed.), 1986, Elsevier.

172. Luca, A. D.
"Optimal production planning for FMS: an 'optimum batching' algorithm"
Proc. of 3rd Int. Conf. on FMS, 1984.

172a. Maione, B., et al.
"Accuracy in evaluating FMS Performances through operational analysis"
Annals of the CIRP, Vol.34/1/1985, pp.401-404.

173. Markowitz, H. M., et al.
SIMSCRIPT: A Simulation Programming Language, Prentice Hall, 1963.

174. Martin, D. and Pritsker
"Simulation in the life cycle of flexible manufacturing systems"
Proc. 1st ORSA/TIMS Conf. on FMS, Ann Arbor, 1984.

175. Martinez, J., H. Alla and M. Silva
"Petri nets for the specification of FMSs"
Modelling and Design of Flexible Manufacturing Systems (A. Kusiak, Ed.), Elsevier, Amsterdam, 1986.

176. Mathewson, S. C.
"The application of program generator software and its extension to discrete event simulation modelling"
IIE Transactions, Vol.16, No.1, March 1984, pp.3-18.

177. Mellichamp, J. M. and A. F. A. Wahab
"An expert system for FMS design"
Simulation, Vol.48, No.5, 1987, pp201-208.

178. McArthur, D. J., P. Klahr, L. Narain
"ROSS: An object-oriented language for constructing simulation"
Expert Systems (P. Klahr and D. Waterman, Eds.), Addison- Wesley, 1986.

179. McBean, D.J., et al.
"Concepts of an FMS"
Proc. of 1st Int. Conf. on FMS, 1982, pp.497-500. CIFS Pubs.

180. McRoberts, M., M. Fox and N. Husain
"Generating model abstraction scenarios in KBS"
AI, Graphics and Simulation (G. Birtwistle, Ed.), SCS, 1985, pp.29-33.

181. McQueen, L.
Loughborough University, Private Communication, 1989.

182. Menon, U.
A Multi-Objective Production Planning Framework for Automated Manufacturing Systems, Ph.D Thesis, Nottingham University, 1985.

183. Menon, U. and P. J. O'Grady
"A flexible multiobjective production planning framework for automated manufacturing systems"
Engineering Costs and Production Economics, 1984.

184. Micheletti, G. M.
"Workload optimisation in a FMS modelled as a closed network of queues"
Annals of CIRP, Vol.28/1/1979.

185. Middleton, S. and R. Zanconato
"BLOBS: An object-oriented language for simulation and reasoning"
AI Applied to Simulation (E. Kerchoffs, et al., Eds.), SCS, San Diego, Calif., 1986, pp.130-135.

186. Milacic, V. R.
"How to build expert systems"
Annals of CIRP, Vol. 35, No. 2, 1986, pp.445-450.

187. Mills, R. I.
"Computer simulation - A feasibility and planning tool for FMS"
Proc. 2nd Int. Conf. on FMS, London, 1983, pp.185-196, IFS Pubs.

188. Mills, R. I.
"Simulation for manufacturing systems - A critical review"
Proc. of 5th Int. Conf. on FMS, 1986, pp.225-234, IFS Pubs.

189. Mills, R. I. and J. J. Talavage
"Simulation programs for FMS design"
Proc. of SIM-1, U.K., 1985, pp.217-227, IFS Pubs.

190. Munakata, K.
"Just what is meant by flexibility in manufacturing problems"
Metal Working Engineering and Marketing, 1982, pp.104-107.

191. Musselman, J.
"Computer simulation: A design tool for FMS"
Manufacturing Engineering, September 1984, pp.117-120.

192. Nakamura, N. and T. Shingu
"Scheduling of flexible manufacturing systems"
Toward the Factory of the Future (H. J. Bullinger and H. J. Warnecke, Eds.), 1985, Stuttgart.

192a. Newman, S.
Forth comming Ph.D Thesis, Loughborough University of Technology, 1989.

193. Newell, A. and H. A. Simon
"GPS: A program that simulates human thought"
Computers and Thought (E. A. Feigenbaum and J. Feldman, Eds.), McGraw-Hill, NY, 1963.

194. Nii, H. P. and N. Aiello
"AGE (attempt to generalise): A knowledge-based program for building knowledge-based programs"

Proc. IJCAI-79, 1979, Tokyo, pp.645-655.

195. Nilsson, N. J.
Principles of Artificial Intelligence
Tioga, Palo Alto, Calif., 1980.

196. Nof, S., M. Barash and J. Solberg
"Operational control of item flow in versatile manufacturing systems"
Int. J. Prod. Res., Vol.17, No.5, 1979.

197. Nordstrom, C.
"Flexible manufacturing systems for turning"
Proc. Ist Int. Conf. on FMS, 1982.

198. Numerical Engineering Society/Institute of Production Engineers
Flexible Manufacturing Systems, Notes to supplement the video cassette, 1985.

199. O'Keefe, R.
"Simulation and expert systems - A taxonomy and some examples"
Simulation, Vol.46, No.1, 1986, pp.10-16.

200. O'Keefe, R. M. and J. W. Roach
"Artificial intelligence approaches to simulation"
J. Opl. Res. Soc., Vol.38, No.8, 1987, pp.713-722.

201. Oren, T. I.
"Artificial intelligence and simulation"
AI Applied to Simulation (E. Kerckhoffs, G. Vansteenkiste and B. Ziegler, Eds.), Simulation Series, Vol.18/1, 1985, pp.3-8.

202. Oren, T. I.
"Implications of machining learning in simulation"
Modelling and Simulation Methodology in the Artificial Intelligence Era (M. S. Elzas, T. I. Oren and B. P. Zeigler, Eds.), 1986, pp.41-57, Elsevier.

203. Oren, T. I. and B. B. Zeigler
"Artificial intelligence in modelling and simulation: Directions to explore"

Simulation, April 1987, pp.131-134.

204. Parris,
The Design of an FMS for Manufacturing Grass Mowers
Ph.D Thesis, Loughborough University of Technology, 1983.

205. Pegden, C. D.
"Introduction to SIMAN"
System Modelling Corporation, State College, Pennsylvania, 1982.

206. Pelusi, J.
"GMF uses the MAP in low volume production"
The FMS Magazine, Jan. 1986, pp.13-15.

207. Peterson, J. L.
Petri Net Theory and the Modelling of Systems
Prentice-Hall, 1981

208. Phelps, R. I.
"Artificial intelligence - An overview of similarities with O.R."
J. of Opl. Res. Soc., Vol.37, No.1, 1986, pp.13-20.

209. Pidd, M.
Computer Simulation in Management Science, Wiley, London, 1984.

210. Powley, C.
"A glimpse of what is on the horizon"
Machinery and Production Engineering, Vol.4, Feb. 1987, pp.40- 44.

211. Pritsker, A. A. B.
The GASP-IV Simulation Language, Wiley, 1974.

212. Pritsker, A. and C. Pegden
Introduction to Simulation and SLAM, John Wiley and Sons, 1979.

213. Purdom, P. B.
"The Citroen flexible manufacturing cell"

Proc. of 2nd FMS, 1983, pp.93-103, IFS Pubs.

214. Purdom, P. B. and T. Palazzo
"The Citroen(CCM) flexible manufacturing cell"
Proc. of 1st FMS, 1982, pp.151-169, IFS Pubs.

215. Rajagopalan, R.
"Qualitative modelling and simulation: A survey"
AI Applied to Simulation (E. Kerckhoffs, G. Vansteenkiste and B. Ziegler, Eds.), Simulation Series, Vol.18/1, 1985.

216. Randhawa, S. U. and D. Bedworth
"Factors identified for use in comparing conventional and flexible manufacturing systems"
Industrial Engineering, June 1985, pp40-44.

217. Ranky, P.
The Design and Operation of Flexible Manufacturing Systems, IFS Publications, 1983.

218. Rathmill, K.
"Trends in the application of computer simulation"
The FMS Magazine, Vol.5, No.3, 1987, pp.150-152.

219. Rathmill, K. and W. W. Chan
"What simulation can do for FMS design and planning"
The FMS Magazine, April 1984

220. Rathmill, K., N. Greenwood and M. Houshmaud
"Computer simulation of FMS"
Proc. 1st Int. Conf. on FMS, Brighton, 1982, IFS Pubs.

221. Reddy, R.
"Epistemology of knowledge based simulation"
Simulation, Vol.48, No.4, 1987, pp.162-166.

222. Reddy, Y., et al.

"INET: A knowledge based simulation model of a corporate distribution system"
IEEE Conf. on Trends and Applications: Automating Intelligent Behaviour Applications and Frontiers, 1983, pp.109-118.

223. Reddy, Y.V.R., et al.
"Knowledge-based simulation system"
IEEE Software, Vol. 3, pp.26-37, 1986.

224. Reddy, Y., M. S. Fox and N. Husain
"Automating the analysis of simulations in KBS"
AI, Graphics and Simulation (G. Birtwistle, Ed.), SCS, 1985, pp.34-40.

225. Reiser, M. and S. S. Lavenberg
"Mean-value analysis of closed multichain queueing networks"
J. ACM, Vol.27, No.2, 1980, pp.313-323.

226. Rich, E.
Artificial Intelligence
McGraw-Hill, New York, 1983.

227. Robert, E. and N. Shires
"Parallel processing applied to the simulation of highly automated batch manufacturing systems"
SIM-1, March 1985, IES Pubs.

228. Ruiz-Mier, S., et al.
"Towards a knowledge-based network simulation environment"
Proc. of IEEE Winter Simulation Conf., Piscataway, N.J., 1985, pp.232-236.

229. Ruiz-Mier, S. and J. Talavage
" A hybrid paradigm for modelling of complex systems"
Simulation, Vol.48, No.4, 1987, pp.135-141.

230. Sacerdoti, E. D.
"Planning in a hierarchy of abstraction spaces"
Artificial Intelligence, Vol.5, 1974, pp.115-135.

231. Sacerdoti, E. D.
"The non-linear nature of plans"
IJCAI-75, Tbilisi, USSR, 1975.

232. Sackett, P. J.
"Control policy for realising high performance in a mixed flexible and conventional manufacturing system"
Proc. of 1st Int. Conf. on FMS, 1982, pp.389-395, IFS Pubs, Ed.

233. Sarin, S. C. and C. S. Chen
"A mathematical model for manufacturing system selection"
Flexible Manufacturing Systems: Methods and Studies (A. Kusiak, Ed.), Elsevier Science Publishers B.V., 1986.

234. Schank, R. C. and R. P. Abelson
Scripts, Plans, Goals and Understanding, Hillsdale, Erlbaum, NJ.

234a. Schutzer, D.
Artificial Intelligence
Van Nostrand Reinhold Company, New York, 1987.

235. Schweitzer, P. J.
"Maximum throughput in finite-capacity open queueing networks with product-form solutions"
Management Science, Vol.24, pp.217-223.

236. Seidmann, A. and S. Y. Nof
"Unitary manufacturing cell design with random product feedback flow"
IIE Trans., Vol.17, No.2, 1985.

237. Seidmann, A. and P. J. Schweitzer
"Part selection policy for a flexible manufacturing cell feeding several production lines"
IIE Trans., Vol.16, No.4, 1984, pp.355-362.

238. Seliger, G., B. Viehweger and B. Wieneke

"Decision support for planning flexible manufacturing systems"
SIM-2, Chicago, June 1986, IFS Pubs.

239. Shalev-Oren, S., A. Seidmann and P. J. Schweitzer
"Analysis of flexible manufacturing systems with priority scheduling: PMVA"
Annals of Operations Research, 3, 1985, pp.115-139.

240. Shanehchi, J.
"EXPRESS: A man-machine interface for simulation"
Proc. of SIM-1, 1985, U.K., IFS Pubs.

241. Shanker, K. and Y. J. Tzen
"A loading and dispatching problem in a random flexible manufacturing system"
Int. J. Prod. Res., Vol.23, No.3, 1985, pp.579-595.

242. Shannon, R. E.
"Knowledge based simulation techniques for manufacturing"
Int. J. Prod. Res., Vol.26, No.5, 1988, pp.953-973.

243. Shaw, M. and A. Winston
"Automated planning and flexible scheduling: A knowledge-based approach"
Proc. of IEEE Conf. on Automation and Robotics, St. Louis, 1985.

244. Shires, N.
The Distributed Simulation of Highly Automated Batch Manufacturing Systems,
Ph.D Thesis, Loughborough University, 1988.

245. Shivnan, J. and J. Browne
"AI based simulation of advanced manufacturing systems"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, 1986, IFS Pubs.

246. Silva, C. M. and J. M. Bastos
"The use of decision mechanisms in visual simulation for flexible manufacturing systems modelling"

AI Applied to Simulation (E. Kerckhoffs, G. Vansteenkiste and B. Ziegler, Eds.), Simulation Series, Vol.18/1, 1985.

247. Smith, A. W., D. R. Hughes and R. S. Haull
"The use of AI techniques in dynamic simulation"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, Chicago, 1986.

248. Solberg, J. J.
"Optimal design and control of computerised manufacturing systems"
Proc.AIIE Systems Engineering Conf., 1976.

249. Solberg, J. J.
"Quantitative design tools for computerised manufacturing systems"
Proc. 6th North American Metalworking Research Committee, Florida, 1978.

250. Solberg, J. J.
"Computer models for design and control of flexible manufacturing systems"
16th NCS, 1979, pp.111-117.

251. Solberg, J. J.
"Mathematical design tools for integrated production systems"
Proc. 23rd MTDR Conf., 1982.

252. Solberg, J. J. and D. C. Anderson
"The factory of the future: A framework for research"
Proc. of 11th Conf. on Prod. Res. & Tech., Pittsburgh, 1984.

253. Spur, G., B. Viehweger and B. Wieneke
"Problem-oriented methods for the planning and optimisation of flexible manufacturing systems"
Proc. of 2nd Int. Conf. on FMS, 1983, IFS Pubs.

254. Stecke, K. E.
"Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems"
Management Science, Vol.29, No.3, 1983, pp.273-288.

255. Stecke, K. E.
"Design, planning, scheduling, and control problems of flexible manufacturing systems"
Annals of Operations Research, 3, 1985, pp.3-12.

256. Stecke, K. E.
"Heuristics for the loading of flexible manufacturing systems"
Flexible Manufacturing (A. Raouf and S. I. Ahmad, Eds.), 1985, Elsevier.

257. Stecke, K. E. and J. J. Solberg
"Loading and control policies for a flexible manufacturing system"
Int. J. Prod. Res., Vol.19, No.5, 1981, pp.481-490.

258. Stecke, K. E. and J. J. Solberg
"The optimality of unbalanced workloads and machine group sizes for flexible manufacturing systems"
Working Paper 290, Graduate School of Business Administration, University of Michigan, 1982.

259. Stefik, M. J.
"Planning with constraints"
Artificial Intelligence, Vol.16, 1981, pp.111-140.

260. Stefik, M. J.
"Planning and meta-planning"
Artificial Intelligence, Vol.16, 1981, pp.141-169.

261. Stefik, M., D. G. Bobrow, S. Mittal and L. Conway
"Knowledge programming in LOOPS: Report on an experimental course"
The AI Magazine, Fall 1983, pp.3-13.

262. Steiner, S.
"Production planning and control - A review"
Production Planning and Control for Flexible Manufacturing, Coventry Lanchester Polytechnic, 1986.

263. Strandhagan, J. O.

"Expert knowledge in object oriented simulation of manufacturing systems"
Proc. of SIM-3, Nov. 1987, pp23-30.

264. Suardo, G.
"Optimisation of a closed network of queues"
Report ESL-FR-834.3, Electronic Systems Lab., MIT, 1978.

265. Suri, R.
"New techniques for modelling and control of flexible automated manufacturing systems"
Proc. IFAC on Control Science and Technology, Kyoto, 1981.

265a. Suri, R.
"An overview of evaluative models for flexible manufacturing systems"
Annals of Operational Research, Vol.3, 1985, pp.13-21.

266. Suri, R. and X. Cao
"Optimisation of flexible manufacturing systems using new techniques in discrete event systems"
Proc. 29th Alterton Conf. on Communication, Control and Comput., Monticello, 1982.

267. Suri, R. and X. Cao
"The phantom customer and marked customer methods for optimisation of multiclass queueing networks with blocking and general service times"
Technical Report, Division of Applied Sciences, Harvard University, 1983.

268. Suri, R. and R. R. Hildebrant
"Modelling flexible manufacturing systems using mean-value analysis"
J. Manufacturing Systems, Vol.3, No.1, pp.27-38.

269. Suri, R. and C. K. Whitney
"Decision support requirements in flexible manufacturing"
J. of Manufacturing Systems, Vol.3, No.1, 1984, pp.61-69.

270. Takayama, S.
"DNC system in a low volume production of a wide variety of parts"

Metal Working Engineering and Marketing, March 1981, pp.28-32.

271. Talavage, J. J.
"Simulation analysis of an operational computerised manufacturing system"
Proc. 2nd Int. Conf. on Computer Aided Manufacture, CAM-78, Glasgow, 1978.

272. Tate, A.
"Introducing goals and their use"
IJCAI-75, Tbilisi, USSR, 1975, pp.215-218.

273. Tate, A.
"Generating project networks"
IJCAI-77, Boston, MA, 1977.

274. Tate, A.
"A review of knowledge-based planning techniques"
Expert Systems 85 (M. Merry, Ed.), University of Warwick, 1985.

275. Thorneycroft, M.
"Recent advances in the technology of flexible turning cells"
Proc. 1st Int. Conf. on FMS, 1982, pp.307-316, IFS Pubs.

276. Tocher, K.
The Art of Simulation, Hodder & Stoughton Educational Publishers, London, 1963.

277. Touretzky, D. S.
LISP: A Gentle Introduction to Symbolic Computation, Harper and Row Publishers, N.Y. 1984.

278. Van Melle, W., E. H. Shortliffe and B. G. Buchanan
"EMYCIN: A knowledge engineer's tool for constructing rule- based expert systems"
Rule-Based Expert Systems (B. Buchanan and E. Shortliffe, Eds.), Addison-Wesley, New York, 1984, pp.302-328.

279. Vaucher, J.
"Views of modelling: Comparing the simulation and AI approaches"
AI, Graphics and Simulation (G. Birtwistle, Ed.), SCS, 1985, pp.3-7.

280. Vere, S. A.
"Planning in time: Windows and durations for activities and goals"
IEEE PAMI, Vol.5, 1983, pp.246-266.

281. Vilain, M.
"A system for reasoning about time"
Proc. of AAAI-82, Pittsburgh, Aug. 1982.

282. Vinod, B. and J. J. Solberg
"Performance models for unreliable flexible manufacturing systems"
Int. J. Management Science, Omega, Vol.12, 1984, pp.299-380.

283. Vinod, B. and J. J. Solberg
"The optimal design of flexible manufacturing systems"
Int. J. Prod. Res., Vol.23, No.6, 1985, pp.1141-1151.

284. Warnecke, H. J. and E. Gericke
"Modelling and simulation of automated manufacturing processes"
Proc. IFAC-77, Tokyo, 1977, pp.1-5.

285. Warnecke, H. J. and P. Scharf
"Some criteria for the develpment of integrated manufacturing systems"
Proc. 2nd Int. Conf. on Prod. Res., 1973, Copenhagen.

286. Warnecke, H. J. and G. Vettin
"Strategies for the organisation and control of discontinuous conveyors in flexible manufacturing systems"
Proc. of 9th CIRP Seminar on Manufacturing Systems, Cranfield, July 1977.

287. Warnecke, H. J. and G. Vettin
"Technical investment planning of flexible manufacturing systems - The application of practice-oriented method"
J. of Manufacturing Systems, No.1, 1982, pp.89-98.

288. Warnecke, H. J., T. Zipse and K. Zeh
"Simulation and computer aided planning of FMS"
Proc. of 3rd Int. Conf. on FMS, 1984, IFS Pubs.

288a. Waterlow, G.
"Development and application of simulation techniques"
ACME Directorate Research Portfolio, Nov. 1988.

289. Waterman, D. A.
A Guide to Expert Systems
Addison-Wesley, 1986.

290. Webb, S.
"User project management for Holset FMS"
Proc. 5th Int. Conf. on FMS, 1986, pp.135-144.

291. Weck, M. and A. Schuring
"What aids are available for the design and development of computer controlled manufacturing systems"
Proc. NAMRC III, S.M.E., 1975, pp.751-770.

292. Whitney, C. K.
"Control principles in flexible manufacturing"
J. of Manufacturing Systems, Vol.4, 1985.

293. Wichmann, K. E.
" An intelligent simulation environment for the design and operation of FMS"
Proc. 2nd Int. Conf. on Simulation in Manufacturing, Chicago, 1986, IFS Pubs.

294. Wichmann, K. E.
"Trends in the development of simulation software tools for analyzing manufacturing systems"
Proc. of SIM-3, 1987, pp.39-50, IFS Pubs.

295. Wilhelm, W. E. and S. C. Sarin
"Models for the design of flexible manufacturing systems"

Proc. Annual Industrial Engineering Conf., 1983.

296. Wilhelm, W. E. and H. M. Shin
"Effectiveness of alternate operations in a flexible manefaturing system"
Int. J. Prod. Res., Vol.23, No.1, 1985.

297. Williamson, D. T.
"System 24 - A new concept of manufacture"
Proc. 8th Int. Machine Tool Design and Research, 1967, pp.327- 376.

298. Wills, K. F., et al.
"Advanced computer aided engineering and manufacturing"
Proc. of Institute of Mechanical Engineers, Vol.197B, May 1983, pp.81-89.

299. Winston, P. H.
Artificial Intelligence, 2nd Edition, Reading, Addison-Wesley, MA, 1984.

300. Winston, P. H. and B. Horn
LISP, Addison-Wesley, Reading, MA.,1980.

301. Wysk, R. A. et al.
"A multi-pass expert control system (MPECS) for flexible manufacturing systems"
Integrated and Intelligent Manufacturing (C.R. Liu and T. C. Chang, Eds.), NY, 1986, pp.9-25.

302. Xerox Corporation
The Interlisp-D Reference Manual, 1983.

303. Zeigler, B. P. and L. D. Wael
"Towards a knowledge-based implementation of multifacetted modelling methodology"
AI Applied to Simulation (E. Kerckhoffs, et al., Eds.), Simulation Series, Vol.18/1, 1985, pp.42-51.

304. Zhang, P.
The management of tool flows in flexible manufacturing systems for cylindrical

parts, Ph.D Thesis, Loughborough University, 1988.

305. Zhang, P. and R. Bell
"Tool management for highly automated turning systems"
Proc. of 3rd Int. Conf. in Metal Cutting, Non-Conventional Machining and Their Automation, Vol.1, May 1987, Nanjing, China.

306. Zhang, P., R. de Souza and R. Bell
"Tool management for highly automated turning systems"
Proc. of 4th Int. Conf. in Metal Cutting, Non-Conventional Machining and Their Automation, 1989, Beijing, China.