Integrated knowledge-based hierarchical modelling of manufacturing organizations

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:


Metadata Record: https://dspace.lboro.ac.uk/2134/32104

Publisher: © Hong Jiao

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
INTEGRATED KNOWLEDGE BASED
HIERARCHICAL MODELLING
OF MANUFACTURING ORGANIZATIONS

by

HONG JIAO

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
June 1991

© by Hong Jiao, 1991
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 Husband, Ping and Our Parents

For Their Love, Support and Encouragement
ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my supervisor, Professor R. Bell, for his supervision, encouragement and interest in this research; co-supervisor, Dr. K. Popplewell for his support and help throughout this work. My gratitude to Dr. W. Wang for his helpful discussion and support of the Xerox Workstation.

I am indebted to GEC ALSTHOM Large Machines Ltd. for their interest in the project and providing the data for the case study, especially Mr. J. Bult, Mr. C. Tayton and Miss. C. Alderton.

I thank Mr. P. Morrissey for his tremendous help for the preparation of this thesis.

My thanks are also made to all my friends for their friendship, inspiration and help throughout the period of my work; in particular, Dr. S. T. Newman, Dr. P. Zhang, Dr. R. De Souza, Miss. L. McQueen, Dr. G. Varvakis, Mr. M. Ozbayrak, Dr. R.I.M. Young, Mr. A. S. Goh, Miss. B. Yu, and Mr. A. Tavakoli.

I would also like to thank all my colleagues at Harbin University of Technology for their support by undertaking the extra work introduced by my absence from the university, particularly Professor, S. J. Wang, for his previous guidance and help.

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.
SYNOPSIS

The objective of this thesis is to research into an integrated knowledge based simulation method, which combines the capability of knowledge based simulation and a structured analysis method, for the design and analysis of complex and hierarchical manufacturing organizations. This means manufacturing organizations analysed according to this methodology can manage the tactical and operational planning as well as the direct operation of shop floor. The thesis consists of three main parts.

In the first part, the state-of-the-art of manufacturing organizations are reviewed with the emphasis laid on the structure and management of manufacturing organizations and operation of computer integrated manufacturing systems. Various modelling techniques are examined to highlight the requirement for the integrated knowledge based hierarchical simulation method. Based on the discussion of limitations of current methods and the challenge of manufacturing organizations and modelling techniques, the thesis moves to the identification of the new design method and establishes the scope of the research.

The second part reports the research on an integrated methodology. The work begins with the general analysis of operational structure and design requirements for manufacturing organizations. Based on this understanding, the formal description of manufacturing organizations is established by a system analysis method IDEF0 to provide the interface directly linking to a system simulation. Then the work is concentrated on the creation of simulation models based on this IDEF0 presentation in the STEM knowledge based simulation environment.

In this simulation model, attention has been given to functional and operational structure building, interface design and output analysis. The functional structure mainly presents the physical arrangement of simulation entities in manufacturing organizations, which parallels the IDEF0 representation. The operational structure indicates the way to manipulate the modelling process, which contains data base, knowledge base, inference engine and working memory. The interface provides an easy user involvement with the system. The interactions include data input, rule entry, and presentation of results with other facilities provided by STEM. Output results present the measurements of the system performance by carefully defined performance figures which can best reflect the behaviour of the system.

The third part is concerned with the application and validation of the model with an industrial case study to examine the design target defined in this methodology, to test the system performance, and to evaluate the alternative strategies. The results also can contribute to the re-organisation plan of the company. Conclusions drawn from this research and recommendation for the future work are finally stated.
Table of Contents

DECLARATION

ACKNOWLEDGEMENTS

SYNOPSIS

Chapter 1 INTRODUCTION

Chapter 2 LITERATURE SURVEY

2.1 Introduction

2.2 Industrial Organizations

2.2.1 Introduction

2.2.2 Changes Faced by Industrial Organizations

2.2.3 Objectives of Industrial Organizations

2.2.4 Categorisation of Industrial Organizations

2.2.5 Structural Representation of Industrial Organizations

2.2.6 ISO Model

2.2.7 Major Functions in Industrial Organizations

2.2.8 Production Planning and Control

2.3 Computer Integrated Manufacturing (CIM)

2.3.1 CIM Definition

2.3.2 Methods of Achieving Integrated Manufacturing

2.4 Modelling Techniques

2.4.1 Introduction

2.4.2 Functions of Models

2.4.3 General Classification of Modelling Methods

2.4.4 General Single-Level Modelling Methods

2.4.4.1 Introduction

2.4.4.2 Static Allocation Method

2.4.4.3 Mathematical Programming

2.4.4.4 Queueing Network

2.4.4.5 Petri Nets

2.4.4.6 Heuristic Algorithms

2.4.5 Multi-Level Modelling Methods

2.4.5.1 Introduction

2.4.5.2 GRAI Method

2.4.5.3 IDEF Techniques
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.6 Simulation Based Modelling Techniques</td>
<td>21</td>
</tr>
<tr>
<td>2.4.6.1 Introduction</td>
<td>21</td>
</tr>
<tr>
<td>2.4.6.2 Conventional Simulation</td>
<td>22</td>
</tr>
<tr>
<td>2.4.6.2.1 General Purpose Computer Languages</td>
<td>22</td>
</tr>
<tr>
<td>2.4.6.2.2 Simulation Softwares</td>
<td>22</td>
</tr>
<tr>
<td>2.4.6.3 AI Based Simulation</td>
<td>23</td>
</tr>
<tr>
<td>2.4.6.3.1 Artificial Intelligence in Simulation</td>
<td>23</td>
</tr>
<tr>
<td>2.4.6.3.2 Knowledge Based Simulation</td>
<td>25</td>
</tr>
<tr>
<td>2.4.6.4 Generalised Manufacturing Systems Simulators</td>
<td>27</td>
</tr>
<tr>
<td>2.5 System Analysis Methods</td>
<td>31</td>
</tr>
<tr>
<td>2.5.1 Functions of System Analysis Methods</td>
<td>31</td>
</tr>
<tr>
<td>2.5.2 Structured Analysis Methods</td>
<td>32</td>
</tr>
</tbody>
</table>

| Chapter 3 STRUCTURE AND PERFORMANCE OF INDUSTRIAL ORGANIZATIONS         | 52   |
| 3.1 Introduction                                                        | 52   |
| 3.2 General Categorisation of Industrial Organizations                  | 52   |
| 3.2.1 Introduction                                                      | 52   |
| 3.2.2 Process Based Classification                                      | 52   |
| 3.2.3 Volume Based Classification                                       | 53   |
| 3.2.4 Scheduling Based Classification                                   | 54   |
| 3.3 Structural Representation of Industrial Organizations               | 55   |
| 3.4 Basic Functions in Industrial Organizations                         | 55   |
| 3.4.1 Sales and Marketing                                               | 56   |
| 3.4.2 Engineering                                                       | 56   |
| 3.4.3 Production                                                        | 57   |
| 3.4.4 Personnel                                                         | 57   |
| 3.4.5 Finance                                                           | 58   |
| 3.4.6 Relationship between Basic Functions                              | 58   |
| 3.5 Performance Judgement of Industrial Organizations                   | 58   |

| Chapter 4 THE PRODUCTION PLANNING AND CONTROL                            | 62   |
| 4.1 Introduction                                                         | 62   |
| 4.2 Forecasting                                                          | 62   |
| 4.3 Production Planning and Control                                      | 62   |
| 4.3.1 Manufacturing Resources Planning                                   | 63   |
| 4.3.2 Just-In-Time                                                       | 64   |
| 4.3.3 Optimized Production Technology                                    | 64   |
| 4.3.4 Comparison between MRPII, JIT and OPT                              | 65   |
| 4.4 Material Control                                                     | 66   |
| 4.5 Conclusion                                                           | 66   |
8.7.5 Expanded Design Stage 112
8.8 Model Validation 113
8.9 Output Analysis 114
8.10 Discussion 115

Chapter 9 OVERALL STRUCTURE OF THE SIMULATION SYSTEM 118

9.1 Introduction 118
9.2 Research Objective 118
9.3 Hierarchical Building of the Simulation Model 119
9.4 Functional Structure of the Simulation Model 121
  9.4.1 Specialised User Interface 121
  9.4.2 Specialised Data Base 122
  9.4.3 Specialised Knowledge Base 125
9.5 Operational Structure of the Simulation Model 128
9.6 Discussion 129

Chapter 10 GENERAL MODELLING KNOWLEDGE 146

10.1 Introduction 146
10.2 Research Objective 146
10.3 Initial Comments 146
10.4 Top Level of the Model 147
  10.4.1 Initial Comments 147
  10.4.2 State Description 147
10.5 Company Level 150
  10.5.1 Initial Comments 150
  10.5.2 State Description 151
10.6 Shop Level 156
  10.6.1 Initial Comments 156
  10.6.2 State Description 156
10.7 Flexible Machining Cell Level 160
  10.7.1 Initial Comments 161
  10.7.2 State Description 161
10.8 Machine Level 163
  10.8.1 Initial Comments 163
  10.8.2 State Description 163
10.9 Discussion 166

Chapter 11 MODEL SPECIALISED INTERFACE 172

11.1 Introduction 172
11.2 Research Objective 172
11.3 Interface Requirements
11.4 Structure of the Interface
11.5 The Global Menu
11.6 Data Entry Menu
11.7 Rule Entry Menu
11.8 Data Output Menu
11.9 Discussion

Chapter 12 SIMULATION OUTPUTS

Chapter 13 THE CASE STUDY

Chapter 14 RESULTS OF THE CASE STUDY
Chapter 15 CONCLUDING DISCUSSION

15.1 Introduction 219
15.2 The Efficient Integrated Methodology 219
15.3 Static Hierarchical Modelling of Manufacturing Organizations 219
15.4 Hierarchical Modelling Approach 220
15.5 Simulation System Structure 220
15.6 Simulation System Operation 220
15.7 Modelling Knowledge of the Simulation Model 221
15.8 Model with Different Levels 221
15.9 Data and Rule Driven Environment 221
15.10 User Interface 222
15.11 Model Output 222
15.12 Computational Environment 222
15.13 Case Study Experience
  15.13.1 Aggregated Manufacturing Organizations Model
  15.13.2 Adaptability
  15.13.3 Approximate Modelling and Detailed Modelling
  15.13.4 Easy Data Base Management
  15.13.5 Influence of Assumptions
  15.13.6 Influence of Different Rules
  15.13.7 Influence of Initial Running Condition
  15.13.8 Accuracy of Results
  15.13.9 Potential Improvements

Chapter 16 CRITIQUE OF THE RESEARCH
  16.1 Introduction
  16.2 Research Objective
  16.3 Generic Nature of the Model versus Specialised Modelling
  16.4 Justification of Hierarchical Modelling and Discussion of Alternatives
  16.5 Rationale for Linking IDEF0 and STEM
  16.6 Requirements to Improve the User Interface
  16.7 Practical Use of the Modelling Tool
  16.8 Concluding Remarks

Chapter 17 CONCLUSIONS

Chapter 18 FURTHER WORK
  18.1 Introduction
  18.2 Integrated Environment
  18.3 Modelling of Interactions
  18.4 Modelling of the Engineering Function
  18.5 Interpretation Facility of Results
  18.6 Improvement on User Interface
  18.7 Linking with Available Packages
  18.8 Global Data Base System
  18.9 Improvement on Computing Environment

REFERENCES

Appendix I MAIN FEATURES OF STEM
  I.1 Introduction
  I.2 Model Formulation
I.3 Hierarchical Structure Building  
I.4 Knowledge Presentation  
I.5 Interactive Program Execution  
I.6 Graphic Capability  
I.7 Output Analysis  
I.8 Time Manager  

Appendix II STEM ENVIRONMENT  
II.1 Introduction  
II.2 Operating Environment  
  II.2.1 LOOPS  
  II.2.2 Libraries  
  II.2.3 Monitoring  
II.3 Menu Operating System  
II.4 Other Facilities  
II.5 Modelling Logic  
  II.5.1 Discrete Event Modelling  
  II.5.2 Modelling Logic in STEM  

Appendix III OBJECT-ORIENTED PROGRAMMING TECHNIQUES  
III.1 Introduction  
III.2 Object-Oriented Programming  

Appendix IV SIMULATION MODEL EXECUTION SPECIFICATION  

Appendix V SPECIFICATIONS ON CNODE MODIFICATION  
V.1 Introduction  
V.2 Add a Simple Node on a CNode  
V.3 Add a CNode on a CNode  
V.4 Modify a CNode  

Appendix VI GEC ALSTHOM LARGE MACHINES LTD  
VI.1 Introduction  
VI.2 System Configuration  
VI.3 Operation of the Company  
  VI.3.1 Company Level  
  VI.3.2 Cell Level
Chapter 1
INTRODUCTION

The growing competition both national and international has driven the production goals of nearly all industrial organizations in the world towards shorter production runs and quick response to market changes, on a variety or range of products with high quality and competitive costs.

To meet these requirements, the factory of the future will be characterised by flexible manufacturing and integrated use of computers in all areas of an industrial organization associated with production. Therefore, the integrated organization as a whole is needed to guarantee the performance of the system. One approach to this is to view generic functions of industrial organizations as a hierarchical model, including enterprise, facility, shop, cell, station and equipment levels suggested by the proposed ISO enterprise model [112].

Computer simulation is considered as a fast, cost effective modelling approach which is used to provide the design and analysis aid. Current simulation systems, however, are restricted to lower level modelling, such as at cell or station level. Therefore, a multi-level modelling technique is needed to predict, evaluate and assess the performance of the whole factory.

Conventional multi-level modelling methods can only provide a 'walk-through' environment and the documentation to the system. They cannot provide the output to assess the dynamic performance of the system. A multi-level simulation modelling technique is needed to generate dynamic statistics and analytical output.

Due to the fact that it is impossible and unnecessary to simulate every aspect of the complex manufacturing system, a certain degree of simplification is required. Multi-level simulation modelling system is, therefore, built to provide fast and approximate modelling to generate realistic and effective system performance output.

Artificial Intelligence techniques are considered as potential tools for effective and easily built multi-level simulation systems due to the way they organize the programme.

To help create this knowledge-based multi-level simulation model, a systematic and structured analysis method is needed to display the functional structure of the model clearly and logically prior to building the simulation model.

The main subject of this thesis is, therefore, to develop a new integrated modelling method which combines both structured analysis methods and knowledge based simulation methods. It represents the functional structure of an industrial organization pictorially and
hierarchically and based on this builds a multi-level simulation modelling system to mimic
the real operation of the system from top level down to the detail level including both
information flow and material flow using artificial intelligence techniques.

The work begins with an extensive literature overview of analysis of industrial
organizations and computer integrated manufacturing. In Chapter 2, it also goes through
the main contributions of various modelling systems, especially simulation modelling and
structured analysis methods.

The detailed analysis of industrial organizations on structure and performance is
conducted in Chapter 3 for providing the general skeleton. Pertinent issues to production
planning and control are discussed in Chapter 4. The manufacturing challenge and modelling
challenge, the requirements for integrated methodology and the framework of the project
are described in Chapter 5.

In Chapter 6, the scope of the research is critically identified, defining the work area
and procedure of the research. In order to represent the industrial organization logically,
correctly and clearly, the basic functions of this system which are presented by using IDEF0
technique are discussed critically in Chapter 7. The original methodology used in the
hierarchical modelling of manufacturing organizations is identified in Chapter 8. The
research into design and structure of knowledge based modelling systems is introduced in
Chapter 9. The expert knowledge required and the operating rules and data needed for the
modelling system are presented in Chapter 10. The user friendly interface of the simulation
model is described in Chapter 11. Chapter 12 presents the general analysis of output of the
simulation system.

Chapter 13, Chapter 14, Appendixes VI, VII and VIII will introduce a case study of
a real manufacturing company, which is used for testing the methodology and evaluating
the original modelling system. In Chapter 15, the merits of the modelling system are
critically assessed based on the case study. Chapter 16 concludes key issues associated in
developing the integrated methodology from criticised point of view. Chapter 17 draws the
clear conclusion on the research project. Finally Chapter 18 recommends the possible
research work for the future. Figure 1.1 shows the general structure of this thesis.
Figure 1.1 General Structure of the Thesis
Chapter 2
LITERATURE SURVEY

2.1 Introduction

This literature review is aimed at examining background concepts relevant to the research project described in this thesis. The contributions are evaluated, existing problems identified and potential improvement methods are discussed. The main emphasis is placed on the general analysis of manufacturing organizations, followed by a description of state of the art of CIM. Then, it moves onto the discussion of modelling methods inclusive of Artificial Intelligence techniques and structured analysis methods.

2.2 Industrial Organizations

2.2.1 Introduction

The primary aim of reviewing the concept of industrial organizations in this literature survey is to identify significant system characteristics and major issues related to this research work. The major topics include the changes faced by manufacturing organizations, categorisations and structural representation of manufacturing organizations.

A model proposed by the ISO [112] is used in this thesis to provide a basic hierarchical functional model structure to present activities and the relationship between them in a conventional manufacturing organization in a systematic way.

Detailed discussion on the categorisation, structural presentation and system performance judgement of manufacturing organizations are found in Chapter 3. The discussion on the issues of forecasting, production planning and control, and material control are found in Chapter 4 with the emphasis on the production planning and control.

2.2.2 Changes Faced by Industrial Organizations

Industrial production has been playing an important role in the development of the world’s environment since the end of the 18th century with the replacement of human energy
by machines. Through the use of electricity linked with inventions for its utilisation, the present industrial development has been characterised by the development of information technology [47].

The changes in industrial production have been influenced by various factors to differing degrees. The influences mainly come from market, technology and society [194]. Even if the factors influencing industrial organizations differ from sector to sector and from country to country, so that both the effects of these factors on organizations and the approaches to be taken for coping with the effects may be very different, it is, nevertheless, possible to give a summary of some general tendencies [47] [86].

**Market Competition.** Nearly all industrial organizations are finding themselves faced with increasingly keen national and international competition [194]. The possible causes for this may be a growing number of competitors within a market segment, the saturated markets, cyclical influences, and the building up of excessive product stocks [47] [86].

**Technological Changes.** New technology and new application of existing technologies offer new opportunities to companies and require new skills from workers and managers. All these new technologies can be divided into two groups: software and equipment.

With the development of computers, more and more powerful software packages have been created and put into use. Most of them are as follows: Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), Computer Aided Manufacturing (CAM), CAD/CAM, MRPII, etc [233].

The development of new equipment started with the introduction of computer-controllable machines, including metal processing machines, industrial robots, inspection stations, material handling systems, flexible manufacturing systems etc.

The concept of computer integrated manufacturing systems has brought the potential development of integrating a series of interrelated functions with the help of computers into a whole automated manufacturing system. The rapid development of software and hardware has made it possible to link them together to generate a more efficient system. For the detailed discussion, refer to section 2.3.

**Other Changes.** Besides the technological changes and severe market competition, there are some other changes which may influence the production of industrial organizations. Economies no longer show steady growth. Government regulations and controls have been cited as one of the reasons for decreased productivity in the manufacturing sector. Com-
munity groups and environmentalists, on the other hand, are bringing increased pressure to bear on manufacturers to improve the reliability and safety of their products and manufacturing processes [194].

2.2.3 Objectives of Industrial Organizations

The growing competition, national and international, and all other changes, at least to some extent, have led to an almost explosive increase in the variety of products, smaller batch sizes, short delivery periods, fluctuations in quantities, higher demands on the quality of the product and the quality of the after-sales services [86] [194].

Therefore, an industrial organization in a market economy must continually drive its production to cope with the above changes if it is to survive. The two principles of knowing the customer and competitively supplying products to that customer define the primary purpose of the organizations. According to Gerelle and Stark [86], the objectives of industrial organizations can be categorized into three types: marketing, innovation, and integration.

The marketing objective provides the industrial organization with its primary goal. It directs the company within the market, in its relation to customers and competitors, and helps management focus on the various external pressures which the company faces.

The innovation objective for the industrial organization addresses the need to offer customers a product that is positively differentiated from those of the competitors. Especially in recent years, average product life times have declined, while average product development times have increased.

The integration objective is introduced to address the resources of the company, integrating the marketing and innovation objectives. It is because of the rapidly changing environment that these two objectives do not span all the activities of the organization.

This integration is clearly a top management responsibility, since only top management has the power to ensure that on a company-wide basis, the company’s resources are used as efficiently as possible.

2.2.4 Categorisation of Industrial Organizations

Various types of organizations exist in manufacturing industries according to the volume of products, transformation type, the approach to dealing with customer orders etc. Since different types of organizations may contain different features, they use different
methods to manage the system. It is of significance to understand the general characteristics of different production types [231] [266] [71] [33] [283]. The detailed discussion on the characteristics of different types of production can be found in Chapter 3.

2.2.5 Structural Representation of Industrial Organizations

The structure defines tasks and responsibilities, work roles and relationships and channels of communication [173] [107]. Usually the basic structure of an industrial organization depends upon the nature of the business, the size of the company, and the complexity of the problems faced [5]. The organization structure can be designed in terms of the line, functional, staff or lateral. The most commonly used forms of organization structure are the line organization and the line and staff organization. Two examples are listed in Figure 2.1 for comparison between them.

In the line organization, the president handles all problems that arise, whether they have to do with production, sales, finance or personnel [5]. In the line and staff organization, the difference from the first case is that additional specialists are involved to share certain responsibilities. At present, many organizations are structured in the extended addition of staff people in the line and staff organization. Therefore conventional manufacturing organizations are now typically structured around a hierarchy of managers, engineers, operators, clerical and support staff [162]. Figure 2.1 has depicted one example of this type. Other examples show the structure of the system in some aspects, such as computer integration, data flow structure [207] [208] [173].

2.2.6 ISO Model

Since a hierarchical model comprising multiple layers provides a common structure for many of these different views of manufacturing as they all embody a sequential concept, a distributed structure and facilities for feedback control, the ideas and concepts developed are being examined by the International Standards Organization (ISO) who are evolving reference models for manufacturing [112]. Figure 2.2 illustrates the hierarchical levels of the proposed ISO model.

It is typical of the hierarchical structure that each entity of one level is comprised of several entities of the lower level [233]. Therefore, the hierarchy is represented in a triangle. This hierarchical model, on the other hand, shows the different levels of control. These levels must be linked to form a single seamless automated system. Figure 2.3 outlines the basic elements and computer types of each level and the main functions [162].
In contrast, by using different terminology but with the same concept, Figure 2.4 summaries important computerised functions in the CIM areas of production planning and control, design and manufacturing [233]. Thus it is evident that this hierarchical structure of a manufacturing organization can be used to represent any type of system.

**Enterprise** level involves strategic planning functions for establishing product and production strategies. Some other comprehensive business functions, such as payment, wage and salary calculations, and controls are also located at this level [233] [123].

At **facility** level, three major functional areas involve: manufacturing management, information management and production management [162]. Two major sub-systems are involved in manufacturing engineering, CAD which is used to develop geometry specifications and bill of materials for assemblies, parts, tools, and fixtures, and process planning to prepare the specifications of all operations. Information management provides user data interfaces to support necessary administrative or business management functions: such as finance accounting and personnel management. Production management identifies production resource requirement, determines excess production capacity and summaries quality performance [162] [233].

The **shop** level is responsible for co-ordinating the production and support jobs on the shop floor [162]. In the production planning and control area, the detailed scheduling of orders received and the associated task of allocation to various sub-orders are carried out.

The **cell** brings some of the efficiency of a flow shop to small batch production by using a set of machines, tools and shared job setups to produce a family of similar parts [162]. In a broad sense, these cells may be flexible production systems, processing centres, assembly islands, etc [233]. Technical functions are the DNC operation and the control of area-specific transport and storage systems.

The activities of small integrated physical grouping of shop floor equipment are directed and co-ordinated by the **workstation** level of control.

The **equipment** level. These are "front-end" systems that are closely tied to commercial equipment or industrial machinery on the shop floor. Equipment controllers are required for robots, NC machines tools, co-ordinate measuring machines, delivery systems and storage/retrieval devices.

The structure of a computer integrated manufacturing organization can be viewed in a hierarchy shown in Figure 2.2. However, the general representation function of the ISO model can be applied to conventional industrial organizations as well as to show the interactions within a company in a model.
It should be realised that the ISO model only provides the information data passing between different levels or within the same level. There are other different viewpoints to look at a manufacturing organization model, such as enterprise, computation, engineering and technology.

The problems of production control and planning of such a system should be approached by using a hierarchy of control as well where major aggregate decisions are made at the highest level and these decisions are gradually broken down into more detail as they pass through the lower levels. The relationship between them is represented in Figure 2.5. Several examples introduced below have employed this concept.

A research project, the Automated Manufacturing Research Facility (AMRF), is being established at the National Bureau of Standards (NBS) in America. Reported by Bloom et al. [39], it provides a testbed where measurement research of computer integrated manufacturing systems can be performed. A hierarchical control system emulator (HCSE) has been developed that allows the system to be designed and tested before implementation on the actual hardware. The complex planning and control problems inherent in the AMRF have been broken down into a series of levels in a planning and control hierarchy, namely facility, shop, cell, station and equipment [179] [228].

Developed by Computer Aided Manufacturing International Inc. (CAM-I), the factory management programme, which is concerned with the design and implementation of a factory management system to manage production efficiently, has designed a computer hierarchy of control to break down the complex problem of planning and controlling shop floor activities into a series of smaller modules. The hierarchy consists of four levels: factory control system, job shop level, work centre level and unit/resource level.

Jackson and King [117] introduce ADEPT (Advanced Distributed Environment for Production Technology) which is envisaged as improving the approach to the integration of manufacturing systems during the 1990s. ADEPT models the physical system by breaking it down into its component processes. The model emulates the three primary control levels, namely those at factory, department and cell level. Figure 2.6 represents a high level view of the ADEPT model. ADEPT provides a complete set of development tools for use when implementing distributed control systems. It also includes a simulator which will enable the verification of the system design and modelling process for a large number of varied manufacturing processes. The simulator models each control level and allows various scenarios to be tried and tested. Figure 2.7 shows the way in which the simulator is integrated into the system.
2.2.7 Major Functions in Industrial Organizations

The general functions involved in an industrial organization according to Rembold *et al.* [213] include marketing research, long-range planning and forecasting, capital equipment and facility planning, customer order servicing, engineering and design, manufacturing process planning, marketing, production order planning and manufacturing monitoring and control, purchasing and receiving, inventory management, maintenance and accounting. The individual functions (above) of a manufacturing organization and their interactions are represented in Figure 2.8. The activities shown are typical for a company manufacturing products in small to medium-size production runs [213].

Due to the fact that there is no standard terminology for most activities, this structure and the terminology used may not be representative of all companies. However, the major functions involved in a company usually include sales and marketing, engineering, production, personnel, finance etc [155][140]. The detail discussion can be found in Chapter 3.

2.2.8 Production Planning and Control

For the machinery of production and the organization of production to be effective, there must be a decision-making apparatus which determines, during any period of time, what products will be produced, how much will be produced, and when they will be produced [71]. However, the real situation always causes imbalance between production plan and the real production. Therefore, production control is an approach to measure these changes, and to manage the production with less cost and fast response [200].

Production planning is often referred as master production planning and master production scheduling. Master production planning usually deals with output in broad terms. The major strategy variables associated with production planning for variable demand are the production rate, the inventory level, the work force size, extra shifts, overtime, the product mix and subcontracting. Refer to Chapter 4 [266].

Master production scheduling, on the other hand, translates the production plan into specific product models and specifies the time periods for their completion. It generates more detailed materials requirements and capacity planning information which enable it to balance demands against resources.
Compared with master production planning, the time period is short, in weeks or days and the planning horizon is short. The master production planning deals with aggregate planning for total output, while the master production schedule relates to specific products or end items.

In [71], Elsayed et al. have introduced several methods to generate aggregate production planning. In [63], an aggregate production planner has been built to generate production quotas for individual products or groups of products to be manufactured over an extended planning horizon by using mathematical formulations.

2.3 Computer Integrated Manufacturing (CIM)

2.3.1 CIM Definition

In contrast to conventional manufacturing organizations discussed above, computer integrated manufacturing concepts bring new ways to look at manufacturing systems. Various definitions of CIM have been given concerning different aspects of CIM.

Boaden et al. [40] have categorized definitions of CIM into ten categories, which are further combined to form three general classes: total organization definitions, information systems definitions and single facet definitions. The intensive analysis and the statistics results of these ten categories have been introduced and the conclusion is to view the organization as part of a total business unit. This means that CIM must be an overall concept which takes account of every aspect of the business, which ties all such aspects and organizational functions together into an integrated system, where all necessary data can be accessed easily by those who need it.

Bunce [48] provides the CAM-I definition of CIM as a series of interrelated activities and operations involving the design, materials selection, planning, production, quality assurance, management and marketing of discrete consumer and durable goods.

Groover [93] defines CIM as the application of computer technology to all of the operational functions and information processing functions in manufacturing organizations from order receipt through design and production, to product shipment.

In contrast, Ranky [207] points out that CIM is concerned with providing computer assistance, control and high level integrated automation at all levels within manufacturing industries, by linking "islands of automation" into a distributed processing system. Further discussions on the CIM definition can be found in [106] [199].
In summary, certain levels of awareness of CIM exist with regard to the understanding of CIM. In objectives of this project, CIM is considered as an overall concept concerning all tied aspects of the business and organizational functions.

2.3.2 Methods of Achieving Integrated Manufacturing

Although there exist various CIM definitions, the main aim of CIM is to bring together all aspects of manufacturing into a unified, automated system under computer control with a direct link to other functions such as design, process planning and production scheduling [107]. Figure 2.9 depicts the structure of such a system.

According to Scheer [233], CIM refers to the integrated information processing requirements for the technical and operational tasks of an industrial organization. Figure 2.10 represents these two tasks [233]. Such integration can be achieved in two ways, one is to improve the existing industrial organization, the other is to design a new system.

Hollingum [107] reported that to some extent this integration is already under way in the sense that the flexible manufacturing systems have been introduced into the industries. FMSs can carry out a sequence of operations on a family of different products. Robot assembly cells are also beginning to appear in some companies, and the use of robot devices in such operations as welding and spray painting is well established, but the operation of these "islands of automation" is controlled by one or more programmable controllers or computers and nearly all of them are self-constrained.

Concerning design, process planning, and production scheduling, there have been well established commercially available software systems, such as CAD, CAPP, CAM, and CAD/CAM [199] [228].

As a result, how to well integrate these islands of automation and available software modules is the main job for the system integration. The current researches have indicated that the proper communications in an well organized manufacturing organization play an important role [107].

The other way that a high degree of integrated manufacturing can at present be achieved in a way that makes economic sense is to start with a new factory and new equipment, but unfortunately there are a few cases where this has been done.

Integration of manufacturing organizations can be seen as an evolving process [136]. These four quality levels of integration are growing originally after initializing actions. Both physical and communication integration have been initialised by General Motors MAP. The objective of the AMICE project is to initialize the process of Application Integration and Business Integration through the development of CIM-OSA [8] [136].
2.4 Modelling Techniques

2.4.1 Introduction

Until now analysis of manufacturing organizations has been carried out. The requirements for and potential ability of integrated manufacturing systems have been identified. The problems, therefore, posed in both improvement of conventional manufacturing systems and the design and control of new integrated manufacturing systems are difficult to solve [262]. On the other hand, the size and complexity of manufacturing systems is matched by their cost, so there is no room for error in the system designs that are going to be built [208]. As a result, there exists a great demand for building models which are used as a design tool. In this section, the functions of modelling techniques are first discussed, then the assessment and general classification of modelling methods are conducted. Finally the emphasis is placed on the discussion of current modelling methods, single-level, multi-level and simulation methods.

2.4.2 Functions of Models

A model, here, is defined as a collection of various components of the system interacting to produce the behaviour of the system [111]. It can be manipulated to reveal the consequences of decisions more readily, or more cheaply, or with less risk than would be involved in direct manipulation.

The broad purpose of modelling, then, is to provide a "handle" on the problems that are faced [250]. It usually provides an aid in explaining, understanding, experimenting with or improving a system. It is also an important aid for training and instruction [239].

To perform these aid functions, the model should be accurate, inexpensive, easy and fast to build. It also should highlight the important aspects of the system.

2.4.3 General Classification of Modelling Methods

There are several different ways to classify the modelling methods used to model manufacturing systems, mainly according to the different aspects of the system and research emphasis on the system [262].
According to Solbery et al. [250], three broad classes of models: physical, simulation, and analytical with respect to the form taken. Aside from the form, any model can also be classified to how it deals with system objectives, time and variability. Suri [262] specifically differentiates models based on the various decisions that can be made by the models in flexible manufacturing systems. Figure 2.11 shows the classifications above.

In Wang's work [274], the modelling methods are divided into approximate modelling, simulation based modelling and knowledge based modelling, simply depending on the complexity of logical details and the level of intelligence contained within the model (Figure 2.12).

Another type of classification of modelling methods can be found in [73] reported by Erkes, based on model classes, levels of abstraction, and the stages in the system development life-cycle (Figure 2.13).

The above examples show that a great number of modelling methods are currently available for the design and analysis of manufacturing facilities. The classification scheme which is proposed for these methods can be made along several dimensions and criteria. Therefore, it is impossible to discuss them effectively along one aspect. Taking Petri nets for example, it is a kind of evaluative model using graph modelling formalism to provide the approximate modelling results.

For the purpose of this thesis, three broad types of modelling methods are distinguished and discussed. They are general single-level modelling methods, multi-level modelling methods and simulation based modelling methods, depending on the scope of the model (Figure 2.14). In section 2.4.4, several single-level modelling methods are discussed, while multi-level modelling methods are highlighted in section 2.4.5. In section 2.4.6, the simulation modelling methods are extensively discussed.

2.4.4 General Single-Level Modelling Methods

2.4.4.1 Introduction

Single-level modelling here means the modelling of one aspect of manufacturing systems in one broad dimension which cannot be presented into the hierarchy. The techniques to be discussed include static allocation method, mathematical programming, queueing network, Petri nets and heuristic algorithms.
### 2.4.4.2 Static Allocation Method

Static allocation method simply adds up the total amount of work allocated to each resource, and estimates the performance from this total. A common example would be to add up the processing time of all operations assigned to a resource in order to estimate its utilization [262]. One example reported by Wichmann [279], is a computer programme called SPAR (System Planning for Aggregate Requirement) which is part of an advanced manufacturing system design tool. SPAR is aimed at setting up targets for the manufacturing system capacity and performance, such as production equipment utilization, work-in-progress levels, pallet cycle times. It gives the input data, generates output with the required number of machines, pallets, transporter capacity and feasible production levels [152].

### 2.4.4.3 Mathematical Programming

Mathematical programming is used to find the best or optimal solution to a problem that requires a decision or set of decisions about how best to use limited resources to achieve a stated goal or objective [228]. It usually includes linear programming, non-linear programming and dynamic programming. As a well-established quantitative method, it is widely used in decision-making in manufacturing systems.

Linear programming has been employed by Afentakis [4] to determine the optimum physical layout of flexible manufacturing systems. Gaskins et al. [85] use the zero-one programming method to design an optimal flow path for automated guided vehicle transport systems. The use of linear programming models for medium term production planning is widespread. It may be used as a single stage planning system to transform a yearly sales plan into a feasible production plan, which indicates for each item in which subperiod and what amount should be produced on which machines such that a given objective function is at its optimum [252] [119].

Non-linear programming models, according to Meredith et al. [167] are similar to linear programming models except that the objective function and constraining equations are not required to be linear functions of the decision variables. An initial model to determine the number of circulating kanbans and hence the inventory level in a kanban system reported by Bitran et al. [38] is nonlinear integer in nature. Fisk and Ballou [77] developed a dynamic programming approach for determining optimal lot sizes in the single-product unconstrained capacity environment.
2.4.4.4 Queueing Network

The queueing network theory, established by Jackson [115] [116] assumes that a network of queues consists of a certain number of service centres, each of which has one or more identical servers and a customer may leave one centre and proceed to another. Jackson [116] and Gordon and Newell [89] develop the equilibrium distribution of states of a class of general networks.

Based on the queueing network, models which simulate manufacturing systems have been developed to analyse the performance of the system in which machine stations and transporters are considered as service centres in a queueing network, while parts as customers flow in the network. Graham [91] concludes that closed models are often better a representation of actual systems, while open models are usually easier to solve than closed models. More discussions can be found in [65] [220] and [54]. The detailed discussions on models: CAN-Q, MVAQ, PMVQ, MANUPLAN can be found in [249] [271] [131] [261] [237] [260] [263].

As concluded by Suri [262], CAN-Q, MVAQ, PMVQ, and MANUPLAN are becoming popular in manufacturing system design and operation as a good compromise between efficiency of the model and the accuracy of the predictions. Queueing network models are particularly useful in situations where management requires quick turn-round on preliminary design.

2.4.4.5 Petri Nets

The Petri Net is a kind of graphical-mathematical technique which is used for the modelling of multi-event synchronization between concurrent and cooperative processes [166]. The most important modelling property of Petri Nets is the ability to represent concurrence and conflict (Figure 2.15) and therefore to enforce deadlock freeness in concurrent processes and mutual exclusion for conflict solution [190] [212]. Now the Petri Net method can answer both qualitative and quantitative questions.

Merabet [166] uses Petri Nets for representing process parallelism and asynchronism for flexible control of machine tools in a job shop environment. Coloured Petri Nets have been emulated in hardware using a uni-processor operating system, and arguments to control nets with command signals and transformations.

Pivi et al. [193] developed FIRST, a software package which is able to model and simulate any manufacturing/assembly system, based on Petri Nets representation. Another example of using Petri Net models to efficiently analyse problems of real-time control as
well as the performance evaluation of production systems has been reported by Dubois and Stecke [68]. Based on the Petri Net modelling capabilities, various realistic aspects and components of production processes, such as flowshop, blocked machines, assembly process, and the FIFO control rules can be modelled.

Tzafestas [268] explored the possibility to combine the Petri Net and knowledge-based methodologies for treating the modelling, simulation and control of manufacturing systems. He concluded that concurrence is one of the major features of discrete manufacturing systems, and Petri Nets are particularly suitable for graphically representing and analysing discrete concurrent systems. Some of the Petri Nets extensions developed over the years are timed Petri Nets, Coloured Petri Nets, Structured adaptive Petri Nets, Modified Petri Nets and the detail discussion of them can be found in [268] [160] [124].

2.4.4.6 Heuristic Algorithms

Heuristic algorithms can be considered as a kind of modelling method in which some problems could be modelled in detail while other problems are ignored. It can provide efficient determinations which are acceptable and feasible in a reasonable period of time [64].

Galante et al. [84] use heuristic algorithms to solve the group scheduling problem for flow-shop scheduling. Ballakur et al. [16] use the heuristic approach for solving part family/machine group of cellular manufacturing system. The distinguishing feature of this method is its consideration of several practical criteria, such as within-cell machine utilization, work load fraction, maximum number of machines that are assigned to a cell, and the percentage of operations of parts completed within a single cell. The computational results show that the heuristic performs well with respect to more than one criterion. The heuristic is flexible, consistent and efficient [259].

Both De Souza [64] and Zhang [287] use the heuristic approach to model the tool flows in flexible manufacturing facilities for prismatic and cylindrical parts respectively. As shown in Figure 2.16, the tool flow in a factory can be represented as a hierarchy of different levels with its own focal point of tool supply. The Primary Tool Store, the Secondary Tool Store and the Central Tool Store are the corresponding focal points to stand-alone machine, cell and factory.

Heuristic algorithms have been used by other researchers, Iwata et al. [114], Stecke [255] to investigate the scheduling of FMS. More applications of this method can be found in [256] for solving the problem of planning machine replacements of a serially dependent production system.
2.4.5 Multi-Level Modelling Methods

2.4.5.1 Introduction

Multi-level modelling methods, in contrast to single-level modelling methods, can provide an overall performance analysis of manufacturing systems. They are based on the concept of top-down partitioning of system functions, which means to break complex systems down into a series of less complex sub-systems which are, more or less, easy to deal with [18]. The mainly used conventional methods are GRAI and IDEF techniques.

2.4.5.2 GRAI Method

The GRAI method was designed by the "Groupe de Recherche en Automatisme Integre" at the University of Bordeaux to describe the decision-making system in a factory [13]. It provides an analysis of the decision-making network in a plant and reveals the various activities taking part in decision-making in production management as well as the interaction between these activities.

It consists of a break-down of the production management system where the general structure can be viewed as a combination of decision systems, information systems, and physical systems [168] of a given factory into "activity centres". This break-down follows two poles: the horizontal axis, which corresponds to the decomposition of the physical production system into execution centre, and the vertical axis which corresponds to the hierarchical decomposition of the decision system within a hierarchical structure. The information sub-system is a link between decision and execution centres [192]. One example is shown in Figure 2.17 [13].

Each activity centre is itself broken down into activities. This second break-down is represented by a "macro-network" on which the interaction between activities is represented in Figure 2.18.

The GRAI-Grid is actually a graphical tool to represent interrelations from a top-down view. As Meyer et al. [168] concluded that GRAI-net formalism has been proved to be a valid tool for knowledge acquisition regarding hierarchical planning, subgoal interaction, and constraint interdependence. However, this approach must be oriented according to the specification, which means that it is not transferable.
2.4.5.3 IDEF Techniques

IDEF (Integrated Computer-Aided Manufacturing DEFINition) was developed as part of the U.S. Air Force ICAM program [248]. IDEF is intended to evolve as the ICAM program proceeds, beginning from an initial version, called IDEF0, to another two versions IDEF1 and IDEF2.

The basic concepts for IDEF techniques are to understand a system by creating a model that graphically shows activities, to distinguish what functions a system performs from how the system is built to accomplish those functions, to structure a model as a hierarchy with major functions at the top and successive levels revealing well-bounded details and to establish an informal review cycle to "proof-read" the developing model and record all decisions in writing [18] [157].

A total IDEF model views the manufacturing system as consisting broadly of three integrated structures: activities, information, and dynamics. However, in any one model, the emphasis will always be on one of these aspects, which means these structures are modelled individually using clearly defined diagraming disciplines, IDEF0, IDEF1, and IDEF2 respectively [18]. These three methods are intended to compliment each other.

IDEF0 developed by SofTech, which was based upon SADT (Structured Analysis and Design Techniques), is a functional modelling methodology [75]. This method decomposes complex systems, using a hierarchical top-down approach and provides a means of understanding activities and their interrelationships. At the highest level, a single box represents the system as a whole, and establishes the context of the model. The interface arrows represent the complete set of external interfaces to the system as a whole. Figure 2.19 is a basic structure of IDEF0 [157].

In Figure 2.19, an activity is anything that can be named with an active verb phrase. An input is converted by the activity into the output. A control describes the conditions of circumstance that govern the transformation. A mechanism may be person or device which carries out the activity [102] [59].

The system is then examined level by level to provide lower level diagrams, which define the model in further detail. Figure 2.20 shows the structure of a system in hierarchy.

It has been argued that IDEF0 can be applied to high level planning system design [157]. Since the specification of a manufacturing system in the IDEF0 methodology can exist and the current system exists (as is) or as the future system in envisioned (to be). If the analysis is intended to provide a basis for future system design, it would probably be
generic so as to avoid the constraints of the existing system. However, if the analysis is intended to permit investigation of how well the existing system will integrate with future design, it will most likely be site specific.

In the report given by Hicks et al. [102], IDEF methodologies have been used for modelling CAPM systems. It is argued that the performance of CAPM systems critically depends upon the "model" of reality, stored as data within the system, being realistic. The IDEF methodologies are helpful at the design stage, but operational factors also have to be taken into account.

Colquhoun et al. [59] reported that IDEF0 can be used to link design and manufacture in a CIM environment. This linkage can be established by use of computer aided process planning. The IDEF0 "as should be" model has defined the information flow necessary to support each activity and thus a functional and information flow requirement can be established for the computer system at each level defined in the ISO proposed model [59].

IDEF1 is an informational modelling methodology. An information model enables the structure of the information required by the systems to be understood, and this helps in designing the databases that are required to support the functions described by classifying the system's entities, defining the relevant attributes of each activity, identifying the relevant entities and specifying and classifying these relationships [248] [157] [102].

The IDEF2 modeling technique aims at describing the time-varying behaviour of manufacturing systems and is therefore more physical, compared with IDEF0 and IDEF1, where the modeller's task is primarily 'conceptual analysis' [157] [102]. An IDEF2 model can be seen as a network combining logical representation with the actual movement of material and its associated information in the factory. These characteristics render the technique pertinent both as a descriptive tool and as an analysis tool. As a descriptive tool, it can be used to identify the components of a system which effect the system's behaviour or cause it to change over a period of time. Once a system is built, IDEF2 can be used as an analysis tool to experiment with the model via simulation to draw reference about the real system. Therefore, IDEF2 models take different forms for different purposes, but use the same data and information base treated by IDEF1 [59] [157].

At the heart of an IDEF2 model lies a set of diagrams called entity flow diagrams (EFD). An EFD portrays the flow of parts, tools in the system together with information such as machine status and queue size. An example of an entity flow network is depicted in Figure 2.21 [75].
In summary, IDEF methodology, for new systems, may be used to specify the requirements and functions, and then to design an implementation which meets the requirements and performs the functions. For existing systems, IDEF can be used to analyse functions it performs, introduce new technology into the system and identify problem areas within the system. These three methods can be used singularly or in combination to provide a comprehensive description of any manufacturing facility.

As analysed above, both GRAI and IDEF mostly use techniques to graphically present the activities in manufacturing systems and their relationships. They, however, only provide a 'walk-through' environment and cannot provide performance outputs to evaluate or assess the performance of systems. Therefore, multi-level simulation techniques are needed to analyse the dynamic behaviour of the manufacturing systems. The following section will discuss the general simulation methods.

2.4.6 Simulation Based Modelling Techniques

2.4.6.1 Introduction

So far analysis of single-level and multi-level modelling techniques have been carried out. However, simulation, as another powerful modelling tool, is more and more recognized.

Simulation, according to Shannon [239] is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies for the operation of the system [264]. Therefore, a simulation model is a simplified representation of real life which allows the understanding and solution of a problem to be achieved by a trial and error approach.

According to the time changing patterns, simulation models are divided into three types:

--discrete event simulation
--continuous simulation
--combined simulation.

In the context of modelling manufacturing systems, simulation refers specifically to computer-based discrete event simulation because the activities in manufacturing systems are predominated by discrete changes.

Simulation can be both single-level and multi-level modelling, depending on the capability provided by the particular software packages.
The following section will mainly discuss the dynamic modelling building methods, including conventional simulation modelling methods, AI-based simulation methods and generalised manufacturing system simulators. Then, the structured analysis methods are introduced in section 2.5.

2.4.6.2 Conventional Simulation

In the context of this thesis, conventional simulation means that the programming of the simulation model is written by high level computer languages, or by a simulation language or package built on these high level computer languages, like FORTRAN, PASCAL etc. Both single-level simulation and multi-level simulation can be found in conventional simulation.

There are four approaches available to build simulation models in conventional simulation, depending on how a program is written which embodies all of the interactions of the entities [191] [110]:

-- the event approach
-- the activity approach
-- the process approach
-- the three phase executive approach.

2.4.6.2.1 General Purpose Computer Languages

From the historical trends of development of simulation software, the first generation were usually written in general purpose computer languages [280] [191] and they are still in use today.

The creation of such a simulation model involves writing routines such as interface operation, control of the simulation. Using this approach, the programmer who writes the programs knows all the details of his own programme, but it takes a long time to write and it is hard to debug. However it is difficult for others (perhaps the user) to understand.

2.4.6.2.2 Simulation Softwares

With the development of computer technology, greater understanding of the simulation concept, and increased application demand for computer simulation [1] [253] [139], the general purpose simulation languages and packages had been developed since the early 1960's [280].
Most commercial simulation software and their relationships are presented in two family trees mainly developed in the U.K. and the U.S.A [51]. The new versions of these software and also new simulation languages are being developed at present mainly because of the general need for simulation or some domain specific requirements like manufacturing systems.

Among them, typical examples are GPSS [24] [235], GASP [189], ECSL [24], SIMSCRIPT [225], SLAMII [129] [201], SIMAN [176] [20], SIMULA [35], TESS [254] [253], and SMALLTALK [137] etc. Detail discussions on these simulation languages can be found among others [188] [1] [195] [230] [19] [21] [188] [110] and [60].

In Figure 2.22, the general characteristic of these simulation applications are displayed based on a general predefined set of criteria. Some simulation languages like GPSS have several implementations, e.g. GPSSII, GPSS/362, GPSSV, each offering enhancements over its predecessor [235]. GPSS/H, is an upwardly compatible superset of IBM’s GPSSV. The improvement in execution speed; the ability to interactively monitor an on-going simulation; the ability to read from and write to external files, have been significantly enhanced by GPSS/H.

Other simulation softwares, such as SYSMOD, PASION, SIMPLE-I, FORSSIGHT, NETWORKII.5, SIMUL-R, HYBSYS, DESIMP, INSIGHT, SIMCAL, Demos, SIMPLEX-II, and CSIM are discussed in [52] [202] [58] [108] [226] [257] [34] [215] [269] [57] [74] and [236].

2.4.6.3 AI Based Simulation

The section above simply outlines the available conventional simulation software and the main features of them. However, the user oriented development trends for the future have been concluded by Wichmann [280] as the intelligent integrated simulation environments incorporating knowledge-based systems. It means to find ways to incorporate Artificial Intelligence (AI) techniques in simulation. In fact, the available knowledge-based simulation systems have proved that they are very powerful in the sense that they aim to be user friendly, goal driven and problem oriented simulation systems.

2.4.6.3.1 Artificial Intelligence in Simulation

Artificial Intelligence (AI) is a field of study that encompasses computational techniques for performing tasks that apparently require intelligence when performed by humans [265]. It is a technology of information processing concerned with processes of reasoning,
learning, and perception [24]. Because of its rapid development and demonstration of its usefulness, AI techniques have been used for a wide range of applications in a number of different fields [139] [24] [134].

The fundamental techniques of AI include knowledge representation, search, logical reasoning, probabilities reasoning, learning etc. A computer system or program which incorporates one or more techniques of AI to perform a family of activities that traditionally would have been performed by a skilled or knowledgeable human is referred to as an expert system [134]. The typical structure of an expert system is shown in Figure 2.23 [97], and types of expert systems are shown in Figure 2.24.

Expert systems are knowledge-intensive computer programs. They contain lots of knowledge about their speciality [97]. The knowledge in an expert system is organized in a way that separates the knowledge about the problem domain from the system's other knowledge. This collection of domain knowledge is called a knowledge base. A program with knowledge organized in this way is called knowledge-based system [97] [223]. The two types of systems can be classified as either expert systems or knowledge based systems according to the scope of the domain knowledge and the size of systems studied [97] [223].

Gaines et al. [82] and Elzas [72] have discussed the relationship between expert systems and simulation modelling techniques and drawn a conclusion that they have a large degree of similarities. Both use various representations to model some aspect of the real world, resulting in a piece of computer software, and both are concerned with modelling: simulation in the field of objects and processes, and expert systems in the field of the human decision making processes [242].

Both simulation models and expert systems can be viewed within a common framework. Both kinds of systems have a state characterization, state transformation operations, and input-output interfaces.

However, there are some differences between them. One is that they use different kinds of information with which the system reasons. Expert systems employ symbolic reasoning [100], while simulation is numeric. The other is that each maintains a different emphasis, i.e., dynamic behaviour for simulation, and logical inference for expert systems.

The similarities can build a strong foundation for combining these two techniques together. These differences, however, can compensate and make the process of modelling, simulation and analysis even easier and more flexible [188] [269] [176]. For example, simulation has developed statistical and graphical output presentation, while expert system can provide explanatory output and natural language input. Time ordering and dynamic processes have been at the centre of simulation modelling, but expert systems have opened up the possibility of integrating traditional dynamic modelling with other symbolic forms.
of state transition representation such as causal inference [127]. The further detail discussions on the relationship between expert system and simulation can be found in [82] [258] [238] [148].

2.4.6.3.2 Knowledge Based Simulation

In contrast to conventional simulation, knowledge based simulation in this thesis means the simulation systems are originally written in Artificial Intelligence languages, such as LISP or PROLOG.

In knowledge based simulation, control and data components are provided as distinct, independent entities (knowledge base and inference engine) permitting the modeller to alter either component independent of the other [148]. Therefore, the simulation construction mainly involves the building of a knowledge base and its management by an inference engine.

In simulation modelling, some of the AI techniques such as rule-based reasoning, frame-based representation and object-oriented programming have been heavily used to drive the simulation [147]. Therefore, the main emphasis put in the simulation system sometimes classifies the knowledge based simulation systems into object-oriented, rule-based or frame-based systems. Rule-based reasoning makes the programming easy to be understood by using an "IF-THEN" scheme. The use of frame-based representation to create the knowledge base allows the incorporation of details of each object and its relationship to other objects in the system more easily. Any object under the frame-based representation scheme could have any number of procedures which can be activated by various means. This means any object knows its behaviour, this is the concept of object-oriented programming.

Until now, many knowledge based simulation systems have been developed and some are under development. Figure 2.25 shows the majority of them according to the different host languages.

LISP, developed at MIT, is a list processing language. Detailed discussion on LISP can be found in [269] [265]. As Vaucher [269] concluded that the power of the language resides in its easy extensionality and all modern LISP systems make heavy use of advanced concepts built on top of the core language, like INTERLISP, LOOPS etc. Most of them add control mechanisms and interpretive capabilities [10] [11] [14] [36] [186].
In PROLOG, problems are described as facts and implications [21]. The descriptive nature of PROLOG has enabled people to use it for simulation purposes and extend it as simulation languages like T-PROLOG, TS-PROLOG, V-GOSS etc. PROLOG uses a backward-chaining mechanism.

In Figure 2.26, the characteristics of most knowledge based simulation software has been presented for comparison.

ROSS, developed at Rand Corporation [161], is viewed as a rule-oriented simulation system in the point of view of encoding knowledge. Rules are specified in an IF-THEN format, describing the behaviour of objects. However, from the language it uses, it can also be classified as an object-oriented simulation system [133]. Objects are used to present the objects in the real world, messages are passed between these objects describing actions to be taken. ROSS provides the flexibility by providing different scenarios through modifications of input, behavioural rules etc. ROSS also has interactive capability in the sense that the simulation model can be stopped during the running to modify the knowledge base, rules or questions about the events happened by tracing them and re-running it again. The statistics output can be displayed graphically. It can also report events.

KBS was developed at Carnegie Mellon University. Similar to ROSS, it is also a LISP-based simulation system, which employs the object-oriented paradigm to describe the real world system to be modelled [211]. Unlike ROSS, KBS uses Scheme Representation Language (SRL) to hierarchically arrange entities in the system. It provides interactive ability to allow the user to examine the model and its behaviour from model creation modification, to monitoring, control and graphical display. Moreover, it develops the facility to model consistency and completeness, and allows the user to build the model at different levels of abstraction. In addition to these, it stresses the automatic analysis of simulation results [2] [142].

EZSIM written in Golden Common LISP was developed at the University of Southern California. It is a general rule-based simulation system that aids the user with little or no knowledge of simulation model building, or computer programming to construct simulation models. EZSIM encompasses the three major models of: user interface, expert system, and program generator [128]. It provides a front end to an existing powerful simulation language SLAM. It operates step by step prompting users for required information in addition to basic data and control parameter specifications input by User Interface. The expert system module of EZSIM uses a special purpose inference engine.

STEM is a general simulation environment and is designed to be extended to support a particular application domain [197] [118]. In STEM, class libraries have been defined for standard static and mobile entities. The specialized classes stored in library files, can be
loaded incrementally on top of STEM, providing a specialized application "personality" of the simulation environment. A most significant feature of STEM is its ability to create, store, maintain and re-use composite nodes (refer to Appendix I and Appendix II).

T-PROLOG [57] is used to provide a logic programming language simulation environment. Unlike the simulation software described above, it is capable of backtracing in time so as to attempt different paths through the simulation. Therefore, this allows for some simple goal-directed simulation, where the user can specify multiple model parameters and goals the model is to achieve. The simulation terminates successfully when the current goal is empty. T-PC PROLOG is the extension of CP (Concurrent PROLOG). A T-PC program consists of a set of guided rules which incorporate a delay clause. Unlike T-PROLOG, it removes the powerful feature, backtracing but does permit CP programs to be non-terminating and yet run in a finite stack space [57]. Multiple goals can be solved simultaneously. While each such goal can be viewed as a synchronous "process" which has its own current state.

Some other simulation softwares are now under development. Mcfall et al. [161] describe an integrated rule and object based implementation and extension of the ROSS language: ART-ROSS. It provides benefits of increased flexibility, simplified architecture, full explanation capability etc.

2.4.6.4 Generalised Manufacturing Systems Simulators

Simulators here refer to a step-by-step calculation of how a proposed system will perform [126]. Generalised manufacturing system simulators may be defined as one type of simulation particularly used by engineers in manufacturing systems [274].

As a special type of simulator, generalised manufacturing systems simulators have some unique features compared with general purpose simulation software.

--They provide specially written software to model particular problems in manufacturing systems [139].
--The modelling process is data driven, the user provides only numerical data as the system prompts [279]. However, in knowledge based manufacturing simulators, rules may be chosen.
--It is relatively easy to use in the sense that within its capabilities, no free-programming is needed, therefore, it is fast to build a simulation model, especially for non-professionals [60].
--They usually use engineering jargon with which the user is quite familiar.
MUSIK, developed in West Germany [275], is a modular simulation system for flexibly linked manufacturing systems. In [276], Warneck has introduced several examples by using this simulator.

SCHED/SIM, reported in [92], is designed to be highly interactive and easy to use for scheduling of the factory floor. One important feature SCHED/SIM provides is an interface to the MRP system to extract master schedule information as well as shop floor status data and another interface to collect information, regarding the current location and status of released orders from shop floor data collection systems.

Other simulators have been reported in [41] [189] [176] etc. APS [176] is a general purpose AGVs simulation package developed in SIMAN and FORTRAN. An on-line FMS simulator is described by Carrie [51], which provides more accurate information to the management due to the use of real time data for driving the simulator. Further discussions on generalised manufacturing system simulators can be found in [76] [92] [126] [198] [90] [108] [143].

Knowledge Based Simulation Environment. Unlike the conventionally generalised manufacturing system simulators discussed above, the AI based manufacturing systems simulation modelling are not fully developed, but the future of simulation tools for manufacturing systems will be bound up in the future of AI [241].

Wichmann [279] has categorised these systems into the following:
--Expert systems as a separate advisory system.
--Expert systems integrated and interfaced with conventional simulation language.
--Expert systems integrated and interfaced with an existing manufacturing system simulator.
--Knowledge based systems.

An expert advisory system serves as a decision support system which can provide some advice to the user about the use of a particular simulation language, and can contain knowledge about various expertise and about a problem domain. Gaines et al. [83] have introduced one example, FMS advisory system, combining expert system, database, and simulation techniques for planning flexible manufacturing systems.

When integrated and interfaced with a simulation language, the expert system could be a 'front-end' for constructing the model. This intelligent front-end is placed between the user and simulation packages for prompting the user for information and interpreting the results from the packages. Examples of such a system can be found in [280] [101] [182] [146].
MUSIK, developed in West Germany [275], is a modular simulation system for flexibly linked manufacturing systems. In [276], Warneck has introduced several examples by using this simulator.

SCHED/SIM, reported in [92], is designed to be highly interactive and easy to use for scheduling of the factory floor. One important feature SCHED/SIM provides is an interface to the MRP system to extract master schedule information as well as shop floor status data and another interface to collect information, regarding the current location and status of released orders from shop floor data collection systems.

Other simulators have been reported in [41] [189] [176] etc. APS [176] is a general purpose AGVs simulation package developed in SIMAN and FORTRAN. An on-line FMS simulator is described by Carrie [51], which provides more accurate information to the management due to the use of real time data for driving the simulator. Further discussions on generalised manufacturing system simulators can be found in [76] [92] [126] [198] [90] [108] [143].

Knowledge Based Simulation Environment. Unlike the conventionally generalised manufacturing system simulators discussed above, the AI based manufacturing systems simulation modelling are not fully developed, but the future of simulation tools for manufacturing systems will be bound up in the future of AI [241].

Wichmann [279] has categorised these systems into the following:
- Expert systems as a separate advisory system.
- Expert systems integrated and interfaced with conventional simulation language.
- Expert systems integrated and interfaced with an existing manufacturing system simulator.
- Knowledge based systems.

An expert advisory system serves as a decision support system which can provide some advice to the user about the use of a particular simulation language, and can contain knowledge about various expertise and about a problem domain. Gaines et al. [83] have introduced one example, FMS advisory system, combining expert system, database, and simulation techniques for planning flexible manufacturing systems.

When integrated and interfaced with a simulation language, the expert system could be a 'front-end' for constructing the model. This intelligent front-end is placed between the user and simulation packages for prompting the user for information and interpreting the results from the packages. Examples of such a system can be found in [280] [101] [182] [146].
Integrated with a manufacturing simulator, an expert system should incorporate structural knowledge about the simulator, such as data structure, format and model in order to allow the simulator to read, or write data and check model consistency. Furthermore, it should have strategic and heuristic knowledge about the type of manufacturing system and this should 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 [280].

An example of such a system is XMAS, described by Wichmann [279]. XMAS is a front-end/back-end for the FMS-simulator MAST [279]. The expert system contains the knowledge to analyse the output results, to control the production schedule, and to guarantee the consistency and to suggest conclusions, therefore, different scenarios can be created. Mellichamp et al. [165] have created an expert system for FMS design. The simulation model is written in SIMAN, while the expert system in KEE running on a SYMBOLIC 3670 LISP processing machine. The expert system uses inputs from the simulation analysis to analyse an FMS design. Design changes recommended by the expert system are incorporated into the simulation model and the process repeats until an acceptable design results.

Knowledge based manufacturing simulators in some way are more advanced than conventional manufacturing simulators. First, they have a structure similar in concept to the knowledge base [280]. Secondly, they have a separate data base with the description of entities in the model represented as objects and a separate control structure which is similar and works similar to the inference engine of an expert system. Therefore, it is easy to modify each part without affecting the others.

Knowledge based scheduling systems have been studied by many researchers [245]. However, a fundamental prerequisite to effective scheduling is an accurate model of the production environment. Detailed discussions can be found in [154] [67] [165] [101].

A knowledge based simulation system has been developed on an investigation into the routing of jobs in a multi-cell FMS [27] [28]. The difference is that the usefulness of this AI based simulation has been demonstrated particularly the real time on-line control and scheduling in automated systems in computer integrated manufacturing environments. Conclusions are drawn that, AI based simulation, unlike conventional simulations, emphasised on information flow rather than entity flow.

LUTKBS [274], developed at Loughborough University of Technology, is a knowledge-based simulation system written in LOOPS for design of flexible manufacturing systems. Major features 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.

2.5 System Analysis Methods

Till now, several modelling methods applied in manufacturing systems have been discussed. Another approach, system analysis method used in this project is introduced in the following section.

2.5.1 Functions of System Analysis Methods

A system may be defined as a group or set of objects united by some form of regular interaction or interdependence to perform a specified function [239]. System analysis is the examination, identification, and evaluation of the components and the interrelationships involved in systems. It is the logical design of the new system: the specification for input and output of the system and the decision logic and processing rules. It is usually a technique used in the initial phase of a system development.

Applied in simulation modelling, a system description entails two steps: a static description and a dynamic representation. The last section has discussed the dynamic aspect of a system by simulation. In this section the static description of the system is discussed.

The static representation of a system deals with determining or defining the existence of the subsystems, such as components of a system included in the model, structural relationships between them etc. It is a very important step in simulation building, especially when the system modelled is complex.

System approach attempts to study the total-system’s performance, rather than to concentrate on the parts [239] [7]. It stems from the recognition that even if each element or subsystem is optimised from a design or operational viewpoint, the total system performance may be suboptimal owing to interactions between the parts. It is particularly useful when problems are complex and affected by many factors, and it entails the creation of a problem model that corresponds as closely as possible in some sense to reality.

There are a number of techniques commonly used to produce system analysis and design. Typical examples among them are informal techniques; formal techniques; special input techniques and structured analysis methods. Among them structured analysis techniques are commonly used in manufacturing systems analysis.
2.5.2 Structured Analysis Methods

In these techniques, diagrams are employed to convey the information necessary for systems. However, they require a reasonable amount of skill, providing a variety of applications to aid structural thinking.

The basic philosophy of structural analysis and design techniques is that systems can be decomposed into small elements, allowing the whole problem to be seen at the same time as its more detailed constituent parts. This decomposition can be carried out until the limit of usefulness has been reached. The hierarchical nature and structure of the system permits the analysis of a system in terms not only of the system under consideration, but also in terms of both higher and lower level systems. The general methods of system analysis methods are shown in Figure 2.27 [7] [208] [78].

In this way, the complexity of the system is ordered so as to provide clarity. Not only are those techniques helpful in analysis and design, but they are also invaluable generally in the area of communication of information due to the consistency of documentation. One potential application of structured analysis techniques is in guiding the development of decision support systems such as simulation tools.

As shown in Figure 2.27, the structured analysis approaches usually include sequence and timing diagrams, action diagrams [70], process flow chart, input/output analysis, structured analysis and design techniques (SADT), IDEF techniques, controlled requirements expression (CORE), GRAI, data flow diagrams [208].

SADT was originally developed at SofTech INC. The concept of SADT is a top-down, modular and hierarchical one which uses the simplest diagraming techniques to represent 'things' and activities performed by people, machines and computers [221]. It views the system as a collection of diagrams, the first and top-most of which is a general abstract description of the entire system. An example for FMS design has been given in [18], in which Evers et al. have introduced SADT/SAINT which is a highly structured, top-down simulation methodology for defining, analysing, communicating and documenting large scale systems.

IDEF techniques have been discussed in section 2.4.5.3. As an example, Hick [102] maintains that IDEF techniques are suitable for modelling computer aided production management systems. Mackulak [157] reports that IDEF0 approach can be used as a potential industry standard for production control system design. In ESPRIT project No. 909 [141], Koriba et al. have used IDEF0 methodology to represent the ways in which the models are used in a 'top-down' manner, the types of information that pass through modules and which tools are appropriate to each module. The aim of this project is to devise a
methodology and supporting software 'toolkit' for helping small to medium-sized manufacturing enterprises in assessing the costs and benefits of computer integrated manufacturing. In a report by Pierreval et al. [192], the importance and role of analysis approach, when modeling with a language such as SLAM is envisaged for evaluation. Based on IDEF0, they have developed a graphic language called GALIS before building a SLAM network.

GALIS (Graphic Analysis Language for Industrial Systems modelling) was developed on the basis in part on an adaptation of IDEF0. Hence GALIS deals with understanding via model building, which requires a precise definition of problems and objectives.

Process flow charts, according to Shannon [239] are easy to use, especially excellent techniques for determining how the step-by-step details of a process are actually performed or anticipated to be performed.

Input/Output analysis method generates all possible ingredients required to give all possible results. This process is first defined, describable outputs identified and necessary inputs generated. Sub-systems are then carefully matched together by connecting inputs and outputs at interfaces [7].

Controlled requirements expression is a methodology based upon SADT, with a number of useful additions. These additions include the explicit inclusion of a section for problem definition and procedural stages for ensuring that models are logical, well documented and easy to follow [7].

The detail discussions on sequence and timing diagrams, action diagrams, data flow diagrams can be found in [70]. The discussion of GRAI can be referred in section 2.4.5.2.
Figure 2.1 Comparison of Two Organization Charts

Figure 2.2 Hierarchical Model of an Organization
<table>
<thead>
<tr>
<th>Layer</th>
<th>Function/Operation Aspects of the Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment</td>
<td>This level consists of sensing devices and control devices that communicate with or respond to commands from the station level.</td>
</tr>
<tr>
<td>Station</td>
<td>This level directs a single machine or a small group of shop-floor equipment that perform a specific task. The four functions identified for this level are equipment set-up, testing, takedown, and alarm monitoring. This level uses general purpose devices specific microcomputer controllers and programmable logic controllers (PLCs).</td>
</tr>
<tr>
<td>Cell</td>
<td>A cell control computer coordinates the functions of multiple machines in each manufacturing cell. It monitors the status of the machines, sequences work through machines, downloads numerical control instructions to machine controller and accumulates production statistics for use of the cell operator and shop level computers. PLCs, minicomputers, microcomputer vendors are all contending at this level.</td>
</tr>
<tr>
<td>Shop</td>
<td>This level schedules production, coordinates jobs between cells and allocates resources to jobs. It gathers information from the cell level for interactive use of shop managers and passes information up to the facility level. Computer devices found at this level are primarily minicomputers.</td>
</tr>
<tr>
<td>Facility</td>
<td>It represents the top level in each manufacturing location or plant. It is responsible for overall planning, execution and control of production within the facility. Computers are usually mainframe or large minicomputers.</td>
</tr>
<tr>
<td>Enterprise</td>
<td>This level is responsible for high-level business operation of the company, as well as many manufacturing operations. Computers at this level are usually frameworks.</td>
</tr>
</tbody>
</table>

Figure 2.3  
Main Functions of Each Layer

35
Figure 2.4

Hierarchy of an Industrial Enterprise

Functional and Computing

[225]

Computer

Control computer

Enterprise

Product area

Factory

Factory computer (ERP/PC/office)

Product computer (ERP/PC/office)

Process computer (ERP/PC/office)

Dedicated control computer (control)

Control

Group/equipment group

Equipment

Equipment components

Hierarchy level

Essential functions

Primary data management: employees, parts, products, materials
Strategic product and production planning
Payments adjustment, inventory, cost control
Purchasing

Wages and salaries
Controlling

Clearance: Strategic product and production plan

Primary data management: bills of materials, work schedules, equipment groups
Design
Marketing
Master planning
Materials management
Time management

Clearance: Production orders, drawings

Primary data management: employees, equipment, tools, devices
Order handling
Equipment design
NC programming
Inventory management
Quality control
Detailed scheduling

Clearance: Conveyancing and production process, inventory movements, NC programs

Order handling
DNC operation
Conveyance control
Inventory control
Cutting optimization
Data collection
Splitting/integration
Detailed scheduling (scheduling, sequencing, equipment assignment)
Data collection processing

Clearance: Conveyancing and production processes, inventory movements

Job shop management
NC and DNC operations
Sequencing
Data collection

Clearance: Control data

Local control

36
Figure 2.5

Relationship Between Planning and Structure in an Organization
Figure 2.6
The Basis of the ADEPT Model

Figure 2.7
The Role of the Simulation in the ADEPT Model
Major Functions in a Manufacturing Organization

Figure 2.8
Figure 2.9

Computer Integrated Manufacturing
Figure 2.10

Information Systems in Production
Figure 2.11

Classification of Models (1)

Figure 2.12

Classification of Models (2)
Levels of abstraction

Analysis
Design
Development
Implementation

Utilization
Conceptual Level
Logical Level
Physical Level

Development life cycle

Physical Model
Informational Model
Decisional Model

Figure 2.13 Classification of Models (3)

Figure 2.14 Classification of Models (4)
Where:

- ○ = Place (condition)
- − = Transition (event)
- * = Token (enabled condition)
- ~ = Arc (control flow)

(a) Concurrency  (b) Conflict

Figure 2.15  Petri Nets Examples

Figure 2.16  Tools Flow System
Figure 2.17

GRAI Example

Figure 2.18

Detail of Production Planning in GRAI
### Figure 2.19: Basic Structure of IDEF0

- **Control**
  - Input → Activity → Output
  - Mechanism

### Figure 2.20: Hierarchy Structure

- Parts which do not require Mch.1.
  - Mch.1
  - Mch.2
  - Queue1 → Activity → Queue2
  - Parts arriving from different group
  - Man 1
  - Man 2

### Figure 2.21: IDEF2 Flow Chart

- Parts Start
  - Queue1
  - Activity
  - Mch.1
  - Queue2
  - Mch.2

---

[18]
<table>
<thead>
<tr>
<th>Software</th>
<th>Host Language</th>
<th>Orientation</th>
<th>Output Collection</th>
<th>Graphic Output</th>
<th>Output Analysis</th>
<th>Simulation/Animation</th>
<th>Interactive Debugger</th>
<th>Data Management</th>
<th>Hardware</th>
<th>Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLAMII</td>
<td>FORTRAN</td>
<td>Process Event Continuous</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Window</td>
<td>No</td>
<td>No</td>
<td>IBM PC/XT</td>
<td>Pritsker and Associates Inc.</td>
</tr>
<tr>
<td>SIMAN</td>
<td>FORTRAN</td>
<td>Process Event Continuous</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Animation</td>
<td>Yes</td>
<td>No</td>
<td>Mainframe/IBM PC</td>
<td>Pritsker and Associates Inc.</td>
</tr>
<tr>
<td>TESS</td>
<td>FORTRAN</td>
<td>Process Event Continuous</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Animation</td>
<td>Yes</td>
<td>Yes</td>
<td>Refer to [19]</td>
<td>Pritsker and Associates Inc.</td>
</tr>
<tr>
<td>SIMSCRIPT II.5</td>
<td>FORTRAN</td>
<td>Process Event Continuous</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Animation (PC)</td>
<td>Yes</td>
<td>No</td>
<td>IBM, VAX, PRIME, NCR, HONEYWELL</td>
<td>CACI</td>
</tr>
<tr>
<td>SEE-WHY</td>
<td>FORTRAN</td>
<td>Event</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Animation</td>
<td>Yes</td>
<td>No</td>
<td>PC Mainframe</td>
<td>Istar Inc.</td>
</tr>
<tr>
<td>GSPP/H</td>
<td>FORTRAN</td>
<td>Process</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Animation</td>
<td>Yes</td>
<td>No</td>
<td>Mainframe/IBM PC-XT, AT Vax</td>
<td>Wolverine Software</td>
</tr>
<tr>
<td>SIMULA</td>
<td>Algol</td>
<td>Event</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Fair</td>
<td>Refer to [35]</td>
<td>Norwegian Computing Center</td>
</tr>
<tr>
<td>SMALLTALK</td>
<td>SIMULA</td>
<td>Event</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>IBM PC-AT Tektronix 4406 Workstation</td>
<td>Softparts Inc.</td>
</tr>
<tr>
<td>INSIGHT</td>
<td>SIMULA</td>
<td>Event</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Window</td>
<td>Yes</td>
<td>Fair</td>
<td>IBM PC DEC VAX etc.</td>
<td>Sys Tech, Inc.</td>
</tr>
<tr>
<td>CSIM</td>
<td>C</td>
<td>Process</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>VAX SUN</td>
<td>Micronormics &amp; Computer Technology Corporation</td>
</tr>
</tbody>
</table>

Figure 2.22

General Comparison Between Several Softwares
Knowledge Based Rules:

Knowledge Acquisition Subsystem

Explanation Subsystem

User Interface

Expert or Knowledge Engineer

Figure 2.23 Structure of Expert System

<table>
<thead>
<tr>
<th>Category</th>
<th>Problem Addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpretation</td>
<td>Inferring situation description from sensor data</td>
</tr>
<tr>
<td>Prediction</td>
<td>Inferring likely consequence of given situation</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Inferring system malfunction from observables</td>
</tr>
<tr>
<td>Design</td>
<td>Configuring objects under constraints</td>
</tr>
<tr>
<td>Planning</td>
<td>Designing actions</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Comparing observations to plan vulnerabilities</td>
</tr>
<tr>
<td>Debugging</td>
<td>Prescribing remedies for malfunctions</td>
</tr>
<tr>
<td>Repair</td>
<td>Executing a plan to administer a prescribed remedy</td>
</tr>
<tr>
<td>Instruction</td>
<td>Diagnosing, debugging, and repairing student behavior</td>
</tr>
<tr>
<td>Control</td>
<td>Interpreting, predicting, repairing, and monitoring system behaviours</td>
</tr>
</tbody>
</table>

Figure 2.24 Types of Expert System
Figure 2.25

AI-Based Simulation Softwares

LISP-Based

EZSIM  ROSS  KBS  OPS5  STEM  KEE

PROLOG-Based

T-PROLOG  T-CP  TS-  V-GOSS

PROLOG  PROLOG  PROLOG
<table>
<thead>
<tr>
<th>Software</th>
<th>Orientation</th>
<th>User-friendly Input</th>
<th>Graphic Output</th>
<th>Automatic Output Analysis</th>
<th>Behaviour Tracing</th>
<th>Simulation Animation</th>
<th>Interactive Capability</th>
<th>Hierarchical Modelling Capability</th>
<th>Hardware</th>
<th>Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROSS</td>
<td>Rules Objects</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Good</td>
<td>No</td>
<td></td>
<td>Rand Corporation AI Group</td>
</tr>
<tr>
<td>KBS</td>
<td>Rules Frames</td>
<td>Yes</td>
<td>Yes</td>
<td>Good</td>
<td>Yes</td>
<td>Window</td>
<td>Good</td>
<td>Yes</td>
<td></td>
<td>Carnegie Mellon University</td>
</tr>
<tr>
<td>EZSIM</td>
<td>Rules Objects</td>
<td>Good</td>
<td>Under Development</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>IBM PC</td>
<td>University of South California</td>
</tr>
<tr>
<td>STEM</td>
<td>Rules Objects</td>
<td>No</td>
<td>Animation</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Xerox Sun</td>
<td>Artificial Intelligence Ltd.</td>
</tr>
<tr>
<td>T-PROLOG</td>
<td>Clause</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
<td></td>
<td>Hungary Institute for Coordination</td>
</tr>
<tr>
<td>TS-PROLOG</td>
<td>Clause</td>
<td>Fair</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Window</td>
<td>Good</td>
<td>No</td>
<td>Siemens 77XX VAX</td>
<td>Institute of Computer Techniques</td>
</tr>
</tbody>
</table>

Figure 2.26

General Comparison of AI Based Simulation Softwares
3.1 Introduction

In this chapter, several approaches to classifying industrial organizations are introduced. The structural representation of and major functions in a general manufacturing organization are then presented and the relationship between them are stated. The performance evaluation of a manufacturing organization is finally discussed.

3.2 General Categorisation of Industrial Organizations

3.2.1 Introduction

Organizations can be viewed as production systems of two distinct varieties -- manufacturing production systems and service production systems. Manufacturing production systems which transform inputs into a tangible physical product are mainly concerned in this thesis as industrial organizations [266]. Conceptually it can be depicted as shown in Figure 3.1.

Industrial organizations can be classified depending on the different criteria, such as the process type in the transformation, volume of products, and the scheduling problems [23] [266] [282] [71] [33]. Figure 3.2 shows the general classifications of industrial organizations, which are discussed in the following sections respectively with the concern of reasons for each classification approach.

3.2.2 Process Based Classification

The organization can be classified into continuous, repetitive and special project type according to the production process. In a continuous process, the tasks are arranged according to the sequence of operations that are needed to make the product. All products follow a definite sequence from one task to the next. Every product follows the same path, uses the same inputs, and neither skips nor recycles [33]. Theoretically, a continuous process will run for 24 hours per day, seven days per week, and 52 weeks per year. In practice,
However, it is rarely achieved [282] [266]. Typical examples are steel making and petrochemicals. A repetitive process is one in which the product is processed in lots, each item of production passing through the same sequence of operations, e.g. the assembly of motor vehicles [282]. An intermittent process is used for low volume, batch, or customized products each of which requires different sets or sequences of tasks. One example is the special order fabrication shop [266]. A special project is often related to unit production. It involves a unique product requiring large amount of resources that are organized into a single process. Construction and ship building are examples of special projects [266].

This type of classification approach allows us to map the entire range of possible organizations graphically [282]. The degree of automation suitable for the various production environments increases in a sequence from special project, intermittent, repetitive to continuous production [231]. A continuous production environment needs a considerable design effort, as each section must be integrated with the others to achieve maximum efficiency. Conversely, the operation of a plant requires good judgement but little skill, often requiring little human intervention in the actual process [86]. Repetitive production, on the other hand, is typical of a decentralized manufacturing organization with high productivity goals. The weakness of repetitive production is the inherent inflexibility of the process.

### 3.2.3 Volume Based Classification

The second classification divides manufacturing into mass, batch, and job. Mass production involves the production of discrete items such as cars and domestic appliances. A single item or a very small range of similar items is manufactured in very large numbers. Batch production occurs in a period is small to enable mass production to be used. Wherever possible, similar items are manufactured together in batches. Job production, sometimes called 'one-off' or 'project' is the manufacturing of a single complete unit by an operator or group of operators [231].

In a job type manufacturing environment, the individual production run must be accurately analysed to obtain the most effective routing through all necessary operations [231]. Batch production methods, on the other hand, require that the work on any product is divided into parts or operations, and that each operation is completed throughout the whole batch before the next operation is undertaken. This technique keeps the capital investment low. As Lockyer [155] maintains it is in batch production that the production control department can produce most benefits. The greater volume of mass production usually results in a reduced unit direct labour cost since a greater total expenditure on
production aids and service functions will produce increases in productivity without an increase in unit indirect costs. But companies using mass production techniques should introduce greater product variety in order to survive in the market competition [86].

3.2.4 Scheduling Based Classification

This approach classifies the organization into make-to-order or make-to-stock according to the way the scheduling is made. In the situation of make-to-stock, the customer is identified after production has been completed. The actual manufacturing function is therefore, 'blind', in the sense that it does not perceive market demand directly, but only through the abstract medium of forecasts [231]. The actual orders remain in the sales department and are related to customer delivery by accounting documents.

In make-to-order manufacturing, the final destination of a product is known right from the beginning of the process. The types of products are changed according to the particular requirements from customers [155] [87] [231].

Usually a company is a kind of combined natural company. They must purchase the components and manufacture common parts of their products according to a forecast schedule, but assemble the complete units only upon receiving specific orders, build-to-stock, assemble-to-order [155] [87].

There is a simple numerical parameter which determines the appropriate scheduling mix in a manufacturing business. This parameter is the scheduling ratio (SR) which is shown in Figure 3.3.

In the problem of manufacture for stock, the tasks are known before the planning period is started and time is available to adjust the programme with some precision. The short delivery times made possible by the ready availability of finished goods allow these manufacturers to reach a wide range of buyers in a market where features and prices of competing products are similar. However it is not an easy task to prepare sales forecasts three to six months in advance. It is even more difficult to develop a new product. In the case of manufacture for order, the delivery times are longer. Sometimes, entire sub-assemblies have to be redesigned to meet particular requirements.

As analysed above, each type of production exhibits distinct characteristics and requires different conditions for its effective inception and working. However, in practice, it is rare to find in any organization that only one type of production is carried on [155]. In addition, there is a close relationship between several classes in each type of production.
Discrete-part production can be any type, such as job, batch, mass production, while process production is mostly continuous. Most of the intermittent and special project production is make-to-order and continuous is make-to-stock.

3.3 Structural Representation of Industrial Organizations

The section above discusses various classification methods to categorize industrial organizations depending on different criteria. In the real world, it is rare to find two organizations exactly alike. However, the general structure of all industrial organizations remains the same.

Such a structure can be presented in a hierarchy shown in Figure 2.2, but this model only presents the general functional vertical layers. In viewing an individual organization, the horizontal differentiation remains from layer to layer [170]. Within the formal structure of an organization work has to be divided among its members and different jobs related to each other. Work can be divided and activities linked together, in a variety of different ways [173]: by a major purpose or function, by product or service, by location, by nature of the work to be performed, by common time scale, by common processes, by the staff employed and by type of customer/people to be served.

The first presentation approach is the most commonly used basis for grouping activities according to specialisation, the use of the same sets of resources or the shared expertise of members of staff. Each individual organization can decide which activities are important enough to be organized into separate functions, departments or sections. Figure 3.4 shows an example which can broadly represent many companies. In this representative diagram, several functional areas are involved: personnel, engineering, operation, finance and marketing [170]. A further specific example, GEC ALSTHOM Large Machines Ltd. is explained in the case study in Appendix VI.

3.4 Basic Functions in Industrial Organizations

Although there exist various types of industrial organizations [170], a typical industrial organization must, at a minimum, consist of sales and marketing, engineering, production, personnel, and finance [156] [140] [86]. Production here is defined in the wide sense that it involves production control, purchasing, quality control, manufacturing and other maintenance activities. These functions are either directly involved with production, in which case they deal with materials, or they involve information about products, processes and the people who use that information [86]. These basic functions can be further refined to another level.
3.4.1 Sales and Marketing

Sales and marketing, in general, defines what is sellable and then sells it [140]. The sales and marketing function is concerned with the strategic and tactical activities which link the organization with its customers and suppliers. As concluded in section 2.2.3, one objective of an industrial organization is to satisfy a market need. The key to survival in market competition is customer satisfaction, which is determined by having appropriate products at the right time [55] [86]. Therefore, the sales and marketing research activity involves collection and analysis of information related to the organization's market and production. Changes in the market and the emergence of new markets brought about by economic and technological changes must also be identified.

3.4.2 Engineering

Engineering functions fall into three categories: product engineering, manufacturing engineering and plant engineering [266]. Product engineering is the design of products for manufacturing purpose. Manufacturing engineering is the design of manufacturing processes while plant engineering is responsible for continued production after the facility is in operation.

The design of a product usually begins with general or customer specification from market analysis which indicates customer needs or requirements. Detailed design then establishes the product function and form. Functional design is concerned with the performance of the product. Product form defines the geometric properties of the parts and the physical properties of the product. Raw materials are identified in terms of desired design and commercial properties. Therefore, the product design includes design and drafting activities. It will release component design, blueprint, standard, performance tests and specifications.

In manufacturing engineering, manufacturing processes are designed on the basis of product design outputs and information about the resources available to manufacture a product. The activities in manufacturing engineering usually include facility design, process planning, routing sheets, design of machines, jigs and tools, part programmes for numerical controlled machines etc [266].

Usually, product design involves both the development of new products, and the improvement of old products. Since products are the major output of organizations, and are usually its prime source of revenue, their design and selection should be identified closely with organizational objectives. On the other hand, product design must take account of the
concepts of quality, cost, and manufacturability, which means that simplification, diversification, standardization, modularity, and value analysis are important considerations in product design.

3.4.3 Production

Production means the process of producing economic goods, including tangible products and intangible services, from factors of production, thus creating utility by increasing value added [270]. The primary objective of manufacturing is to make products with a desired function, fast and at the lowest cost. The activity providing an adequate schedule for this efficient production is 'production management' [140].

Production management consists of the following five stages: aggregate production planning, production process planning, production scheduling, production implementation, and production control. The first three stages are usually included into a planning stage.

Aggregate production planning determines the kinds of product items and the quantities to be produced during the specified time period. Production process planning determines the production processes (process routes) by which raw materials are effectively transformed into finished products. Production scheduling determines an actual implementation plan defining the time schedule for every job contained in the process route adopted: that is, when, with what machine and who does what operation? Production implementation executes actual production operations according to the time schedule. Whenever the actual production progress and performance deviate from the production plan and schedules set at the planning stages, such deviations are measured and modifications are made. This is the function of production control [266] [270] [282].

3.4.4 Personnel

In spite of modern technology and all the systems and computers coming into widespread use, people remain the most important factor in modern manufacturing. Therefore, efficient personnel management will influence the performance of the system. Personnel as one of the element functions [173] provides a supportive advisory function common to the organization as a whole. It usually covers the following activities: employment and manpower planning, education, training and development, industrial relations, welfare, wages and health and safety [155] [5].
3.4.5 Finance

Financial department mainly manages the capital to make profits for the company [5]. Two activities are usually involved: financial accounting and cost accounting. Financial accounting deals with the business as a whole. The major issues in the financial management usually involve kind of capital, sources of capital, financial statement, budgets and financial control [5] [155] etc.

Cost accounting, on the other hand, deals with the determination and analysis of the costs of particular processes, jobs, or department within the company. Such cost accounting becomes the basis of cost control systems and budget control plans [213] [5].

3.4.6 Relationship between Basic Functions

Five main functions in an industrial organization have been briefly introduced in previous sections. The following section puts the emphasis on the discussion of relationship between three key areas concerned, sales and marketing, engineering and production [140].

Marketing cannot obtain orders for products which production cannot produce or design cannot engineer. Design cannot specify products that are beyond the scope of production. Production, on the other hand, must make the product within the specifications required by design, and it must also deliver finished goods in accordance with the desires of the customers as defined by marketing. Marketing should take into account the effects on production of all delivery promises made to customers.

3.5 Performance Judgement of Industrial Organizations

To judge industrial organization’s success or failure to meet their objectives, it is very important to evaluate the performance of individuals and the organization through performance criteria [7] [46].

To be more specific, the purpose of performance measurements are to allow comparison with the world’s best practice, to motivate their employees, and to record performance achievements that can be used to establish future performance goals [7].

Hence, it is necessary for every organization to identify a list of performance criteria. Globerson [88] has outlined some of them for industrial organizations: such as efficiency, percentage of defects, satisfaction, profitability, growth of profit, cost per item and response time etc. However, not all of them have well-defined measurement methods [104]. The selection guide-lines of performance criteria are discussed in greater detail in [88].
These performance criteria should then be indicated by some quantitative figures which are easy to obtain. The usually used quantitative indicators are as follows \[140\] \[33\] \[234\] \[208\]:
- throughput time of orders
- time in system for jobs
- percentage of defects
- work-in-progress
- waiting, storage and transport time for an order
- utilization of equipment or personnel
- payback period
- processing time per job
- waiting time per job
- number of jobs processed
- queue length.

Some of them are the high level system performance measure. The others are for detailed lower level, such as shop and machine level. One of them, may indicate several performance criteria. For example, throughput time of order represents satisfactions, if it is within the lead time of an order. It may indicate profit loss if it is beyond the due date of an order. Therefore, there is a close relationship between them. It is the target for management to balance them so as to get the main objective: productivity.
The Production Function

The SR Scheduling Ratio

Figure 3.2 Categorization of Industrial Organizations

Figure 3.3 The SR Scheduling Ratio
Figure 3.4  
Structure of a Manufacturing Organization
Chapter 4
THE PRODUCTION PLANNING AND CONTROL FUNCTION

4.1 Introduction

Last chapter has discussed several general aspects in an organization. This Chapter expands the key production facility of a company to discuss those issues which are concerned in the thesis to provide a foundation for the understanding of the simulation model conducted by this project.

4.2 Forecasting

The purpose of forecasting is to make use of best available present information to guide activities toward organizational goals [266].

There are many types of forecasting used in organizations. In this thesis, the focus is on demand forecasting. Demand forecasting here means the statistical treatment of past data to give an estimate of the future demand [200].

Demand forecasting is important because many companies have to purchase raw materials and other stocks in advance of actual orders, the annual budgets of revenue and expenditure are necessary for the financial needs of the business.

There are several forecasting bases available, the physical units will be used in this project. The forecasting models can be qualitative and quantitative. There are long range, medium range and short range forecasting according to the time horizon. In organization, these sales forecasts are used to establish production levels, facilities scheduling, set inventory levels, determine manpower loading, make purchasing decisions etc. Therefore, it is the main inputs to production planning and control function. More explanation is found in Chapter 10 [266].

4.3 Production Planning and Control

The goal of production planning is to effectively allocate system capacity (plant, equipment, and manpower) over a designed time horizon.
Production planning and control is the term which is now frequently used to mean the first three stages and the final stage discussed in section 3.4.3. Under this broad sense, three widely recognized operating strategies are often used: MRPII (Manufacturing Resources Planning), JIT (Just-In-Time), and OPT (Optimised Production Technology) [209] [53] [163]. OPT in nature belongs to one type of scheduling approach, JIT production is one of the topics in the field of production and MRPII covers both scheduling and control.

4.3.1 Manufacturing Resources Planning

Manufacturing Resources Planning (MRPII) is the extension of material requirements planning (MRP) in the sense that MRPII will support many other manufacturing functions such as purchasing, inventory and financial functions. An MRP system by itself is an open-end material planning system. While its material planning concepts are good, but it has no mechanism for matching the plan to resources or for comparing actual results to the plan [150].

The general structure of a typical MRPII system is shown in Figure 4.1 [46]. It works as follows. The requirements of products are calculated based on the forecast and customer orders. These requirements are then explored in the bill of material (BOM) files which break down a product into its constituent parts. Net requirements for these parts are then calculated by deducting available inventory from gross requirements. Finally a schedule is generated. Then the system issues work orders to the relevant work centres.

As a well-established philosophy, MRPII is the most widely used production management system. It is the only long term planning tool for manufacturers of complex goods. It is capable of recording the progress of manufacturing and the size of inventory at any given time. It has the ability to give accurate dates for manufacture at the point of order, control engineering changes and shop-floor work orders. In general, an MRPII system can provide management with an accurate record of all that is going on in the factory, enabling optimization of resources [150] [227] [49] [184] [209].

However, an MRPII system has fundamental drawbacks, which include the consequent need to set up procedures to attempt to achieve database accuracy, serious problems maintaining file integrity, lacking any strategy for quality control, line balancing, production smoothing etc.
4.3.2 Just-In-Time

Just-In-Time (JIT) may be viewed as a production philosophy which aims to improve overall productivity through the elimination of waste and which leads to improved quality [272]. Other potential benefits of JIT include less work in process, less raw materials, fewer finished goods in inventory, reduced space requirements and lower overheads [273].

JIT is not a single technique. There are a very large number of techniques and approaches associated with JIT, which are mainly divided into three areas: manufacturing techniques, production/material control, and organising for changes [273].

In manufacturing techniques, JIT focuses on flow through the operation and cellular manufacturing. The pull scheduling technique normally known as kanban is another key ordering and delivering system used in JIT. There are usually two forms of kanban, one-card or two-card. The card is used to signal the need to delivery more parts and an identical or similar card to signal the need to produce more parts [273].

Manufacturing planning and control systems require adaptation to a JIT environment. This means that MRP, OPT and JIT can mutually support each other. JIT purchasing requires that goods are supplied in small quantities, in exact amounts. To implement JIT effectively, some issues should be considered, such as quality and continual improvement [99] [272].

Therefore, JIT can provide manufacturing managers with a strategic framework with which to pursue excellence, low inventory, short lead times, high product quality, rationalise supplier relationships and ensure a smooth flow of goods on the factory floor [150] [284].

4.3.3 Optimized Production Technology

Optimized Production Technology (OPT), as the new production management strategy is the philosophy which improves productivity at any step that takes the company closer to its goal: making money [46]. This goal can be defined in terms of three criteria -- throughput, inventory and operating expenses.

The OPT philosophy differs from one traditional approach of maximising resources which is to balance capacity as near 100% as possible, irrespective of whether or not each resource is critical to the overall manufacturing process [163]. The OPT philosophy is based on some rules which, when followed, are claimed to help move the organization towards the goal of making money.
The rules are as following:
-- Balance flow not capacity.
-- The level of utilisation of a non-bottleneck is not determined by its own potential but by some other constraints in the system.
-- Utilisation and activation of a resource are not synonymous.
-- An hour lost at a bottleneck is an hour lost for the total system.
-- An hour saved at a non-bottleneck is just a mirage.
-- Bottlenecks govern both throughput and inventories.
-- The transfer batch may not and often should not be equal to the process batches.
-- The process batch should be variable not fixed.
-- Schedules should be established by looking at all of the constraints simultaneously. Lead times are the result of a schedule and cannot be predetermined.

The OPT system incorporates production and material requirements planning, it incorporates financial analysis and offers some strategies for marketing and for measuring plant productivity. It can quickly targets problems with quality, set-up times, and high inventories. It, however, needs the establishment of a detailed database which must be accurate and it also directly challenges traditional cost accounting.

4.3.4 Comparison between MRPII, JIT and OPT

MRPII, JIT, and OPT as production management strategies, each have their own advantages and disadvantages. There are also some similarities and differences between them.

MRPII is best suitable for companies which make discrete multi-component items, while JIT is suited to flow-line environment, which means that the higher the volume and the less the variety, the more useful JIT is. OPT, on the other hand, is suitable for job shop, repetitive manufacturing and the process industry, and companies with complex manufacturing operations [150].

OPT and MRPII are similar in some aspects. Both require product and machine information for the calculation of schedules, and both have computer software available. But OPT assumes that the lot size and lead times should not be fixed and are available according to finite capacity at any given time. This is very difficult to MRPII which assumes infinite capacity but tries to set up lead time buckets and lot sizes [150].

OPT is, on the other hand, more like JIT in the way it concentrates on issues such as quality, inventory reduction, set-up times, lot sizes and lead times.
4.4. Material Control

Materials are the largest single resource in any industrial organization. Therefore, material control is the key area for management. The management of materials concerns their flow to, within, and from the organization. The efficiency of the flow can substantially influence cost and revenue generation and thus hold serious implication for marketing, finance and production [267]. The major functions of material control include inventory control and material requirements planning.

The objective of inventory control is to have the appropriate amount of raw materials, suppliers, and finished goods in the right place, at the right time, and at low cost [266]. There are several inventory models, such as Economic Order Quantity (EOQ), Economic Order Interval (EOI) and Economic Production Quantity (EPQ). They are mainly concerned with independent demand like finished products and spare parts.

While material requirements planning systems were developed to cope better with the dependent items. Demand for dependent items is the results of the requirements generated for their use in the manufacture of another items such as raw materials, parts and subassemblies used in the manufacture of a finished product.

The demand for independent items should be forecast while the demand for dependent items should be calculated from the production requirements for independent demand items.

4.5 Conclusion

This chapter and last chapter have discussed the key issues in manufacturing organizations. Due to the objective of this thesis and time limitation, the research is restricted on three major production areas: sales and marketing, engineering and production. In production, the emphasis is placed on the production planning and control.
Demand management
Forecasting
Customer orders

Production Planning

Master
Production Scheduling

Material
Requirement Planning

Inventory control

Purchasing

BOM control

Production
Activity Control

Dispatching
Input/Output Control

Figure 4.1
Manufacturing Resources Planning
Chapter 5
METHODOLOGY FOR MANUFACTURING ORGANIZATIONS DESIGN

5.1 Introduction

This chapter establishes the need for the integrated methodology for manufacturing organizations design and analysis. It first states the specification, procedure and problems of manufacturing systems design. Based on this discussion, the need for the new integrated method is identified. Finally, the general difficulties remaining and the requirements for the integrated methodology are discussed.

5.2 Design of Manufacturing Organizations

The design of manufacturing organizations is becoming more complex due to the size of the system and the need for integrating various components and various functions. Considering the life cycle of a manufacturing organization, several phases can be identified: design, development, implementation and operation [66].

The design phase of a manufacturing organization usually includes three steps besides the plant/facility location decision and plant layout consideration. The first step is to set up the objectives of the system, which means to specify the manufacturing needs for future demand. The precise building of organizational production goals can prevent the system from being over or under designed. It also influences the configuration of a manufacturing organization. The second step is to select an appropriate candidate design among alternatives through system modelling. This step will investigate all possible scenarios and find out the proper configuration. The third step is to implement the design step by step [125]. The process is drawn in Figure 5.1.

The design phase is aimed at making out accurate specifications of manufacturing organizations. This point is very important in order to make sure that the system will operate properly and take into account every constraint, improve the choice for equipment, lay some reliable foundations for the development phase and provide thus a high quality and a cost within previously defined boundaries [45].
5.3 Need for Integrated Methodology

In recent years, manufacturing organizations have become increasingly complex in terms of number, diversity, integration across all levels of the organizational structure, and flexibility to cope with rapid changes in the market [217]. At the same time, these systems are more and more dependent on information technology as a key adjunct to the basic manufacturing technology itself. As a result, the task of designing and implementing integrated manufacturing organizations has become difficult [66]. The design study, therefore, requires high capital cost, a large commitment of skill, manpower and time to guarantee the efficient operation of manufacturing organizations [208].

It is possible to study a small domain with a specific methodology, but if we want to analyse a large system, we have to decompose it into sub-systems and apply several methods. Although extensive research work has been carried out on the related sub-systems in an organization and a cross section of modelling methods have been available, the potential benefits of a manufacturing organization cannot be fully realised as a result of the lack of an overall system analysis method and the difference between the design performance and actual performance.

With regard to the perspective of the systems development process, requirements specification is the first and important phase in a system life-cycle model. It should reflect an understanding of the system, guide the subsequent design and programming phases and serve as a basis for all communications concerning the software system being developed.

Furthermore the implementation phase in a manufacturing organization’s life cycle is very important and the high cost is required if it is to be put into practice, so it is important first to design the model from the static point of view [208]. Three main operands: information, materials and resources in a manufacturing organization [204] have to be considered.

As a result, an integrated methodology is needed which enables modelling large and complex manufacturing organizations and reduces the potential errors raising during the process of developing a software system. Thus the integrated methodology should contain both the structured analysis method to present activities in the system and the modelling method to simulate the behaviour of the system according to the time changing, including information flow and material flow. Figure 5.2 presents the general principles of integrated methodology and Figure 5.3 indicates the reasons for developing this integrated methodology.
5.4 Requirements of Integrated Methodology

The integrated methodology is in fact a grouping of several existing methods completed by specific elements which ensure integration. In developing this integrated methodology, certain requirements should be considered in both viewpoints of target system and modelling methods.

Present problems in manufacturing organizations. To design and analyse a complex and large manufacturing organization, the whole system should be taken into consideration to guarantee the integration of sub-systems and get the overall performance. The integrated method therefore should cover the design of three substructures of the system: physical, information and decisional structure. In parallel, the three main classes of objects in the system should be involved, materials, information and resources [204].

In practice, a manufacturing organization is usually organised by the function which each particular area performs. The hierarchical structure is the way that the organization is managed. The integrated methodology should present this hierarchy of the system clearly and the model should simulate exactly the activities the system acts and handle the cooperation between the sub-systems.

Future need in manufacturing organizations. To cope with a continuously changing environment, a manufacturing organization has to be able to manage in real time the required internal changes, i.e. to keep the internal adaptation cycle shorter than the external change cycle [8]. This requires the proper control of the whole manufacturing process from material source to product service in the market. In general, the future needs for the manufacturing industry are summarised as following:

--ability to manage in face of a continuously changing environment
--adaptability and flexibility of the total manufacturing system
--explicit processional, functional and dynamic-behavioural description of the total manufacturing organization
--availability of the right information at the right place at the right time
--ability to source equipment from different vendors.

As a result the integrated method has to be able to have the capability to keep the adaptability and flexibility of the system and to provide good communication in the system.

Generality and specialisation. The structure and management of manufacturing organizations change from one company to another. As a generic modelling tool, the integrated simulation model should contain major activities of manufacturing organizations which are performed in a general manufacturing system.
To deal with a particular case, the integrated method should provide the capability to specialize the personal requirements by a special user without many difficulties. On the other hand, different sub-systems perform differently, some are complex and important, some simple and not significant. For these important sub-systems, more effort is usually put for analysis. The method should have the flexibility to allow the detailed analysis as required.

The general requirements for the integrated methodology are presented in Figure 5.4. These requirements are drawn from the viewpoint of manufacturing organizations.

5.5 Selection of Structured Analysis Methods

There are several structured analysis methods available, such as IDEF0, SADT, CORE, data flow diagrams, input/output analysis method, process flow chart, and data flow diagram. Therefore, it is of significance to select the proper one to be used in this integrated methodology.

To indicate the system with realism, the top-down decomposition approach is preferred and diagrams should present the system clearly and they should be easy to understand. To provide a general design tool, the method should be widely recognized and used.

The integration with the simulation system requires that the structured analysis method should possess the capability to link easily with the modelling method used in the second part of the integrated methodology. For helping the development of the simulation system, this method should contain available and appropriate information. Figure 5.5 represents the requirements for this structured analysis method.

With the consideration of meeting requirements for the integrated method and the above discussion, the IDEF0 technique has been selected as the static representation tool.

The key elements of IDEF0 are its hierarchical structure and graphical representation. As a model evolves from general to the specific, activities and their information flows can be decomposed into more and more detail. Therefore, the top-down approach allows the complexities of manufacturing to be systematically organised into logically consistent and presented in a graphical context easily understood by managers and manufacturing personnel (refer to section 2.4.5.2 and 2.5.2).

5.6 Modelling Methods Selection

There are now many modelling methods available [286] [149]. Each of them holds unique features and is suitable for some specialised application domains. These methods
have been discussed in Chapter 2 in detail. Therefore, the right selection of the modelling method remains a key issue for manufacturing organizations design. The necessary consideration in selecting a suitable modelling method is discussed in the following section based on important criteria. The candidate modelling methods include static allocation method, mathematical programming method, queueing network, Petri Nets, heuristic algorithms and simulation method.

The system to be modelled. It is an important factor first to consider the scope of the target system to be modelled. The size and complexity of the system to be modelled can influence the choice of modelling methods because of the limitation and capability of the method.

Static allocation models are simple in the sense that they add up the total amount of work allocated to each resource, therefore, they cannot predict the whole performance of the system, especially the decision making.

Mathematical programming methods are typically used in the production planning or the design of optimized layout configuration of the system. They can only provide the optimized design of subsystems. On the other hand, it is not an easy job or may be impossible to abstract a mathematical model to design a manufacturing organization.

Queueing networks usually produce average values which assume a steady-state operation of the system. They tend to give reasonable estimates of performance based on the analysis of physical entities such as parts, but in considering the information flow in manufacturing system, the theory is not well established [115].

Petri Nets provide a graphical environment to permit a dynamic deterministic model of the system, but some questions remain when incorporating detailed system features such as many machines with finite buffer size and real-time routing policies [116].

Simulation, however, can mimic the detailed operation of the system through a computer program. In practice, it can model both information flow and part flow efficiently. It can analyse both static and dynamic behaviour of the whole system [262].

Objective of modelling. When considering the modelling methods used in the design and analysis of manufacturing systems, it is important, particularly from the practical user’s point of view, to differentiate between generative methods and evaluative methods [262].

Methods based on generative methods help find ‘good’ candidate decisions. The typical example is mathematical programming techniques. Essentially, these models can help resolve complex situations, but on the other hand, they may suffer from the "black
box syndrome, and may not work in some contexts. They also appear to remove the decision maker from the decision making process and do not allow the easy modification of decisions [234].

Evaluative methods, in contrast, help evaluate a given set of decisions. They include static allocation method, queueing network, Petri Nets, and simulation methods etc. Generally speaking, they are more of a tool to help the decision maker sharpen his intuition about the system since they can provide insight rather than decisions. They allow easy modification of decisions. However, it usually takes a long time to find good decisions using such methods.

System behaviour to be modelled. In designing and implementing manufacturing organizations, it is important to consider both static behaviour and dynamic behaviour of the system.

Physical models can describe the functional, procedural and if necessary, the finite state aspects of the system. One way of doing this is to follow a systematic approach which is based upon a modular decomposition of the planned system. This philosophy will be discussed in greater detail in Chapter 7.

Static and mathematical programming methods can only provide static performance through mathematical calculation for each resource. They ignore all dynamics, interactions and uncertainties. Queueing networks, Petri Nets, and simulation techniques, on the other hand, do account for dynamics [262].

Decomposition of the model. Manufacturing organizations are complex systems with regard to the fact that the system is large, and the behaviour of activities is complicated. Therefore, in considering design of such systems, it is necessary to divide the system into a number of connected subsystems, each of subsystem can, in turn, be subdivided.

The modelling method, however, should provide a powerful and easy environment for this hierarchical construction. Static allocation method, mathematical programming method, queueing network, heuristic algorithms, and Petri Nets methods are restricted to specific components, such as FMS, assembly line and restricted to certain topics like production control, layout optimization and scheduling. They are designed only for part solution of the whole system [116] [151] [166].

GRAI and IDEF0 techniques, on the other hand, can provide the hierarchical models for decision making, functional design of the system. However they can only provide a static walk-through environment and cannot give output for efficient analysis of the system.
Simulation methods, in contrast, can provide dynamic behaviour of the system. Simulation models can be built in hierarchy especially with the application of Artificial Intelligence techniques.

Other factors. Besides the above considerations for selecting the appropriate modelling method, there are some other factors which should be taken into account in the method selection, such as the ease of use, experience and skill required, available staff, the development tendency, computational requirement, typical input and output, and the time it takes to build the required model etc.

The static allocation and mathematical programming methods usually are easy to use, do not need much skill. The input to this model is quite simple, including production quantity and part routings. Output from this model contains the minimum number of resources needed, utilisation etc [116]. However, the basic theory used for the models should be well understood. While queueing networks and Petri Nets always take a long time to build, especially when the model is complex. With the graphic representation, the Petri Nets model is easy to understand and the theoretical development will make it more widely used in the future [268] [271] [220]. Input data for Petri nets models are usually part data, equipment data, process data and control constraints etc.

The conventional simulation model, in general, takes a long time to build and is not easy to understand [234], but it is now possibly the most widely used modelling method. The inputs for simulation models vary for the different models [19]. The research work and results have proved that it is a powerful and efficient tool for the design and analysis of manufacturing systems.

5.7 Simulation Methods Selection

Compared with the first part of the integrated methodology, the modelling method is more important and more effort and considerations have been put into this during the period of study. The requirements for the modelling method cover many factors. All of them should be taken into account while choosing the software package and building the model.

In order to analyse the dynamic performance of manufacturing organizations, the model should be able to handle the problem of the large size system. In addition, the model should present a manufacturing organization in a hierarchy whose structure parallels that of the static representation.
To cope with any change in manufacturing organizations, the environment has to have flexibility and adaptability. It can not only provide general modelling environment but also allows the user to create any specialised personality without difficulty. To help the user with the analysis capability, the model should provide a useful output dynamically and an explanation facility. The list in Figure 5.6 presents the overall requirements for the simulation method.

Because of so many available methods, it is necessary to consider some important factors when selecting suitable software from the user’s point of view. The choice between commercial software and home-made simulators depends on a lot of peculiar factors, such as time available, programming skills existing in the company etc [240].

The commercial software can be divided into two groups: general purpose simulation languages and application oriented tools. Simulation languages are general purpose instruments which contain subroutines to help model building and simulation reporting like GPSS, SIMSCRIPT [1][219]. They are easy to use, but still require the user to go conceptually from his manufacturing system of parts, machines, and material handling equipment into the abstract world of entities, queues, servers, and resources.

Application oriented tools, on the other hand, are software specially built to model a specific class of problem, like manufacturing systems. They are usually parametric general models for a class of application with a user-friendly interface to input and output simulation data. With the pre-defined model, there is some loss of generality [31][202][41].

Therefore, from the user’s point of view, the selection of general purpose simulation languages and application oriented tools largely depends on the size of the application field they are to model, flexibility of transportation capability on different computers, training and ease of use etc.

There is another selection between conventional simulation and knowledge-based simulation. This choice is more or less distinguished from the viewpoint of model builders.

The main factors which will influence the choice are as follows:

--size of the system modelled
--flexibility
--easy to use and understand
--modelling efficiency
--debugging
--model building and modification
--stochastic capacity
Current conventional simulation methods mainly use procedural programming techniques which deal with primarily numerical data. The solution steps are explicitly built into the subroutines [240]. Conventional simulations are difficulty to understand because of the incomprehensible code used [30]. They are also difficult to modify and debug since information and control are integrated together into the programmes and no debug facilities are available for efficient inspection. On the other hand, it cannot provide the credibility to the user because of lack of explanation facility, dynamic description. Flexibility is limited and it is very difficult or impossible to model a complex system.

Knowledge-based simulation, in contrast, has overcome the above drawbacks and is becoming widely recognized [67] [279]. The object-oriented programming method reduces the repetition of work (refer to Appendix III). The simulation models are easy to modify because of the separation of control and data. This can keep the knowledge up-to-date. Based on the knowledge base, the models allow the incorporation of different types of information such as rules, judgement, intuition and experience [148]. This allows more detailed and more flexible models to be created. Some knowledge-based simulation models also provide the interactive graphical model construction, graphical animation and graphical statistical output. The explanation facility can supply information which clarifies the structure and problem domain of a computer program for the user. The debugging ability makes modification and building of models easier, therefore, the different scenarios can be tested without difficulties.

5.8 STEM

STEM, which stands for Simulated Time Event Mechanism, is a software product developed by Artificial Intelligence Ltd, Watford, England. It runs in the ENVOS LISP and LOOPS environment on Xerox or SUN workstations [14].

STEM contains a library of standard NODES and a TOKEN to represent static and mobile entities, providing a general modelling environment. To simulate a specialised process, STEM possesses the capability to specialise any existing object class, which is the superclass of the specialisation. This specialisation allows the user to develop any desired application personality of the simulation environment.
STEM makes extensive use of the interactive graphics capabilities of LOOPS so that a model can be built and edited using a graphic editor on a screen window containing icons representing nodes [196]. During the execution of the simulation, animation and run-time monitors can provide demonstration and debugging features. STEM also provides the subsequent analysis of the logged data by graphic presentation and event chain.

The most unique and powerful feature provided by STEM is its ability to create and use "composites", in which a group of nodes are defined to be a single composite node. This defined composite node appears as a single icon in the original window, while a subsidiary window can be opened showing the composite structure. It can be treated as any other simple node.

5.9 Framework of the Project

Based on the above discussion, the project is established to create an integrated methodology for the design and analysis of manufacturing organizations.

Through the recognition of the requirements for the integrated methodology from both manufacturing organizations design view and modelling view, the structured analysis method and knowledge based simulation modelling method have been chosen to contribute to the integrated methodology.

The work then develops the first part of the integrated method, presenting functions of manufacturing organizations by IDEF0 technique.

Next it moves on to the development of the simulation model built on STEM. The emphasis of the work is put on the specialised manufacturing organizations model and the additional facility design to enhance the generalised environment of STEM.

Finally a case study of an industrial company is used to critically assess the value of the integrated methodology. Figure 5.7 shows the framework of the study.
Design Phase of Manufacturing Organizations

- Complexity of the target system
- Requirements for the accuracy, reliability, stability, and completeness of the target system
- Overall optimization of the global system
- The novelty of the field
- Different analysis methods of the sub-systems
- Life Cycle of a System Development Process
- Limitation of current design methods
- Limitations of modeling methods
  - Size
  - Incompleteness
  - Inflexibility
  - Lack of adaptability
- Difficult to model complex system
- Unable to communicate between different design methods
- Unable to represent and model the system correctly

Figure 5.2 Principle of Integrated Methodology

Figure 5.3 Need for Integrated Methodology
-- Inclusion of substructures
-- Inclusion of materials, information and resources
-- Hierarchical presentation and modelling
-- Capability to cooperate the integration
-- Adaptability and flexibility to handle the changes
-- Generality and specification capability
-- Detail analysis capability
-- General analysis capability

Figure 5.4 Requirements for Integrated Methodology

-- to present the static activity properly
-- to allow the hierarchical representation
-- to show the functions clearly, graphically and logically
-- to ensure completeness and consistency
-- to be easy to put into use
-- to possess capability to link with modelling method
-- to contain information for simulation model

Figure 5.5 Requirements for Structured Analysis Method

-- to present dynamic behaviour properly
-- to provide the capability to design the whole system
-- to represent the model in hierarchy paralleling with the structure of system analysis method
-- to have the flexibility
-- to be easily modified, edited and debugged
-- to provide useful output dynamically
-- to provide explanation facility
-- to link with structured analysis method
-- to link with other environment in the future

Figure 5.6 Requirements for Modelling Method
Figure 5.7

Framework of the Research Work

Manufacturing Challenge

Assessment of System Analysis Methods

Assessment of Modelling Methods

Assessment of Simulation Methods

Identification of New Integrated Methodology

Development of Integrated Methodology

Assessment of Integrated Methodology

Case Study
Chapter 6
SCOPE OF THE RESEARCH

6.1 Introduction

This chapter presents objectives of this research project, states the requirements and then discusses the approaches to achieving them.

6.2 Objective of the Research

The overall objective of this research project is to research into an integrated methodology for the design and analysis of complex and hierarchical industrial organizations. This integrated methodology combines the capability of the IDEF0 system analysis method and knowledge based simulation. This not only provides dynamic system performance analysis but also static system analysis. To achieve this, the integrated method must exhibit a great degree of interactiveness, comprehensibility, reliability, flexibility, modularity, efficient approximation and quick estimation capability.

In achieving this forwards the major objectives of the research work are as follows:
--to explore a novel integrated methodology for manufacturing organizations design and analysis;
--to research into the formal description method to analyse a hierarchical manufacturing organization in the static point of view;
--to research into the dynamic modelling method to analyse a manufacturing organization in the dynamic point of view;
--to demonstrate the efficient approximation capability of the integrated methodology.

6.3 Hierarchical Representation of Industrial Organizations

In a typical manufacturing organization development project, the designers’ aim is to establish two main things, the system’s physical requirements and the way the system is to operate. To achieve this, a thorough understanding of the processing requirements of those components to be produced by the system, knowledge of the environment of the appropriate technology, and development of a suitable information system for supporting operational decisions are required.
However, most current simulation packages are either too limited or general to fulfil the needs of all industrial users [45]. Another limitation is the general inability to relate all functional elements in the system. The building of the simulation model, on the other hand, will necessarily involve many interactions of objects, events, activities, information etc. Tackling such a problem in an unstructured manner is likely to increase complexity and amplify confusion [18].

The objective of this partition of the research is therefore to create a 'functional model' of a manufacturing system to provide a user interface for a thorough understanding of the system and the design aid for the creation of the simulation model. The IDEFO technique has been selected based on the criteria posed for the requirements of the integrated methodology (refer to section 5.6).

Furthermore, other possible methods such as data flow diagram, input/output analysis method, process flow charts and CORE are restricted in either the small system and process modelling or not fully investigated (refer to section 2.5).

6.4 Hierarchical Simulation Model

To design and analyse manufacturing organizations, it is important to understand the way the system is to operate. As a result, there has been a dramatic increase in the use of simulation in manufacturing systems during the past few years [67]. It has proved that computer simulation has been a powerful tool for evaluating and analysing manufacturing organizations [177].

The objective of this part of the research work is to build a hierarchical simulation model to provide dynamic behaviour of the system by using a commercially available knowledge based simulation package STEM. It provides an analysis and design aid for industrial organizations.

To meet this objective of the work, the following work should be taken to achieve this objective:
--to research into the original methodology used in the development of hierarchical modelling of manufacturing organizations,
--to research and explain the general functional and operational structure of the simulation model based on the IDEFO presentation and the knowledge based simulation environment,
--to enhance manufacturing functions capability,
--to identify the general data and knowledge required for simulation model building and the execution,
--to build a user friendly interface to enhance the capability of the simulation model, to provide quick estimation and alternative personalities, and to provide efficient and accurate output for the post analysis.

Unlike other simulation models, this model is built with some important and unique features.

1. First of all, the hierarchical model should cover major activities contained in a manufacturing organization. This implies that the simulation environment allows the involvement and modelling of sub-systems and their interactions, thus providing a realistic model.

   There are many functional areas found in a manufacturing organization to perform certain roles. Many single level modelling methods have covered some of them, such as production planning, scheduling, design of a flexible machining facility etc which have been discussed in Chapter 5 [250] [109] [23] [121]. According to the general procedure processed to a customer arriving at the company till the product is produced, it was decided to develop the modelling system which involves three major areas, sales and marketing, engineering and production, covering the activities related to the decision making aggregated to the generation of production order, production planning and manufacturing (refer to Figure 3.4).

2. The hierarchical model should provide the hierarchical configuration of manufacturing organizations. This is a key aspect of hierarchical modelling and is a major problem found in many simulation systems [175] [211]. This requirement indicates that the simulation model should be structured in the same way as the hierarchical structure of the target system and has to be shown on the computer, thus providing a physical and structural presentation of the system.

   With powerful graphic facilities provided by the LOOPS environment, this is achieved in the model by building an IDEF0-like structure which represents the hierarchical structure of a manufacturing organization. The configuration of the model is only a representative model which abstracts general functions found in many manufacturing organizations. For any particular user, this configuration can be easily modified with the LOOPS icon facilities (refer to Appendix V).

3. Since the model usually contains various types of knowledge and data, the organization of data base and knowledge base is another requirement for the hierarchical model. This means that the data and knowledge should be structured in a way that they can be easily modified and re-used.
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 manufacturing organization using objects [161] [164] [172]. With advanced knowledge representation facilities provided by LOOPS, 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 manufacturing and be separate from the application specific information [14] [172]. The control structure is manipulated by the inference engine embedded in STEM. The domain dependent knowledge is expressed in terms of rules and stored in knowledge base. The application specific information is stored in the database.

4. Another major problem of current simulation tools lies in the lack of capability to describe the system in a systematic way. This makes it difficult for the user to fully understand the modelling knowledge embedded in the model. This brings another requirement for the hierarchical model.

Based on the static description of the system, the best way to describe the activities involved in the model is level by level and box by box, leading to a clear and logic description of the system performance. It was decided by the author that to explain the decision made at each functional area, the following procedure is used:

\- to identify the level
\- to identify those areas contained at that level
\- to explain decision making at each area.

5. To provide flexibility, hierarchical modelling should allow the abstraction of different levels contained in the model. This illustrates that certain sub-systems can be analysed either including the detail elements of that sub-system or ignoring the detail. This enhances the application capability of the model.

This modelling system is intended to be capable of modelling a manufacturing organization over multiple levels of detail. It is made possible by applying the AI hierarchical abstraction concept. The levels of abstraction have to be consistent with the decisions made at the various stages during the design process. The model was defined to contain two levels of modelling detail. At the approximate level, it includes activities contained at company and shop level. The primary objective is to provide a quick estimation of the performance of the designed system. This estimation help identify the overall capacity of each shop, the work in process level and product performance. The detailed level involves two more lower levels, flexible machining cell and machine level. Thus the major objective of this level is to study the cell and machine performance. This helps determine the number of machine, the capacity and utilisation of work stations in cell and machines, AGV and transporter utilisation and provide real component performance.
6. Another major problem found in many conventional simulation tools is the level of user involvement. Most of the methods are data driven. This restricts the user to only designing alternative models by entering the data which define the physical structure of a manufacturing system. Therefore, it was intended that the AI based modelling system should be knowledge driven, i.e., both data and rules are used in the formulation of a model [274]. Thus in addition to the physical design, the user can also govern the behaviour of the system.

Since a lot of decisions have to be made in the system, an important issue in developing the model is therefore to identify which decision rules can be selected by the user, which are model defined rules. According to the effects the different rules can cause and the purpose of rules, two types of rules have been identified. A decision rule can be defined as a rule which is used to handle the conflict between certain actions of objects because of many possibilities, such as material movement rule and material supply rule. The rule which is used to control the model based on common principles is defined as the model contained rules which cannot be accessed by the user through the user interface, such as decisions made at forecasting and production planning.

7. To be efficiently used by industrial engineers, a user-friendly interface should be developed, thus providing an easy access for the selection of desired level of modelling detail, entry of input data and rules and collection of outputs.

Although STEM has the capability for physical model building, simulation debugging, graphic output presentation, they require more modelling knowledge, and have no manufacturing functionality at all. Based on LOOPS and STEM, a specialised user interface was intended to build in the model with the explanation facility.

8. As a design and analysis tool, the simulation model should be able to provide, with confidence, rapid feedback of system performance figures as desired by the user [274]. Thus certain measurements have to be identified, collected and provided automatically with regard to the major aspects of manufacturing organization performance.

Two types of outputs have been identified and can be provided in terms of the way they are collected, manufacturing oriented results in text form and standardised dynamic results by means of STEM in graphic form [14].

6.5 Approximate Modelling

It is generally impossible to simulate every aspect of a manufacturing organization. The computer requirements would be excessive and in most cases current computer hardware is not sufficient enough to allow complex modelling [143]. This means that the
organization model should be 'scaled down' to a manageable size. In addition, the operation of a manufacturing organization cannot simply be represented in a simple relation between functional areas.

Thus one important aspect in modelling is the level of detail [109]. The consideration is influenced by the purpose of the model, available environment and data. The integrated model is developed under some degree of approximation because of the complexity of manufacturing organizations and limitation of computing equipment. The objective of this part of the work will explore, in a systematic way, the challenge posed in the multi-level manufacturing organization modelling system.

The case study analysed in Chapter 13, 14 and Appendixes VI, VII and VIII will demonstrate the efficient approximation employed in the method which, in general, can be made without serious loss of model validity.
Chapter 7
HIERARCHICAL REPRESENTATION OF INDUSTRIAL ORGANIZATIONS

7.1 Introduction

Based on the discussion in Chapters 5 and 6, IDEF0 has been selected as the manufacturing organization representation method to help the user fully understand the structure of the system and provide an aid for building simulation models. Thus in this chapter, the issue of representing industrial organizations by using the IDEF0 technique is first discussed. The relationship between IDEF0 and STEM is then analysed. Finally the advantages of applying the IDEF0 method to present the structure of manufacturing organizations as the simulation model building aid are critically evaluated and assessed.

7.2 Research Objective

The IDEF0 technique has been recognized and reported by many researchers. Ranky in his book [208] has stated that in the modelling of FMS systems, static IDEF0 models can be used for system specification, definition and planning and thus a framework around which to build the dynamic models which are used to simulate the system’s operational performance by means of discrete event modelling, solid modelling and simulation etc. An example of this usage can be found in a report by Pierreval et al. [192]. Other reports can be found in [102] [59] [157] (refer to sections 2.4.5.3 and 2.5.2).

However, the IDEF0 model in these research projects is restricted to represent a small system [192] or only act as a static description tool [102] [157] [59]. No research project which integrates the static IDEF0 models and dynamic models for large scale manufacturing systems design and analysis has been reported. Thus the purpose of this chapter is to establish a representative IDEF0 model of manufacturing organizations and explore its role in the achievement of the whole research objective stated in Chapter 6.

7.3 Hierarchical Representation by IDEF0

Comparing the physical identity with the functional world shows a similarity between the target 'enterprise' and the main function "manufacture products" which represents the highest level description of the enterprise's function. Since there exists a physical
decomposition the function has to be decomposed as well [162] [285]. The decomposition of functions can be done in various ways and in those representations, the physical hierarchy cannot be easily identified [162]. For the detailed and logical functional decomposition, IDEF0 has been chosen to undertake this task [18] [75]. In IDEF0 representation, diagrams are composed simply of boxes and arrows. These boxes can be broken down into still more diagrams, until the system is described to any desired level of detail [59].

Each level in the reference model proposed by the ISO performs certain functions. The functional model therefore, presents the activities taken at each level [59]. However, it should be realised that no generic model exists which could represent the general structure of industrial organizations. A representative IDEF0 model is therefore created in this chapter to explain the methodology and to represent one example of IDEF0 representation of manufacturing organizations.

The representative IDEF0 model of a manufacturing organization is composed of sub-activities, each of which is identified as a unique activity related hierarchically to the top level, generic manufacturing activity [59]. To create a model composed of diagrams, it is required to indicate the position of each diagram in a model. It should be stated that there are alternative ways to organize these sub-activities. Following the IDEF0 recommendation that the number of boxes per diagram should be no fewer than 3 and no greater than 6 so that the diagrams remain comprehensible [208], the representative IDEF0 model is discussed in the following section.

The top functional level of a manufacturing organization is defined as "operate a company" whose main purpose is to produce products. This is presented in Figure 7.1, in which zero in the activity box means the top level. It is the case in any type of industrial organizations such as examples reported in [197] [140]. The external interfaces, inputs, outputs, controls and mechanisms create a boundary with the wider environment providing the context of the subsequent decomposition [248].

A single box at node A0 captures all the activities and information flow displayed in the multi-activity diagram at node A1 (Figure 7.2). Figure 7.2 contains detailed diagram part. Thus the diagram labelled with number A1 comes from a diagram labelled with number A0, which is the highest level in this system. If the diagram A1 has three boxes, they should be labelled A11, A12 and A13.

The following section discusses the major functions of each activity contained in each box. The detailed explanation of modelling knowledge can be found in Chapter 10, which explains how these functions are realised in the simulation model.
At factory level, three main functions are taken into account: sales and marketing, engineering and production. Three activities, therefore, within A1 "manage sales and marketing", "do engineering" and "do production" are selected as the essential sub-activities of "operate a company" and are decomposed in further detail.

The activity "manage sales and marketing" mainly dealing with the inventory management, decides if the required products from the customer can be released immediately or the production order should be released to the production area. It consists of three sub-activities, "manage inventory", "release products" and "generate production order", which is depicted in Figure 7.3. The marketing research is not considered at this stage [5] [278].

The activity "do engineering" generates a process plan for a newly designed product or simply makes some modifications to pre-designed products [140] [278]. In the model, it is supposed that all the customer orders are placed only for these pre-designed products, no modification is needed.

The activity "do production" mainly concerns high level decision making aggregated as production planning and low level physical manufacturing. From product forecast data, the production plan is generated and this plan is used to schedule the production [156]. Three sub-activities are involved, "forecast", "plan production" and "manufacture products" (Figure 7.4). The last two activities are, in turn, decomposed into the detail sub-activities.

The activity "plan production" generates a production plan according to the forecasting data and certain strategies used to calculate the demand for products [278]. There are several approaches available [170] [87]. This activity can be extended into a detailed level containing three sub-activities, "make master production planning", "make master production schedule" and "make material requirement planning" (Figure 7.5).

"Manufacture products" is the activity happening at low levels defined in the ISO model (refer to Figure 3.4), namely shop level, cell level and machine level. This activity involves four sub-activities "control production activity", "purchase materials", "make products", and "ship products". This is shown in Figure 7.6. The activity "control production activity" controls the production sequence for each part contained in the end product. Here parts represent components which are required by both sub-assemblies and assembly, and sub-assemblies which are needed by the assembly. "Purchase materials" involves purchasing for raw materials and components. The "make products" represents the general activity making the end products [278].
The activity "make products" involves three major sub-activities, "make parts", "assemble products", and "inspect products" (Figure 7.7). The "make parts" activity shows the general function producing all the required components and sub-assemblies. The only difference is that when making sub-assemblies, several components may be required. Therefore, the sub-assembly cannot be put into process until all the required components are available [156] [278]. The "assemble products" functionally is the same as the production of sub-assemblies except that several sub-assemblies may be needed for the final assembly.

The activity "make parts" can be sub-divided into four activities without considering the details happening at shop, cell, and machine levels. They are "put parts into input store", "machine parts", "put parts into output store" and "transport parts" (Figure 7.8). The activity "put parts into input store" simply puts all the arriving parts into the input store in each shop and release them into the working area after checking if all required parts are available. The "machine parts" activity represents the machining function in the working area. After machining, the completed part will be put into the output store or released out according to certain operating rules applied. The activity "transport parts" represents the movement of parts to other shops if required. It should be stated that the activities at cell and machine levels contained in the working area by "machine parts" are not extended into further level with the consideration that they cannot show significant difference with the above activities.

As a result, the main activities performed in an industrial organization, mainly at factory and shop levels have been presented by IDEF0 technique, which can be used as a helpful aid for simulation model building at the second stage of the project.

### 7.4 Relationship between IDEF0 and STEM

IDEF0 methodology has been used as part of the integrated methodology in the form of functional representation of the system which is combined with timing and precedence requirements to form a dynamic computer simulation model [75]. The similarity and difference between IDEF0 representation and STEM simulation environment can provide a complete understanding to the user what the system is, how well the model represents the system and what the simulation results mean [192].

**Simulation process.** As reported in [239] and [75], the first step in developing a simulation model is to describe the system. This activity uses a system description technique to construct a validated static system model [75]. Based on this static system model and the establishment of the simulation objectives, the next step is to construct the system simulation model. This activity develops a dynamic model of the system using a simulation technique.
and using the static model as a guide. Therefore, the IDEF0 representation of manufacturing organizations provides the initial model for the later development of the simulation model on STEM.

Hierarchical model. An IDEF0 model is structured so that it gradually exposes more and more detail with the level of detail being dictated by the analysis requirements [75]. This decomposition process continues until the system is described at the level of detail required [59]. The unique hierarchical building capability of STEM [14] allows the model to be built into different levels of composite nodes. Similar to the IDEF0 representation model, any node can be extended into further levels to show the detailed structure of the node. This similarity makes it possible to integrate these two techniques together to provide a full understanding of manufacturing organizations.

Graphical representation. IDEF0 representation describes the activities and their relationships pictorially (refer to section 7.4). The first diagram in a model is a single box that is a general description of the whole system (Figure 7.1) [102] [59]. The activity boxes and data arrows present the activity performed by the system and the relationship among the activities clearly. STEM provides sufficient flexibility in graphic representation for a simulation model. The graphical model shows the process points in the system (nodes), the relation between nodes are linked by paths (refer to Appendixes I and II).

Modelling viewpoint. IDEF0 is a multi-level static modelling method, which shows the inputs to the system, outputs from the system, and controls and mechanisms to govern the way the transformation is done and how the activity is accomplished [59]. Sequence and time is not explicit in an activity diagram. Feedback represents update information to a previous activity [75]. All these data are shown clearly on the diagrams. However, in STEM, the activity the node performs is controlled by the methods defined around the node class. They are written in a computer programming language which is not easily understood by the user like the IDEF0 representation diagram. The processing procedure is controlled by the time manager (refer to Appendix II).

7.5 Value of Hierarchical Representation Technique

IDEF0 is used prior to simulation model building in STEM not only because of the relationships existing between them (section 7.4) but also due to the fact that it provides some benefits.

1. IDEF0 is proved to be a powerful tool which can display the structure of activities performed by the system clearly and logically [59]. As a descriptive tool, it can identify the components of a system which cause change over a period of time [208]. Experience has
shown that the graphic capability of STEM and other simulation models cannot fully present the system structure, system boundaries, and various levels of functional specifications [75]. Thus the current simulation models make it difficult for the user to fully understand the system.

2. IDEF0 is a top-down design method widely recommended, especially for the design of complex manufacturing system strategies [59]. The analysis of "as is" decomposition and "to be" decomposition makes it possible to define requirement specifications [285]. Thus it is easy to obtain consistency between design and specification [192].

3. IDEF0 helps the designer build a complementary, consistent and correct model in the first place. As discussed in section 7.5, the system description is in fact the first step in developing a simulation model. In this context, decisions regarding the structure of the system, what activities to write, the functions of each activity and how they relate to each other can be made in a systematic way [18]. Thus the correct representation of the system plays a significant role in developing a simulation model because the dynamic model can be no better than the static functional model from which they are derived [75].

4. Decomposing a complex system into sub-activities allows attention to be concentrated on those activities while retaining the influence on the wider environment [192]. In addition, the consistency provides the possibility to allow teams to work on different partitions of a whole project at the same time in a systematic way.

5. The combination of the IDEF0 static modelling method and STEM's dynamic simulation method complements the weakness of each approach and enhances the capability and reliability of the simulation model. This provides a full understanding of manufacturing organizations in both static and dynamic point of view.

6. IDEF0 provides a good interface between the user and the dynamic simulation model, a clear 'map of the forest' before commitment is made and offers a direct route to dynamic modelling of manufacturing systems. The graphical representation ability provided by STEM makes it possible to automatically build an IDEF0 model on the computer which implements the STEM environment and it can be used directly as the simulation model or as the model building aid.

7.6 Discussion

This chapter describes a representative IDEF0 model to present the major functional activities performed in a manufacturing organization. Although the IDEF0 representation
built by the project does not include the detailed description of the system, it does abstract major functions, assists the designer to build the dynamic model at the second stage of the project and helps the user to understand the structure of the system modelled.

In comparison with other research projects investigating the role of IDEF0 representation, the strongest aspects of the IDEF0 model conducted by this research lies in the integration with the simulation model and the creation of an organizational wide model (refer to sections 7.3, 7.4 and 7.5).

It has to be stated that the IDEF0 model has limitation in presenting manufacturing organization with more functions. A more realistic IDEF0 model could be built if there were no time limitation and computing environment problems.
Figure 7.1

Business Environment
Group Constraints

Customer's Requirements
Product Records

Orders
Materials

Trading Information
Responses and Goods To Customers
Supply Requirements

Staff System Plant

Operate A Company

Top Level of IDEF0 Representation
Figure 7.2

First Level of IDEF0 Representation

NODE: A1  TITLE: Operate A Company
Orders

Manage Inventory

Production Status

Group Constraints

Customer's Requirements

Release Products

Generate Production Orders

Customer Order

Production Demand

Responses And Goods To Customers

Staff, System, Plant

NODE: A11
TITLE: Manage Sales And Marketing

Figure 7.3

Detail IDEF0 Representation of Node 1.1
Figure 7.4

Detail IDEF0 Representation of Node 1.3
NODE: A132
TITLE: Plan Production

Figure 7.5

Detail IDEF0 Representation of Node 1.3.2
 NODE: A133  TITLE: Manufacture Products

**Figure 7.6**  
Detail IDEF0 Representation of Node 1.3.3
Plan And Part Program

Materials, Components

Make Parts

Finished Parts

Semi-Finished Parts

Assemble Products

Products

Re-worked Products

Staff, System, Plant

Inspect Products

Finished Products

Production Schedule

Work Order

Customer's Requirements

Process Plan And Part Program

Materials, Components

Make Parts

Finished Parts

Semi-Finished Parts

Assemble Products

Products

Re-worked Products

Staff, System, Plant

Inspect Products

Finished Products

Production Status

TILE: Make Products

Detail IDEF0 Representation of Node 1.3.3.3
Chapter 8
HIERARCHICAL MODELLING METHODOLOGY

8.1 Introduction

This chapter establishes an original methodology for hierarchical modelling of manufacturing organizations.

8.2 Research Objective

The study in Chapter 5 has established the need for a new integrated approach to hierarchical modelling of manufacturing organizations. In the integrated methodology, two types of modelling methods have been identified, system description and dynamic modelling. Based on the discussion of requirements for the integrated methodology and system description (refer to Chapter 5), IDEF0 has been chosen to represent major activities of a manufacturing organization (refer to Chapter 7). The dynamic modelling of manufacturing organizations requires a novel simulation approach: hierarchical modelling to simulate the whole system.

Thus the purpose of this chapter is to research into the original methodology used in the building of the hierarchical model of manufacturing organizations. It identifies key issues posed in the development of the simulation model: what major problems the model can solve, how to collect information, why approximation is needed, how IDEF0 representation can be used, how to design the model and how to validate and analyse the model.

8.3 General Procedure for Building of Hierarchical Model

As analysed in Chapter 3, a manufacturing organization is structured hierarchically. Its structure indicates not only how the system is organised but also how it is operated. Therefore, hierarchical modelling of manufacturing organizations may be defined as an approach to simulating the target system following the system’s structural level, thus the model is developed in hierarchy (Figure 8.1). As a design and analysis tool, hierarchical modelling is different from single level modelling (refer to Chapter 6). Figure 8.2 lists general requirements for hierarchical modelling approach.
These requirements lead to the emergence of the novel hierarchical modelling approach. Six phases have been identified to be contained in this approach by the author. Figure 8.3 shows the life cycle of the hierarchical modelling [144] [234]. A number of steps have been presented, as well as a number of cycles through which they are related. It should be emphasized that these stages are not distinct and do not occur serially. Most of the time, efforts performed on each stage had to be conducted in parallel with other stages. Also the stages went through many iterations before a satisfactory simulation model resulted. The following sections will discuss each of these steps and illustrate how the hierarchical model conducted by this project is established.

8.4 Problem Identification

The first step is often referred to as problem identification. It highlights the scope of a model and illustrates what problems are going to be solved by the model. The setting of the objective of the model is an important step, otherwise the simulation model cannot provide the right insight into the system [144] [51].

Simulation of a manufacturing organization is a difficult task since the system involves many functional areas and covers a lot of issues. Due to the problem of time limitation and practical value, it was decided that the scope of this hierarchical model should focus on significant issues addressed at design and operation stages.

Thus the model is used to address following issues:

(1) The need for and quantity of equipment
   --resource requirements
   --capacity.

(2) Performance evaluation
   --throughput analysis
   --makespan analysis
   --bottleneck analysis.

(3) Evaluation of operational procedures
   --production scheduling
   --inventory level
   --work in progress
   --material control strategies
   --evaluation of production plan.
Compared with other simulation systems (refer to sections 2.4.6.2, 2.4.6.3 and 2.4.6.4), the model cannot answer questions like what is the basic configuration of the system. One reason is that it is very difficult to abstract generic modules to present functional areas at company level like engineering, production planning and production control since they are mostly associated with decision making not manufacturing resources (refer to Chapter 3). The other is that it is not practical with the available simulation environment. For other issues which are concerned in the model such as scheduling, the model cannot provide the detailed analysis and make adjustments based on the dynamic changes [234] [84].

8.5 Information Collection and Collation

The second phase is to obtain required data and information for model building. The collection of information has proved a key step in developing the simulation model since the quality of available data and knowledge may be a key factor in determining the level of detail and accuracy of the model. Also this process challenges the modeller to articulate, organise and quantify his/her knowledge. Building the knowledge based simulation model has forced the modeller to reconsider the rules he/she was using and look closely to the relationships between those rules. The result is that the modeller thinks much more clearly about the problem and the problem solving process.

Through the project, the following issues have been identified during the process of information collection and collation:

--what is the key information
--how to collect information
--how to analyse information.

There is usually a lot of information contained in a company, to identify that which is essential requires more consideration. This is again determined by the model’s purpose. To decide the need for and quantity of equipment, resources information is needed such as machine, work station, working area and transporter etc. To evaluate system performance, more information is needed about customer orders, products and components. To test operation procedure of the system, operating strategies, control logic, inventory management, production capacity and purchasing information should be available. In summary, the principal data is identified as: customer order, product, component, working area, work station, machine, transporter and operating strategies.

After having identified the key information, the next issue is to collect information. The key factors which should be considered is the scope of the model. The attitude used in
the interview and knowledge of manufacturing systems can also effect the information collection. The approach to collecting information is through the study of technical references and visiting industrial companies.

It should be noted that information collection may be involved at two different stages, general data collection before designing the model and the specific information collection for validating the model, i.e., industrial case study. At the first stage, the gathering of related information can be done through study of technical references. At the second stage, it is mainly through industrial experiences, i.e., visiting the industrial company. However, this does not mean that it is always happens serially. From the author’s experience, visiting industrial companies during designing the model is of value. The following discussion is mainly emphasised on the first stage, otherwise it will be specially stated.

To simulate manufacturing organizations, a major requirement is to understand how a company is operated in practice. The key issues are to identify major departments related to production in a company, to understand the role of each department and to figure out the interaction between them. This depends on the purpose of the model. Based on the issues addressed in the simulation model discussed in section 8.4, the key departments may include: sales, engineering, production planning, production control, inventory, purchasing, manufacture, test etc. Although different companies may use different terminology to name these departments, functionally they are the same.

The procedure and technique used in the case study are described below to explain how to collect information and interview experts from the company. The first step was to visit cells (shops in other companies) of GEC ALSTHOM Large Machines Ltd. to obtain general knowledge of manufacturing systems. This helps the author to understand the manufacturing capability and the major operations at each cell.

The second step was to interview domain experts from the key departments identified at company level. The interview was to follow a procedure: to describe the simulation system, to explain the objective of the case study, to state the role of the key department in the model, to explain what information is needed and to obtain the required information. It should be stated that the preparation for the interview, the attitude of enthusiasm and optimism in the project and a willingness to learn from domain experts can influence the result of information collection.

The sequence of interview is to follow the general production procedure from a customer order arrival at the company till the final product is completed. After that the interview for each domain expert has characterised the tasks in his/her domain. Major stages have been identified:

--sales
--design
--production planning and control
--manufacture.

The sales stage is initialised when a new order is received from a customer. According to the technical details supplied by the customer, the engineer from sales department studies the requirements of the product and then proceeds to draft a proposed design of the product and its process. The proposed design and its quoted cost is tendered to the customer for acceptance.

The design stage begins when a quotation of the proposed product is accepted. After a series of calculation, the product design is produced on a drawing and sent to production planning to check the available capacity.

Once it is proved that the production capacity is available, the production order is generated and released to manufacture for fabrication, machining etc.

The knowledge about the company has lead to the formulation of Appendix VI which describes the operation of the company.

After obtaining the required information, the next step is information analysis. The format of the information analysis is effected by several factors. One factor is that if there is a high degree of understanding of the expert's domain, the function of each department, a simple representation will suffice. Another is the complexity and representative power of the simulation environment.

The information analysis was done based on the information in its raw form obtained from the collection stage and produced a representation ready for implementation into the simulation system (refer to Appendix VII). The procedure used in the case study is found in Chapter 13.

8.6 Approximation

The third phase is to make assumptions on the model. Since it is impossible to involve every aspect of the system, approximation has to be made to decide how many levels and what activities should be involved. This simplification again depends on objectives and context of its application [144]. Great cares must be taken to preserve only those characteristics of the system that are essential. The potential for detail may be limited by several factors in the author's view: knowledge, data and computational environment. They can be summarised as follows:
1. The abstracted model of manufacturing organizations usually contains various types of knowledge. Some of them can be easily represented and understood, such as properties of system elements like the size and capacity of a machine. Others may be difficult to present and understand, such as the behaviour and interaction between sub-systems and decision making at management level. The conversion from a real system to a model usually results in the loss of some information. In addition, the knowledge cannot be clearly represented and well structured, especially the functions of the model cannot be easily structured.

2. Usually the model can provide adequate output data which describe the target system. However, for a full analysis of a complex manufacturing organization, even massive outputs cannot properly present the main features of the system performance. On the other hand, the model can only provide certain range of outputs.

3. There are alternative methods which can be used to investigate the characteristics of systems. In practice, analytic models require considerable effort to indicate explicitly whether an optimal system design exists when applicable. For simulation models, it is possible to identify the probable best alternative among those considered. Thus the user requires more effort to locate the right environment to investigate alternatives.

4. Modelling of the system is usually conducted using computer software. When a model is designed, the computing environment to some extent makes it impossible to present the realistic system because of its limited capacity.

Due to these problems, the model has to be approximate. Therefore how to approximate the model is another major problem which should be solved. In [234], Schriber has categorized three types of simulation systems, each with its own implications for the level of detail needed:

-- Determining the design basic for a new system
This category may involve the least detail. Furthermore in this category there may be a shortfall in knowledge and/or data.

-- Evaluating the operational aspects of an existing system
This category usually involves greater detail than does the preceding category.

-- Testing control strategies (such as production schedules and job release rules)
This category usually requires the greatest amount of detail if the model is to be valid for purposes of the decision at hand.

However, the modelling of a whole organization covers all these three categories. The above approach is therefore incapable of illustrating the problems proposed by the model.
In the author's view, the approximation of hierarchical modelling system needs determining general structure of the model, how many levels should be contained and what activities should be involved at each level in the model. The decision was mainly effected by the model's purpose, knowledge about manufacturing organizations, available computing environment, time limitation etc. The detailed discussion of model structure, model levels and major activities are found in section 8.7.

It has to be noted that the decision was gradually modified. Correspondingly, the evolution of the model started from simple to detail. At first stage, it was decided that only two levels defined in the ISO model (refer to section 2.2.6) were included in the model, i.e., company and shop levels. Once this was proved working, the model was then developed to cover two lower levels, cell and machine levels.

According to the procedure used in the design and analysis stage, the author has identified two different detail levels which are contained in the model, namely approximate level and the detailed analysis level. The approximate level covers only company and shop levels to get a simple model. The detailed analysis level includes two more lower levels: cell and machine levels. The discussion on each of those levels and different design stages are found in section 8.7.

8.7 Model Design

The fourth phase deals with model design. Based on the assumptions made at the third phase, this stage seeks a formal representation of symbol structure and their transformations into data structure and computational procedures in some programming language or package [51].

Major issues found at this stage are to decide overall structure of the model and to state why certain levels should be contained and at each level which activities should be involved. Some of them may be done at approximation stage.

As the project progressed, there were several steps of modelling when assumptions and data were gradually refined or different questions were being asked [51]. Briefly two steps have been identified and used through the project by the author: preliminary design stage and expanded design stage.

8.7.1 General Structure of the Model

The first issue found in the model design stage is to decide overall structure of the model. The decision was influenced by the following factors.
One is the identification of general entities which can broadly represent common features of elements contained in the system. In a manufacturing organization, the entities that flow through the system can be considered as three types. One is information, such as customer order or production order. The other is work, i.e., components, sub-assemblies, and end products (refer to Figures 7.1 and 7.2). The third is transporters and AGVs. These are mobile entities and their location is part of a description of the state of the system. The other functional areas, working areas, work stations, and machine stations are all stationary entities and are requested by production order and workpiece.

The other is to decide how the mobile entities and stationary entities are related. The operation of a manufacturing organization is the flowing of these mobile entities within or between these stationary entities. Associated with it is the decision making by people involved in the system. As a result it is natural to view the system elements as being connected by orders and workpieces while they are flowing through the system. Thus the state transformation of the system can be effectively described through the order and workpiece since they present the general input and output data of each activity contained in the system and link other elements of the system together (refer to Figures 7.1 to 7.8).

Finally how the system is organised may also influence determining the model structure. As discussed in Chapter 7, a manufacturing organization is usually structured hierarchically. Therefore, the model should reflect this structural aspect. Thus it was decided that the simulation model was organised in the way as described in IDEF0 representative model. Each activity in IDEF0 is defined as a functional area whose function is to perform that activity. The interaction between functional areas is triggered by the mobile entities, customer or production orders and workpieces which present input and output flow for each activity. The detailed discussion is found in Chapter 9.

8.7.2 Role of IDEF0 Representation

The discussion in Chapter 7 has illustrated the contribution of IDEF0 representation to the overall integrated methodology. This section identifies particularly its functions found in the development of the simulation model.

Firstly, the IDEF0 representation helps determine the overall scope of the model. Since it is the initial design stage of the simulation model, it identifies basic activities and their relationships and specifies the input, output and constraints of the system, thus leading to the scope establishment of the model. Although it may be modified according to the later progress and new requirements made at the model design stage, the initial representation helps configure the simulation model.
The IDEF0 representation shows the information flow in the model clearly and the interaction between different activities. Experience has proved that this is of value especially at the preliminary design stage. In order to mimic the real system, the methods defined in the simulation model should reflect these functions properly. Thus the identification of required information is the most important factor.

It is usually difficult for the user to figure out the structure and functions of the model simply from the computer code. A graphic configuration of the model can only show the structural aspect. The IDEF0 representation therefore can help the user fully understand the simulation model, thus overcoming a problem found in many simulation systems. It should be stated that although the IDEF0 representation is simple in terms of details, it not only provides efficient guidance between the modeller and system description, but also a friendly interface between the user and the dynamic simulation model.

8.7.3 Modelling Levels

The major factors to influence determining how many levels should be involved in the model is from practical point of view, which means that the simulation model is aimed at representing general manufacturing organizations. Thus the levels found in most cases should be involved.

Based on the ISO model (refer to Figure 2.2), four levels have been included in the model, company, shop, flexible machining cell and machine. One reason is that these four levels can be mostly found in industrial organizations. The other is the major issues established to be addressed by the model are involved in these four levels (refer to section 8.4). The other two levels are not covered in the simulation model. It is because the equipment level presents the basic hardware facility defined in a CIM system. Its function is similar to that of machine level in a conventional system. Furthermore it requires too much detailed information. In contrast, enterprise level presents the highest decision making level of the company and is usually beyond the scope of a manufacturing plant.

Company Level. This topmost level manages the whole organisation and controls overall production performance. The functions involved at company level usually include sales and marketing, engineering, production planning and control, finance, personnel and other maintenance service. The simulation model is intended to cover three major functions: sales and marketing, engineering and production since they can influence the system performance by dealing with the issues addressed in section 8.4.
Sales and marketing deals with customer orders, checks the inventory status of products required by customers and generates production orders according to operational strategies and stock.

Modelling of production involves getting the production plan, generating sub-orders based on the material requirements planning, bill of materials and manufacturing the components and products.

**Shop Level.** This level receives production order from company level and controls the manufacturing activity. The major functions include co-ordinating jobs between cells and machines, allocating resources to jobs, monitoring the operation of machines or cells.

To perform the required functions, all shops should have material storage capabilities. These stores are used to keep arriving and completed components for machining, queueing and transporting. These stores can be theoretically distinguished as input store and output store to deal with coming materials and finished components separately.

The processing of components is carried out by all machines and cells contained at each shop. Since this level is mainly responsible for the planning and control of real manufacturing, all these machines and cells can be simplified as working area. This working area performs the real machining operation of each shop.

**Flexible Machining Facility Level.** Flexible machining facility here means a simple flexible machining cell which consists of work stations, material handling systems and auxiliary stations. These facilities are usually arranged in shops if there are such facilities.

Work stations are sued for general machining function, while material handling systems are responsible for transferring parts within the cell, such as AGV, roller conveyer and robot.

This level is defined to be involved in the model mainly with the consideration that the model should vertically cover major levels proposed in the ISO model. This provides the possibility that the model can be extended into detail to concentrate on the cell performance evaluation and makes the model to be more generic.

**Machine Level.** Machine contained at this level usually means the single purpose machine. These machines are used to accomplish the machining operation which cannot be performed by machining cells or where these cells are not available. NC machines or CNC machines are also classified into this category. The only difference is that the operation of a NC or CNC machine is done by programming code not manually.
This level is used to present the work station level defined in the ISO model (Figure 2.2). Since in most companies these are the mostly used lowest level and key facilities, this level is defined in the model to provide the detailed analysis environment for different user and purpose.

8.7.4 Preliminary Design Stage

Like other simulation systems, the procedure of building the model is from simple to detail [51]. At this stage, the key issue identified by the author is to create a generic structure of the model which presents the correct relationship between major functional areas.

Since several levels are involved in the model and many activities are contained at each level, the interaction between these activities should be considered first. The useful aid at this stage is the IDEFO representation of manufacturing organizations. Thus based on the description conducted in Chapter 7 (refer to Figures 7.1 to 7.8), the IDEFO-like configuration of the model was created.

At this stage, only those important functional areas are considered, such as production planning, manufacture. The detailed information concerning sales, forecast, generation of production plan, operation times etc. are not available. Consequently, the simplified model was built containing only company level and shop level.

It was found that this procedure is different from other simulation systems. An example reported in [51] illustrated that at this stage, a simple model of FMS means not only fewer entities but also a simple relationship between these entities. As a hierarchical modelling system, this stage requires an accurate configuration or at least the higher levels should be to some extent realistic enough. Of course this does not imply no modification is needed at the expanded design stage. For example, the inspection shop was designed as a simple node with one destination at this stage. When the project progressed to the expanded stage, it was found that this could not represent the real situation, the inspection shop thus was extended as a composite node with two destinations available (refer to Figure 9.7).

8.7.5 Expanded Design Stage

With more knowledge gained on manufacturing systems and simulation environment, the project moved to the expanded design stage. The earlier model was revised for the new configuration with more manufacturing functionalities embedded [51].
The procedure is not to cover detailed operation at each functional area but to create an overall simple model first and then to put more information at each area gradually. More effort has be put on the production node since it is a key functional area to which the issues addressed in section 8.4 are within or related in a company.

The first step is to send the right components to the right place. Suppose that a production order has been generated and released to material requirement planning, based on the bill of material, sub-orders are generated to represent components and released to different shops. Process plan information used is provided by a company involved in the project reported in [197].

Once this was proved to be right, the next step was to generate a production plan according to the customer order and forecast data. This has been proved to be a difficult step since generating a master production schedule is a complex process and requires too much information. Therefore, certain assumptions had to be made. The production plan in the model was generated only based on the fixed figures and cannot deal with the dynamic changes. For detailed discussion, the reader is recommended to read Chapter 10.

The following step is to decide the production quantity according to the customer order and inventory status. It was assumed that the inventory management was also contained at sales department or at least inventory information was available.

Once the system becomes operational, attentions switched to questions such as scheduling rules, material movement rules, material supply rule etc.

After the approximate model proved working, two lower levels were developed. At machine level, the setting-up of machines and tool requirement are not considered. The design of a cell is simple compared with simulation systems which were designed to model FMSs reported in [175] [51] [274]. In the cell, only workstations, load/unload stations, AGV, pallets were included. Other elements such as tools, fixtures, tool transport were not considered. The detailed description is found in Chapter 10.

It has been realised that at every stage, the model should be verified to establish that the model correctly captures the logic and data which the modeller sets out to capture in the model and guarantee that the computing code is a correct implementation of the model.

8.8 Model Validation

After the whole model has been proved operational, the next step is to validate the model to ensure that the results are reliable. This was done by comparing the results from the model with those of the real system. A company, GEC ALSTHOM Large Machines Ltd. has provided realistic data and information to test the model.
It should be stated that the model was built to represent general manufacturing organizations not a particular one. Take GEC for example, flexible manufacturing systems are not implemented in the company, thus the cell performance cannot be tested. However, this would not influence the results of evaluating other elements of the system.

The result of model validation is mainly influenced by several factors, such as quality and quantity of available data, assumptions made in the model, computer running time etc. Generally speaking, the more accurate the data, the better result the model can provide. The less the available data, the less accurate the result. The longer the computer running time, the more realistic the result.

The case study was carried out according to the methodology established above. The procedure follows: visiting the company, interviewing domain experts from each department, collecting related data and information, converting the data into the computing format, running the model, analysing the results and making recommendations to the company.

The detailed discussion of case study objective for both the project and the company is found in Chapter 13. The results discussion of the case study is reported in Chapter 14. Finally the validation of the model is done in Chapter 15.

8.9 Output Analysis

The last stage is output analysis. This stage can also be considered as part of model validation. The investigation of which outputs are required by manufacturing engineers is an important issue in all modelling research [175] [274]. The accuracy of outputs to some extent determines the confidence level the model can provide to the user. Therefore how to analyse the output is another difficult the author met during the project.

The first consideration in output analysis is to identify these figures which are capable of indicating the system performance (refer to Chapter 12). With regard to the objective of the model, the following measurements are identified and used in the model:

--throughput
--makespan
--time in queue
--inventory level
--work in progress
--resource usage
--resource idle time.
The second is to design facilities which are capable of collecting those measurements. Two types of outputs have been identified in the model, manufacturing oriented outputs specialised by the model and standard outputs provided by STEM facilities. For the first type, the following categories have been identified: overall performance, product performance, component performance, inventory performance, working area performance, machine performance, work station performance and transporter performance.

Under the second type, the following parameters have been identified as important measurement figures:

--capacity of input store
--capacity of output store
--capacity of working area
--queue length
--time in queue.

The detailed discussion on available outputs is found in Chapter 12.

The final step is to interpret these outputs via running the model. This is done through a real case study (refer to Chapter 14).

8.10 Discussion

This chapter describes a new hierarchical modelling methodology based on the identification of problems posed in manufacturing organizations design and limitation of single level modelling methods. It discusses six major phases in the development of the hierarchical model, highlights the major steps involved in the project and at each stage what decisions have been made.

Based on the description of the original methodology established, the thesis then leads to the subsequent chapters to illustrate methods used in this research at key stages, identify decisions made and discuss major advantages contributed to the hierarchical modelling and limitations exited. The research includes the following key issues:

--configuration of hierarchical model
--identification and representation of modelling knowledge and data
--user interface
--interpretation of modelling results
--application capability.
Different Levels of Inputs

Hierarchical Modelling

Level 1

Level 2

Level n

Different Levels of Outputs

Figure 8.1
Hierarchical Modelling Architecture

--to model large and complex system
--to configure model in hierarchy
--to organise database and knowledge base efficiently
--to describe the model in a systematic way
--to model the system in different abstraction levels
--to allow different levels of inputs
--to provide different levels of outputs
--to contain user-friendly interface
--to provide statistic outputs
--to provide text outputs
--to provide dynamic outputs

Figure 8.2
General Requirements for Hierarchical Modelling
Problem Identification

Obtain Data and Information

Model Approximation

Preliminary Model Design

Expanded Model Design

Model Validation

Model Experimentation

Output Analysis

Figure 8.3

Hierarchical Modelling Methodology
Chapter 9
OVERALL STRUCTURE OF
THE SIMULATION SYSTEM

9.1 Introduction

Based on the IDEF0 representation model conducted at the first stage of the project (refer to Chapter 7), the second stage is to build an IDEF0-like simulation model to provide the dynamic performance of the system which will be reported in the following chapters.

The first target at the second stage is to build the physical structure and the operational structure of the simulation model. Thus the purpose of this chapter is to describe the functional and operational structure of the model and explain why the major components of the simulation model: the specialised user interface, specialised data base and specialised knowledge base are required, how they are organised, what difficulties the author has met and the major features of this simulation model.

9.2 Research Objective

The simulation structure environment is the facility for model structuring and execution. The most natural view is that of a collection of interacting objects moving through sequences of actions [144].

There are several ways to organise the simulation structure. One approach reported by Newman [175] is to create a data driven simulation model for FMS design. All the related information is stored in the data base, the cell configurator enables the user to efficiently design cells with the capability to automatically generate data for modelling. The structure of the designed system can be shown on the screen. In contrast to this approach, another method used by Wang [274] is to group objects according to the class they belong to. Compared with the first approach, the hierarchy structure of objects can clearly present the relation between objects used in the simulation model, the variables associated with each class are well organised in the class browser. The similar approach has been reported in [181] [161].

The first approach cannot clearly present the overall structure of the objects used in the simulation where as the second approach overcomes this problem and allows the user
to design his own system by selecting different resource objects and arranging them graphically on the computer. However, it only includes limited types of objects and is unable and inefficient to deal with a large scale system [243].

Thus this chapter researches into a new way -- IDEF0-like model to represent the physical structure of the target system and explains the functional and operational structure of the simulation system. For the modelling logic embedded in the simulation, the reader is recommended to refer to Appendix II.

9.3 Hierarchical Building of the Simulation Model

In knowledge based simulation systems, both the simulation and expert systems can be written in one single environment or two separate environments [145] [243]. STEM belongs to the first category of knowledge based simulation systems.

With the enhancement of the interactive graphics capabilities of LOOPS, a model can be built and edited using a graphic editor on a screen window containing icons presenting nodes. Therefore, the system structure, representing the system’s components and the relationship between them is described by various machines and resources in a manufacturing organization [243].

There are two ways to build the simulation model, a single level with all functional blocks organised in it and several levels with sub-functions arranged in each level hierarchically. Since a manufacturing organization is organised in a hierarchical nature, the simulation model has to be structured hierarchically in order to provide a realistic model. On the other hand, with fewer nodes contained at each level, it is easy to build and edit.

The discussion in Chapter 7 has explored the formal description of a manufacturing organization to represent the activities performed by the system. The structural building of the simulation model is therefore to convert this IDEF0 representation into a hierarchical presentation of the simulation model, with no particular restriction on the number of layers [32].

Supported by LOOPS, the objects become the principal focus of attention [243]. Detailed discussion of object-oriented programming can be found in Appendix III. Two types of standard objects in STEM are tokens and nodes (refer to Appendix I).

The token represents mobile entities passing through the system. It performs two functions, one is to present a physical presentation, like a production order or a part, the other is to trigger the action linked nodes such as the sending node and receiving node.
The node represents the static entities in the system, like machines in manufacturing systems. The generalised nodes, performing typical discrete event simulation functions, involved in STEM are originators, routers, stages and terminators. The basic functions and images of these nodes are listed in Figure 9.1. The image of these nodes can be modified.

Thus the construction of the simulation model is to organise these nodes graphically on the screen window from the description of the system [172] shown in Figure 7.1. This single box presents the general activity of the company, to make products. This can be built as a composite node in the simulation model which represents the resource: company. This is depicted in Figure 9.2.

In Figure 9.2, two originators are defined to generate orders from different markets. There are two ways to send the products to the customer, from the available stock directly or to manufacture them first and then to ship them out. The terminator means to receive the products by the customer. The detailed description of the manufacturing system in the simulation model is found in Chapter 10.

The diagram in Figure 7.2 presents the first level of the company. In STEM, this can be shown like Figure 9.3 which is the detailed structure of the composite node company of Figure 9.2. The image difference between the root map and the composite node is that the input and output path in the composite node is ended with a tag node which means the connection to a cNode map. In this company cNode, three nodes are defined. Sales and production nodes are cNodes, while the engineering node is a general router process node (refer to Figure 9.1 and Chapter 10).

The sales cNode is depicted in Figure 9.4 based on Figure 7.3 which shows the detailed activities in the sales department. PO-Originator in the diagram can generate a new production order token required upon receiving the arrival customer order token or simply pass the arrival token to the connected node. The path named 'exit' means to send the available products from the stock. This is one path connected to the terminator shown in Figure 9.2.

In the same way, the production node in Figure 7.3 is diagramed in Figure 9.5, in which the production planning and manufacturing nodes are zoomed in Figure 9.6 and Figure 9.7. In Figure 9.7, several shops are included. The finished products can be sent out to two destinations: stock and the terminator on the root map in Figure 9.2. The second destination presents another path to the production terminator on the root map. This structure is developed based on the function diagrams in Figure 7.6, in which the "purchase materials" box is represented as "outside" in Figure 9.7. Since IDEF0 representation only shows the functional activities of the system [157], one single box "make product" in Figure 7.6 is converted into the arrangement of all shops in the model in Figure 9.7.
Each shop in Figure 9.7 can be opened down to another detail level, including input store, output store, and working area shown in Figures 9.8, 9.9, 9.10 and 9.11. The cNode of shop 8 is different from the other shops, in which another input path may receive products from the inspection shop if necessary. In the inspection shop (Figure 9.12), the products to be inspected may be sent through three available ways: rework-station, shipping or back to shop 8.

To look at the detail activity at machine level, the shop node can be decomposed again like Figure 9.13. It is a general shop layout, the difference remains that the performance at detail level can be obtained as required. Thus the cNode "detailed layout" can be decomposed as Figure 9.14. This decomposition level can contain both single machine or a machining cell which in turn is composed of several working stations and an AGV (Figure 9.15).

Abstraction over structure can be used to view the system being simulated at different levels of detail as shown from Figure 9.2 to Figure 9.15 [174]. Representing the hierarchical structure of a manufacturing system in the simulation model can incorporate the hierarchy which is inherent in the system and represent the system at varying levels of abstraction.

9.4 Functional Structure of the Simulation Model

In knowledge based simulation systems, both the simulation and expert system can be written in one single environment or two separate environments [145][243][180]. STEM belongs to the first category of knowledge based simulation system. A typical knowledge based simulation is usually composed of data base, knowledge base, and inference engine [29][274].

The physical structure of the simulation model has been discussed above to illustrate how the model is organised, but how the model is operated and executed depends on other elements: specialised user interface, specialised data base, specialised knowledge base and inference engine (Figure 9.16). The last one is discussed in Appendix II, thus the rest is described in the following sections respectively.

9.4.1 Specialised User Interface

The simulation model user interface in STEM is highly graphical with menu-based and directly manipulable icons for user input, like other simulation models [32]. The user interface here means two different things, one is the simulation model building and simu-
lation process control provided by STEM and the other is the model defined interface for input information entry and output results presentation. The detail of these STEM defined menus are found in Appendix II.

These STEM defined menus can only help the user to build, edit and manipulate the simulation model in some aspects. For this specialised manufacturing organization model, a more enhanced user interface is needed to provide easy information entry and output collection. This is especially useful for a user who has manufacturing experience but limited knowledge about simulation. The detailed discussion on the user interface structure is discussed in Chapter 11.

9.4.2 Specialised Data Base

In an object-oriented programming style, an object is defined as a symbol associated with a unique database of properties and operations which represent the object [281]. Objects communicate with each other by the passing of messages, which define a protocol of all the requests an object will respond to. Other data structures, functions, procedures are private and can only be used from within an object [144].

A data base contains all the static data associated with a designed model. To build a specialised simulation model of manufacturing organizations, the objects which can best present the system need to be identified.

A manufacturing organization usually contains several kinds of objects, such as resource objects, entity objects, and consumable objects [178]. The resource objects involve machines, stores and transportation units. Their common characteristic is that together they sum up the physical limitations of the simulation system. The entity objects represent the flowing entities in the system, such as parts moving around and information passing within the system. The consumable objects could be considered as something which cannot be classified into resource objects like energy, machine lubricants etc [178] [139] [180] which are not considered in this simulation.

With these identified objects, the key issue is how to organise them into classes since the quantity of classes defined in the model can influence the model running efficiency in terms of computational time and model size. The fewer the number of classes contained in the model, the quicker the running of the model. Therefore, it is important to create a model with fewer classes involved but without loss of functionality and flexibility.
As a simulation model which contains sufficient detail of a manufacturing organization, this model involves four hierarchical levels, company, shop, flexible machining cell, and machine levels. The major elements contained at these levels can be defined as different types of nodes according to the function each of them performs. The flowing entities between these four levels are orders which present information of end products and sub-orders presenting component information.

There are several ways to define these classes. One approach is to define a general class to present common features of objects such as machines, the way used in [181]. This is efficient in presenting lower level activities like machining, transporting and queueing etc. For the higher level decision making activity, it is difficult to find two identical areas. Therefore, the following issues should be considered when defining these classes:

-- best present manufacturing functionality
-- easy to be re-used in other models
-- efficient model execution
-- available computing environment.

With these considerations, the following approach has been adapted: defining each higher level functional area into a single class and common lower level functional resources as a single class. Thus, two broad classes have been defined: node class and token class. The specialised sub-classes under each class are discussed in the following section.

Originator Class. Several originator classes are defined in the model, HMS-CORDER-ORIGINATOR, HMS-DMCORDER-ORIGINATOR, HMS-PO-ORIGINATOR, and HMS-MRPLANNING (Figure 9.17). The difference between the first two classes is that one generates the normal customer order into the simulation model. The other generates special customer orders in large quantities which may change the production plan. This class is defined to provide a wide option. HMS-PO-ORIGINATOR is defined to provide the potentiality to generate production orders. HMS-MRPLANNING generates sub-orders according to the customer order.

Terminator Class. Two terminator classes have been created in the model, HMS-STOCK and HMS-PRODUCT-TERMINATOR (Figure 9.18). One represents the products that have been shipped out. The other indicates that the finished products are stored within the company.

Router Class. There are several classes defined under this classes, including: simple router class, router process class, router probee class, and router prober class (Figure 9.19). Since many areas are involved in a company, it is not easy to simply create one single class like machine which can represent the common functions of various machines. These classes simply decide where to send tokens based on certain conditions.
Most process points in a manufacturing system belong to this type of class where processing time is needed before passing tokens out. One class is defined not following the regulation (with a dash between HMS and INVENTORY) since the ruleset can only be defined by a class with a name without a dash in it.

Two classes are built under the router-probee class, HMS-WSTATION and HMS-CELLSTATION. In a flexible machining cell, an AGV can only send palletised parts to work stations when those work stations are inactive. Only one class is defined under router-prober, HMS-AGV. This class only takes action when the related probee objects are idle.

One consideration should be made clear that AGV’s status also influences the parts sent from work stations, which means that workstations cannot send finished parts to AGV unless AGV is free. On the other hand, work stations should have the capability to check AGV’s status. Therefore, both work stations and AGV are specialised classes with functions of both router probee and router prober.

Stage Class. Several classes have been defined under this class, stage-process and simple stage class (Figure 9.20). Classes defined under stage-process are those resources where the action takes some time but has only one destination. Only one class is defined under simple stage class, HMS-EXIT. It is a common class to present a process point, simply to deliver tokens out to one destination. This class is used several times in the model. This node is defined for editing purposes.

Token Class. Two types of mobile entities move within a company, customer orders and components sub-orders (Figure 9.21). Customer order presents the information about end products and sub-order about components from which the final product is assembled.

From the discussion above, it can be realised that these specialised classes are defined directly from standard STEM instead of having to define general classes first, and then more specialised classes. It is considered in the following considerations:

1. Some aspects of a manufacturing organization cannot be easily categorised into one single class, like forecasting and master production scheduling etc. Those objects which can be easily classified into one class, like machines, input stores, output stores and working areas should be defined into one class.

2. STEM has already provided a general ground for specialising classes. For any particular application, the specialised classes can be directly defined from STEM standard classes.
3. The individual definition of specialised classes makes the modification and editing much easier. Therefore, any user may easily develop his own interest on the model and concentrate on any particular area.

4. The simulation model deals with each instance of the mobile entity, each instance has got its individual attributes. The inheritance between classes has to be defined clearly. Therefore, customer order and sub-order have been defined as two separate token classes.

The data base of each class includes the following information: class name, superclass, class variables, instance variables and method functions (refer to section 10.3.2 for an example).

In this model, the class variables are not employed. But more instance variables have been defined to constitute the complete definition. In instance variable definition, the first one presents the variable name, the second is the initialised value, which will change during the simulation running. The last part is the explanation information about this variable.

9.4.3 Specialised Knowledge Base

The knowledge base is the important element in the modelling system because it is a collection of simple facts and general rules representing manufacturing organizations. It is built around the specialised data base in a format of methods. Functional behaviour of each class of objects is defined by its methods. The specialised methods can be built through specialising defined STEM methods or creating new methods around that class.

Decision making knowledge in a manufacturing system can be categorised into two types: system behaviour or general decisions. The rules affecting the system behaviour involve those rules that may be decided by the user through the user interface. One example is the scheduling rules used at shop level or machining level. Different scenarios can be obtained through applying different rules [274].

The general decision rules define what functions should be taken under certain condition. These rules are not linked directly with the status changes of the manufacturing system. For example, in the assembly shop, the final product order can only be passed from the input store to working area when all required sub-assemblies have arrived. Otherwise the final order has to be queued in the input store, checking the required sub-order list till all required sub-orders are available.

These two types of rules are both defined as the classes’ methods. The detailed discussion on the knowledge base can be found in Chapter 10.
According to the way that knowledge is represented, two approaches are available to express the knowledge: rule sets or LISP procedures. Rule sets are based on LOOPS, procedures may be defined both in LOOPS or using LISP. The following is an example of the rule sets definition around HMSINVENTORY class to describe the action by the inventory to search the proper data information upon receiving a customer order.

```
WorkSpace Class : HMSINVENTORY ;
Compiler Options ;
Temporary Vars ;
Control Structure : DO1 ;
Args : order time item ;
***********

IF ~INVENTORY:inventorydatalist

THEN INVENTORY:inventorydatalist-(LIST (LIST order:ordertype
time item:productinventorydata));

THEN INVENTORY:inventorydatalist-(APPEND INVENTORY:inventorydatalist
(LIST (LIST order:ordertype time item:productinventorydata)));
```

In this rule set, one rule is defined. The rule defines that if the inventorydatalist variable of the object INVENTORY (instance of HMSINVENTORY class) has already contained some information in a list, the new information can be inserted into the list as an element of that list, otherwise a list which contains the product data information is created and can be used later on.

The method defined in this approach is clearly presented, but the execution needs more computing time and the editing procedure takes more time because of the compiling process each time.

Methods defined by using the procedure-oriented programming paradigm of LOOPS can work more efficiently although they are less comprehensible and take a long time to type. An example of procedure methods defined around HMSINVENTORY class is shown in the following section:

```
(Method
((HMSINVENTORY PickDestination) self token)
"Self has more than one output gate - to decide where to go according to"
```
certain rules."

\[\text{LET } ((\text{nextdepartments } (\text{GET-CONNECTED-NODES output}))
\]
\[\text{(found } \text{NIL})
\]
\[\text{(time } (\text{NOW})))
\]
\[\text{(if } (\text{NOT } (@ \text{ token woflg}))
\]
\[\text{then } (- \text{self GetInventoryData token})
\]
\[\text{;; to decide whether the re-order point is reached or not)
\]
\[\text{ (for a in (if } (@ \text{ ORDERORIGINATOR productorderdataflg})
\]
\[\text{ then } (@ \text{ ORDERORIGINATOR possibleorder})
\]
\[\text{ else (LIST 'OrderA 'OrderB 'OrderC 'OrderD'
\]
\[\text{ 'OrderE 'OrderF 'OrderG}')
\]
\[\text{ do (if } (\text{EQUAL } (@ \text{ token ordertype} a))
\]
\[\text{ then (if } (\text{GEQ } (@ \text{ (EVAL a) productinventorydata})
\]
\[\text{ (PLUS
\]
\[\text{ (@ (EVAL a) productsafetystock)
\]
\[\text{ (@ (EVAL a) pmeanleadtimedemand)))
\]
\[\text{ then (for section in nextdepartments}
\]
\[\text{ while (NOT found)
\]
\[\text{ do (if } (\text{EQUAL } (@ \text{ section name})
\]
\[\text{ 'Exit)}
\]
\[\text{ then } (\text{SETQ found section})))))
\]
\[\text{else } (- \text{self GetInventoryData token})
\]
\[\text{(if } (\text{NOT } (@ \text{ token ptflg}))
\]
\[\text{then (for section in nextdepartments}
\]
\[\text{while (NOT found)
\]
\[\text{do (if } (\text{EQUAL } (@ \text{ section name}) 'PO-Originator)
\]
\[\text{then } (\text{SETQ found section})))))
\]
\[\text{found)))
\]

The method presents the general knowledge about the selection of destination from inventory node. When there are enough products in stock, the products can be released to the customer directly and the product order goes through the node with the name 'Exit'. When no products in stock, the customer order will go to HMS-PO-ORIGINATOR node with the name 'PO-Originator'. Any other cases will release customer order both to 'Exit' and 'PO-Originator'.

127
9.5 Operational Structure of the Simulation Model

Like other simulation systems, four major areas are identified as being pertinent in applying the simulation model to solve manufacturing modelling problem: model configuration, data specification and operational rules formulation, model running and analysis of results [144] [274].

The discussion in section 9.3 has illustrated the physical structure of the simulation model. The fixed model configuration is identified by the author as being a representative system to explore the hierarchical modelling methodology. These specialised functional modes can be used in other simulation systems (refer to Appendix V). At this stage, some data has to be specified within the data base.

Once the model configuration is accepted, the next step is to specify the required data and describe the operating rules applied in manufacturing organizations. The decisions made include product shipping, part scheduling, material supply, material movement, inspection etc. The facilities embedded in the knowledge based modelling system allows the user to review the selected rules and to change to another rule.

After the functional model is established, output facilities are required to help the user understand the behaviour of the model and explanation of the results. This is done by developing a textual output facility (section 9.4) and graphical output facilities provided by LOOPS graphic techniques. To help understand the happening sequence of events, STEM provides the mechanism to trace them automatically.

Figure 9.22 shows the operational structure of the modelling system, the user is first required to choose the appropriate level of modelling. Once the level is selected, the user is then asked to enter the data through the user interface. After this, operational rules have to be entered to define the behaviour of the system.

The model is now ready to be run. As stated before, the operation of the model is controlled by the embedded inference engine. During the running of the model, animation shows the mobile entity flowing within the system. Furthermore, monitors can also be attached to specific objects to dynamically display the value of particular variables of an object [274].

When the running of the model is completed, the user can select output results. If the results from the model are not satisfactory, the user may want to initiate further runs. This can be done simply following the steps described above.
9.6 Discussion

This chapter describes the physical structure of the simulation model -- an IDEFO-like structure, functional and operational structure of the model as data and knowledge management system.

The hierarchical IDEFO-like structure of the system displays the relationship between major activities found in manufacturing organizations. It includes only those activities which will determine the performance and identify the influence of the various factors. Furthermore, the model represents the general structure of manufacturing organizations. However, the pre-defined structure implies that if the user intends to include more functional areas, he/she has to do some modifications on the overall structure. It has been proved that it can be easily done. The procedures for the modification can be found in Appendix V.

The IDEFO-like physical structure of the simulation model is different from the IDEFO representative model in some aspects. One is the relationship between different boxes presented by one path. The other is that the control and mechanism information are not physically shown in the IDEFO-like simulation model, the operating strategies and constraints are defined in the methods of the each class associated to each box.

Each node defined in the model can be used again in other models. This means that they can be selected just like any other standard STEM defined classes and re-arranged by the user to establish a new simulation model in the same way as reported in [181] [274]. In this case, the IDEFO representation needs to be modified as well. This capability is especially important in providing a wide application area as it potentially allows the user to define his own simulation model with particular interest.
<table>
<thead>
<tr>
<th>NODE</th>
<th>FUNCTION</th>
<th>NAME</th>
<th>IMAGE</th>
<th>SPECIFIC FUNCTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGINATOR</td>
<td>To generate tokens and pass them into the simulation</td>
<td>Originator</td>
<td>🔄</td>
<td>One path attached to the output gate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multi-Originator</td>
<td>🔄</td>
<td>Up to 100 paths attached to the output gate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Originator-Prober</td>
<td>🔄</td>
<td>To probe the activity of the destination node and to send token to the connected node only when it is inactive.</td>
</tr>
<tr>
<td>ROUTER</td>
<td>To control the destination of the tokens</td>
<td>Router</td>
<td>🔄</td>
<td>No time needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router-Process</td>
<td>🔄</td>
<td>To take activity time before passing token to the next node. Arriving token is enqueued while the process is active.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router-Prober</td>
<td>🔄</td>
<td>To probe the activity of the connected node and to send token only when the destination node is inactive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Router-Probee</td>
<td>🔄</td>
<td>To send information to the connected probe node and to accept token while it is free.</td>
</tr>
<tr>
<td>STAGE</td>
<td>To a single connected destination</td>
<td>Stage</td>
<td>🔄</td>
<td>No time needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stage-Process</td>
<td>🔄</td>
<td>To take activity time before passing it to the connected single node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stage-Prober</td>
<td>🔄</td>
<td>To probe the activity of the only one connected node and to send token only when it is inactive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stage-Probee</td>
<td>🔄</td>
<td>To send information to the connected probe node and to accept token while it is free.</td>
</tr>
<tr>
<td>TERMINATOR</td>
<td>To destroy token</td>
<td>Terminator</td>
<td>📞</td>
<td>No time needed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Terminator-Process</td>
<td>📞</td>
<td>To take time to dispose of the received token.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Terminator-Probee</td>
<td>📞</td>
<td>To send information to a probing node and to receive token only when it is free.</td>
</tr>
</tbody>
</table>

Figure 9.1

STEM Node Class
Figure 9.7
CNode Map of Manufacturing
Figure 9.8
CNode Maps of Shop 2, 3 and 4
Figure 9.9

CNode Maps of Shop 5 and 6
Figure 9.10

CNode Maps of Shop 7 and Assembly
Figure 9.11: CNode Map of Shop 8

Figure 9.12: CNode Map of Inspection Shop
Figure 9.13

CNode Map of Shop 1
Figure 9.14

CNode Map of Detailed Shop 1
Figure 9.15  CNode Map of Cell
Figure 9.16
Operating Procedure of STEM

Knowledge Base
Simulation Model Representation
Simulation Model Building Support
Simulation Controller
Data Base
Time Manager
Figure 9.17  
Originator Class in Simulation Model

Figure 9.18  
Terminator Class in Simulation Model
Router Classes in Simulation Model
Figure 9.20  Stage Class in Simulation Model

Figure 9.21  Token Class in Simulation Model
Operational Structure of the Hierarchical Modelling System

Figure 9.22
Chapter 10
GENERAL MODELLING KNOWLEDGE

10.1 Introduction

This chapter discusses the general modelling knowledge embedded in the simulation model. The discussion follows four levels defined in the simulation system, company, shop, flexible machining cell, and machine with consideration of the functions not the physical structure.

10.2 Research Objective

The functionality of a simulation model is determined by the knowledge contained in the model [243] [174]. The major problem of many simulation tools lies in the lack of capability to describe manufacturing functions in a systematic way. Furthermore, a general simulation package like STEM contains no manufacturing functionality at all.

In addition, manufacturing organizations are complex in terms of size, structure and operation. Thus it is impossible to involve every aspect in the simulation model. To be applied at different design stages, the model should provide flexibility for different users. On the other hand, the right level of involvement with the model is another key issue which should be considered. Thus the objective of this chapter is to identify the general data and knowledge required in modelling a manufacturing organization and research into the approach used in presenting these manufacturing functionalities. It also highlights the need for different purposes, thus leading to the development of different levels contained in the model. Based on the recognition of general data and knowledge required, this chapter identifies those major ones from which the user can select and change, allowing the user to govern the performance of the simulation model.

10.3 Initial Comments

The six layer ISO model was introduced in the consideration of both functions and computer networks [162] [233] in order to support a general standard hardware system (refer to Figure 2.2). The allocation of basic tasks at each level of the hierarchy has been discussed in section 2.4. With respect to the structure of the simulation model, four levels have been taken into account. However, they are not directly defined in the model with the
clear sequence of the hierarchy in terms of these four levels. The distinction between levels is put on nodes rather than composite nodes. The discussion is therefore carried out at each level but with the concentration on individual node contained in each cNode.

At each level, the discussion is divided into initial comments and state description. The first one briefly describes the associated level and indicates the assumptions made in the model. The second illustrates the manufacturing activities involved and states how they are realised in the model with the emphasis on possible alternatives and decisions made by the author.

10.4 Top Level of the Model

10.4.1 Initial Comments

The top level of the model represents the major activity of a manufacturing organization, "operate a company". As described in Figure 7.1, the general inputs to a company are the customer order and brought in materials. The outputs are the finished products and related information. To present the customer arrival activity and product shipping activity, two types of functional nodes have to be specialised at this level (Figure 9.2). One is the originator generating customer orders for the company. The other is the terminator to dispose of the arriving products to handle the outputs from the company.

10.4.2 State Description

The general procedure that a company uses to finally get the contract from customers usually includes the enquiry from customers, tender from the company, negotiation and final decision. The customer order therefore represents the final fixed contract.

A customer order usually contains related information about customer requirements and products, such as customer name, unique product number, description, quantity, shipping time, time now and price. In some particular situation, the detailed financial calculation on the product is also found in the customer order (Appendix VII). A general customer order format is summarised in Figure 10.1.

Customer Order Originator. To generate customer orders into a company, some aspects should be considered. One is the arrival pattern. Theoretically, the arrival of orders can be modelled in one of the following ways which are used to model the jobs arriving at
a job shop [132]:
-- instantaneous release of the order into the company
-- periodic release of all available orders at the beginning of the simulation period.

However, a single pattern cannot represent the general arrival performance in the real situation. It is usually stochastic, depending on the product type, market situation etc. The arrival pattern in fact presents the specified input level of the system. Thus the selection of arrival pattern should best reflect the real situation. In most cases, the first approach is used in the investigations. In this approach, the most popular arrival pattern is the use of Poisson process [132]. Poisson distributions are typically used to represent the number of events that occur in an interval of time when events occur independently to each other [14]. The function generates a pseudo-random variable from a Poisson distribution with mean of poissonMean. The function parameter is poissonMean, which is both the mean and the variance of the Poisson distribution.

Another aspect is the order type which presents product type. Customers only ask for the available products made by the company, with perhaps slight modifications. Therefore, the customer orders generated should involve all important product types.

When customer orders are generated, each order is defined with the information about product type, customer name, order quantity, product due date and new product flag. The last variable: new product flag will be used in sales to decide whether the product needs some modification on the design in engineering department or can be released to the production area directly. This means that the company should provide customer information based on previous orders or forecast information. The default customer information can be found in the specialised class definition HMS-CORDER-ORIGINATOR:

```lisp
((MetaClass Class Edited% ; Edited 6-Feb-91 by jiao)
  (Supers ORIGINATOR)
  (ClassVariables)
    (InstanceVariables
      (Possibleorders (OrderA OrderB OrderC OrderD OrderE OrderF OrderG)
        doc (* possible customer order list))
      (tokenName HMS-CUSTOMER-ORDER doc (* specified customer token))
      (productorderdata NIL doc (* product order record))
      (productorderdataflag NIL doc (* product order data flag))
    (availableorderlist
      ((OrderA (name1 1 75 100 NIL))
       (OrderB (name2 1 75 95 NIL))
       (OrderC (name3 1 75 100 NIL))
       (OrderD (name4 1 65 85 NIL)))
```
A new token is generated according to the order type with the physical representation of the order name. The interval between two orders can be defined by using a random distribution process, Poisson process [132]. The mean value of the distribution can be changed by the user which represents a different input level. This value is usually determined by the specified production output level. The STEM function of GetTokenInterval method is defined as follows:

\[
\text{Method}\hspace{1em}\text{((HMS-CORDER-ORIGINATOR GetTokenInterval) self)}
\]

"Determines the interval between tokens - returned value is the time that will elapse before the next token is generated - to get input data or POISSON Distribution."

\[
\text{(if (NOT (EQUAL (@ ORDERORIGINATOR orivariable) 0))}
\text{ then (POISSON (@ ORDERORIGINATOR orivariable))}
\text{ else (POISSON 10))}).
\]

ORDERORIGINATOR used in the method definition is an instance of HMS-CORDER-ORIGINATOR class. It should be noted that this process can be easily modified by the user according to his own specification.

The time generating the first order can also be defined by the user. This means that the user can control the start time of the simulation execution. In the simulation model, it was set to be 1.
Since a customer order presents general requests from the customer, the model should be capable of allowing different users to specify their own customers. This implies that user access to the model should be available. To allow this, two possible ways have been identified. One is to change the value of instance variables defined in HMS-CORDER-ORIGINATOR class. The other is to enter related information through the specialised interface (refer to Chapter 11).

**Different Market Customer Order Originator.** The marketing activity varies for different types of companies. Although the order interval and product quantity can be easily changed (as discussed above) in the model, one case should be considered, that of a different market existence. This means it is different from general customer activity in terms of quantity and the happening sequence. The possible approach to dealing with this special customer order is through the extra shift, overtime or subcontract [266]. Thus the general functions and class definition of this object HMS-DMCORDER-ORIGINATOR are the same as those in class HMS-CORDER-ORIGINATOR but with one exception. The difference lies in the definition of a method called Start. It controls the function to generate orders to the simulation. The rules are:

- **IF** there is such a kind of customer order,
- **THEN** new orders are generated;
- **ELSE** Start function is put to be NIL.

**Product Terminator.** Product terminator simply receives the completed products and disposes of them from the simulation. This node tracks all products and provides the general product information, such as order type, order quantity, order arrival time, start time, finish time, shipping quantity, and due date.

Usually products arriving at this terminator come from two areas, inventory or production area. The first case means that there are some available products at stock. They can be released to the customer immediately after receiving the customer order by the sales department. In this case, the information of product quantity in inventory should be updated. This usually happens to a company with Make-To-Stock type production [266] [87].

**10.5 Company Level**

**10.5.1 Initial Comments**

In general, the structure and operating strategies of one company are usually to some extent different from any other company. It is therefore difficult to generate a generic model which can represent each individual company. In addition, many areas are contained in a
company (Figure 3.4). Certain assumptions have to be made at this level without the loss of generality and capability. The discussion in Chapter 3 has identified that sales and marketing, engineering and production are the key areas with regard to decision making in production planning, control and manufacturing. Thus only these three areas are involved in the simulation model [155].

The interaction between different areas in a company are usually very complex (Figure 7.2). Due to the time limitations and the computing environment only one way information flow is modelled, which is depicted in Figure 9.3.

10.5.2 State Description

Sales and Marketing. Sales and marketing receives customer orders and releases them to production areas or releases available products to customers. This function is represented as a composite node in the simulation model, including several nodes.

--INVENTORY. The inventory node is defined functionally to manage stock and generate production orders accordingly. Inventory usually consists of supplies, raw materials, in-process goods and finished goods [267]. In the model, it mainly presents the finished products. At this node, two product shipping rules were defined to control the way to release products from inventory: releasing the whole order quantity at one time (rule 1) or releasing the available products in stock first and then releasing the rest after they are completed in production areas (rule 2).

In the first case, when a customer order arrives, the model first checks the product quantity in stock and the order quantity first. If the product quantity in stock is greater than the order quantity, it will release the whole order at one time and reduce the product quantity in stock by order quantity. If the product quantity in stock is less than the order quantity, the production order quantity is generated with the quantity reducing the order quantity by product quantity in stock.

In the second case, the decision made is nearly the same as in the first case. The only difference is that if the product quantity in stock is less than the order quantity, the available products are shipped to customer first [267] [266] with the quantity which is equal to the quantity in stock.

Thus different rules and different inventory status lead to different destinations for the output from this activity (Figure 7.3). This conflict is solved in the simulation model as following (three possible destinations):
When the product quantity in stock is greater than the order quantity and the product quantity left in stock is greater than the re-order quantity required by the inventory management [246]. Re-order point here presents a predetermined point that when the inventory position reaches it, an order for a fixed number of units is placed [266].

This occurs when there are not enough products available in stock with the shipping rule 1 and when the product quantity in inventory is equal to 0 with the shipping rule 2.

There are two situations when this occurs. When shipping rule 1 and 2 are applied, the product quantity left in stock is less than the re-order point after the whole order has been taken out from the stock. Therefore a production order has to be generated and released to the production area. When rule 2 is used and there are not enough products in stock but it is not equal to 0. This means that the available products are released to the customer and a production order is released to production area.

This node simply releases the products from the inventory to product terminator. There are no decision making issues concerned in this node.

At present, this node receives customer orders from inventory and releases them to engineering or the production area accordingly. The new product flag is defaulted to NIL. This means that customer orders are sent to the production area only.

Engineering. The major function of engineering is to design the product to meet the customers specification and to generate drawings and process plans [155]. At present, all customer orders do not go through this area. Suppose that all the production routings are pre-defined. Furthermore this leaves the possibility to link with other packages of CAD or CAPP. The simulation model only analyses the system performance and provides a design aid for manufacturing organisations. Engineering here is defined as a simple node which only takes some active time to perform certain function (Figure 9.3).

Production. Production is a key area in which all the other three levels have been included (shop, cell and machine). Therefore, the state description of company level at production area only covers the forecast and production planning. Manufacturing is discussed at shop level.

Forecasts are estimates of the occurrence or magnitude of uncertain future events [171]. The purpose of forecasting is to use best available information to guide future activities toward organization goals [266].
In many manufacturing organizations, sales forecasts are used to establish production levels, facilitate scheduling, set inventory levels, determine manpower loading, make purchasing decisions, and aid financial planning [267]. Therefore, many different forecasting bases exist: sales revenue, physical units, cost of goods manufactured, direct labour hours, and machine hours. The selection of a forecasting base is dependent upon plans for forecasting the necessary factor requirements [266]. For the production planning in this project, physical units (demand) are used. A general forecast format is summarized in Figure 10.2.

Forecast node in the model is a router process node. Its function is to get related forecast data of the product (quantity in each period) and send the data to production planning. Upon receiving the customer order, it first decides the production quantity for that order. If there are enough products in stock for the order, but the remaining quantity is lower than the re-order point, the production order is defined with the quantity which is equal to product Economic Order Quantity (EOQ). The size of an order that minimizes the total inventory cost is defined as the Economic Order Quantity.

If the product shipping rule 1 is applied in inventory node, the quantity of production order is equal to product EOQ plus order quantity when the product quantity in stock is equal to 0. When the product quantity in stock is not equal to 0 but less than the order quantity, the production order is defined with a quantity \( \text{ProductEOQ} + (\text{orderquantity} - \text{productinventorydata}) \).

After calculating production order quantity, the forecast data is searched and used to generate production plans. This data is then converted into net requirements for the product according to the safety stock and beginning inventory of the product. Since forecast data is the key input to production planning, the model has been designed to allow the user to enter this information.

Product safety stock is the quantity left in each period in case there is an emergency, which is usually defined as the percentage of production requirement [246] [87]. This value can also be defined by the user. If the user doesn’t enter this percentage, the default value 10% is used. The beginning inventory of the products is \( 0 + \text{safety stock} \). Therefore, if the forecast data is defined as:

\[
(100 \ 90 \ 100 \ 105 \ 80 \ 100 \ 105 \ 110 \ 90 \ 80 \ 95 \ 100)
\]

the safety stock will be defined as:

\[
(10 \ 9 \ 11 \ 10.5 \ 8 \ 10 \ 10.5 \ 11 \ 9 \ 8 \ 9.5 \ 10)
\]

and the beginning inventory is:

\[
(0 \ 10 \ 9 \ 11 \ 10.5 \ 8 \ 10 \ 10.5 \ 11 \ 9 \ 8.5).
\]

153
The net demand quantity is the gross demand plus product safety stock minus product beginning inventory:

\[(\text{perioddata} + \text{productsafety} - \text{probegin})\].

They are summarised in a table shown in Figure 10.3.

---Production Planning. Production planning is concerned with the overall operation of an organization over a specified time horizon [266]. From forecasts and customer orders, production planning determines the human and material resource necessary to produce the output demanded in an efficient manner [87] [266]. The production planning method will take the expected demand per period and develop a production plan from it. The general methods range from simple charts, through linear programming models to sophisticated nonlinear and heuristic search techniques. Usually several strategies should be concerned in determining production plans, such as equipment, personnel, materials, overtime, extra shifts and subcontracting.

Production planning in the model is defined as a cNode which contains master production planning, master production scheduling, and material requirement planning (Figures 7.5 and 9.6). The overall function is to generate detailed production plans for the components of which the end product is composed.

---master production planning. The master production planning node in the model determines the production plan according to the forecast data and available production capacity. If the production capacity is greater than that of the planned one, the plan remains unchanged. If it is less than that of production plan, it will be changed to the quantity of production capacity. This is based on the strategy: regular production. Since it is difficult to consider various strategies such as an example in [266]. The calculation is shown in Figure 10.4 [266].

---master production scheduling. A master production schedule presents the types and quantities of products to be provided in each time period in the future [87] [266]. It is derived from the production plan but contains detail. The production plan deals with the aggregated plan for total output, while the master schedule usually relates to specific products or end items [266].

Master production scheduling sometimes is the only production plan method used, with the detail production information [87]. In this model, the master production schedule presents the day by day production plan with the consideration of working days in each period, but not the production status on the shop floor. With the available information on working days and quantity in each period, the schedule shows the quantity each day in each period. An example is shown in Figure 10.5.
In the simulation system, the model generates production schedules according to the working days and the final production plan, showing how many products are required by which day.

If the production type is MTS, the production orders are then generated according to the master production schedule. If the time on schedule minus product lead time multiplied by product quantity is greater than time now, the production orders will be generated with the quantity of planned net requirements (planned time - (product lead time * order quantity)). If the time now is greater than the planned time in master production schedule, no production orders are generated.

In summary, the generation of production order in the model is determined by the inventory status, production type of the company etc.

---material requirements planning. MRP is concerned with both production scheduling and inventory control [266]. The three major inputs of an MRP system are the master production schedule, the inventory status records, and the product structure records. The master production schedule outlines the production plan for end items. The product structure records contain information on all materials, components, or sub-assemblies required for each end item. The inventory status records contain the on-hand and on-order status of inventory items. Thus MRP takes the master production schedule for end items and determines the gross quantities of components according to the requirements given in the product structure records.

The MRP node in the model receives the production order from master production schedule and generates the material purchasing orders and production orders for each component. According to the product structure, processing time and due date, sub-orders are generated and released to the shop level sending the sub-order to the right place at the right time [266] [132]. In the model, the stock status of components and open orders are not considered. Therefore, the manufacturing and purchasing plan is just for the arriving production order.

Since the model will deal with component rather than customer orders from here on, the node is therefore defined as a specialised originator with the condition that when an order arrives, it starts to generate sub-orders with the related data assigned to them: lead time, start time, order due date, batch size, routing, and shipping quantity.

The sub-orders are presented in a new token class: HMS-SUBORDER. This class defines sub-orders for components and sub-assemblies with all the required variables defined as instance variable.
In this node, an important variable is the bill of materials (BOM). It lists all of the sub-assemblies, parts, and raw materials that go into a parent assembly, showing the quantity of each required component to make an assembly. It shows how much of what material is needed in what order to manufacture a product. An example is shown in Figure 10.6. This variable is defined as a list which contains all required information, providing the following advantages, clear presentation, easy modification and fast searching.

The new sub-orders are generated based on the structural data provided. For each individual sub-order, all the required variables are assigned. They are then released to the job-schedule which dispatches these sub-orders to different shops.

10.6 Shop Level

10.6.1 Initial Comments

At this level, the released orders from materials and capacity management are handled [233]. It includes both production activity control and manufacturing activity. In the IDEF0 representative model, activities defined in Figures 7.6 and 7.7 are converted into one cNode in the IDEF0-like simulation model, namely manufacturing depicted in Figure 9.7. It consists of several nodes, job-sender, job-scheduler, shops, assembly, inspection, shipping and stock.

This may cause some problems. One is the inefficiency for the transmission from IDEF0 to the structural creation of the simulation model. The other is that it is not easy for the user to understand. Thus the integration of IDEF0 and STEM is identified as a further requirement (refer to Chapter 17).

Eight shops have been defined in the model. For any particular case, a new shop can be added into the model. The cNode shop added should contain three nodes: input store, working area, and output store. They have been defined as the specialised classes in the model. The detailed procedure of adding a new cNode into the model is found in Appendix V.

10.6.2 State Description

The shop level in the model deals with sub-orders from material requirements planning and dispatches them to different shops according to the scheduling rules applied.
Manufacture. Manufacture is a cNode which represents the real production process in the company. It receives sub-orders from production planning and sends the completed products to customers directly or puts them in stock. The later case means that the production order was generated according to the stock quantity or production plan not the customer order.

--Job-Sender. This node simply deals with the coming sub-order and release it to the job-scheduler. It receives all sub-orders from production planning at one time and then sends all of them to the Job-scheduler node. No processing time is needed by this node.

--Job-Scheduler. In order to control the situation [266], two principal methods for scheduling are usually used, backward scheduling to meet a deadline and forward scheduling to produce as soon as possible [56] [245].

Job shop scheduling usually involves allocating jobs to specific work centres, prioritising all jobs at each work centre, revising priorities as changes occur and monitoring the process of jobs. The goals of job shop scheduling usually are [266] [189]:
--high percentage of order completed on time
--high utilisation of workers and facilities
--low in-process inventory
--low over time.

The allocation of jobs to work centres is called shop loading. Loading determines the work centres to receive the jobs. It assigns jobs to the work centres, but it does not necessarily specify the order in which jobs will be performed [266] [61]. Therefore once jobs are loaded, the next task is to sequence them. Sequencing frequently referred to as dispatching, establishes the priority of jobs at a given shop. Both allocation and sequencing have been assigned to this node. At shop level, the allocation is mainly based on the process plan which determines the route of each sub-order.

The job-scheduler node in the model at the shop level receives sub-orders from the MRP node and decides when, where and in what sequence to release these sub-orders [79]. It simply assigns jobs to shops without regard to capacity limitation since the capacity of working area in each shop has been defined as infinite. This means that the infinite loading method is employed [266].

Priority rules are used to rank the jobs waiting so that the decision can be made about the next job to be performed. Several priority rules have been used in this model to decide the sequence to release sub-orders to different shops:
--first come first served
--last come first served
--shortest total processing time
--longest total processing time
--early start time
--early due date.

With the first come first served rule, the node dispatches sub-orders in the order they arrive. The sequence of sub-orders generated by MRP are according to the start time. In contrast to this rule, last come first served releases the sub-order in the opposite sequence. These two rules only concern the arrival sequence [56] [132].

Considering processing time, two other rules have been implemented, shortest total processing time and longest total processing time. Total processing time means the time spent in each shop. If detailed machine level is concerned, several operations may be involved in a shop.

The last two rules are concerned with sub-order start time and due date. They are calculated based on the product lead time, product due date, sub-order processing time, and off set (refer to Appendix VII).

--Shop. Shop is defined as a cNode in the model representing the processing areas arranged in one building or several buildings to carry out certain types of machining processes, such as press, fabrication etc. With regard to the spatial structure of a shop three nodes have been defined in each shop: input store, working area, and output store.

--input store. Input store is the area to hold the coming sub-orders and send them to the working area. Sub-orders usually pass through two shops. After having been manufactured in the first shop, the completed component is required by sub-assemblies or other components in the second shop. Therefore, several decisions have to be made for each coming sub-order.

Once a sub-order arrives, the input store node checks its status first. If a sub-order arrives at its first shop, the model decides whether it requires any other components or not. If it does not require any other sub-orders, it is simply released to the working area. If it needs other sub-orders, it cannot be released to the working area until all required components are available. If a sub-order arrives at its second shop, it will be required by another sub-order.

--working area. Working area represents the functional part of a shop that processes components. One important parameter is its capacity which illustrates the maximum quantity of parts a shop can produce at one time. To decide the dynamic work in process in the working area, the capacity of working area has to be set great enough to machine all arriving components. Therefore, at present, the sub-order can be processed immediately. Otherwise, the sub-order will be put into a queue till the capacity of working area is available.
The processing time of a component in a shop varies. It is defined in a time range based on its standard lead time. The factors which can influence its processing time usually involve machine breakdown, information delay etc. Since the case changes from one company to another, the model is designed to allow the user to enter early or later variability of lead time describing the range of time change.

After having processed the components, the working area has to decide where and when to send the completed components with regard to process plan and material movement rules. Two rules are implemented:

-- waiting till the due date
-- releasing immediately after processing.

In the first case, if the finishing time at working area is earlier than its due date, the node releases the component to output store, otherwise the completed component is released to next shop directly. In the second case, completed components are immediately sent out to next shop.

There is one situation which should be considered: if the final product is sent back to the shop again after inspection, it should be directly released to the working area and the re-worked product is then sent to the inspection area again.

-- output store. Output store is an area to store the completed components if required. For each arriving sub-order, the processing time at output store is its due date minus its finishing time at working area. Then they are released to the next shop through the transporter.

-- Assembly. Assembly is the same as other shops in the model. It is individually structured in the model simply to show the assembly function in a manufacturing system clearly and make the distinction between general processing shops and the assembly shop.

-- Inspection. Inspection presents the test or final inspection function in the company. Usually there is a test area in each shop for component inspection, the end product test is done depending on the customer request. Thus the model allows the user to decide whether the product needs to be inspected or not. It is defined in the model, involving several areas. Test node presents the inspection area. The detail of this area is not considered since the node is defined here only presenting the test function (Figure 9.12).

-- Transporter. Transporter presents the components delivery facilities in a company. Usually transporters are available to transport components between different areas. Therefore, once components are completed in one area, they are released to this node and the transporter decides where to send these completed components. This node may need some time or need no time to perform the transportation function.
--Shipping. Shipping shows the activity to dispatch products to customers. It is defined as a simple node in the model with one more function. It has to decide where to send the finished products, to customers, to stock or both. These decisions are made according to the production order generated at sales.

--Stock. Stock is an area storing materials, components and products. Its status influences the decision making at sales and marketing and purchasing. It is defined as a terminator node in the model to present the physical area of inventory, while the inventory node in the sales cNode functionally acts as a decision making node. It keeps the record of products, including product inventory data, product finishing time etc. The available product inventory data is the same one used in inventory to record the product quantity in stock.

--Outside. Production planning and master scheduling establish the manufacturing plan of products to be produced during a given time frame. MRP takes the master schedule for end items and calculates the plan for all dependent demand items composing the end items. Manufacturing and purchasing are responsible for executing the overall material plans [266]. Thus purchasing is functionally the same as manufacturing.

Outside node is defined in the model to represent the purchasing function. Therefore purchase orders are generated from MRP and released to this node for purchasing purposes. A purchase order usually contains information on part quantity and lead time. Thus the sub-order arriving at this node represents a purchasing order not the component released to other shops.

Till now, major activities contained at company and shop levels have been described. It has been found that these two levels cover major decision making and manufacturing functionalities. They are sufficient enough to quickly estimate system performance, providing the approximate modelling of manufacturing organizations. Thus the approximate model has been defined to contain only these two levels. The detailed analysis model is extended including two other lower levels, cell and machine levels (Figure 10.7).

At present, these two detailed levels are only defined in one shop (Figure 9.13) with the purpose of indicating model capability and the consideration of model size. They can be easily extended in other shops as well. The level selection depends on the user's purpose, thus the model makes it possible through the user interface.

10.7 Flexible Machining Cell Level
10.7.1 Initial Comments

Within a shop, various manufacturing areas can be defined [233]. They may be flexible production systems, production islands, kanban routes, processing centres, assembly islands etc. In shop 1, it is possible to extend the detailed layout of the shop. It consists of a flexible machining cell and several stand alone machines. These machines are discussed in the next section 10.8. This section is concentrated on the flexible machining cell, one category of flexible production systems [67].

The flexible machining cell would include automatic storage and retrieval systems, automatic material handling systems, robots and numerical control machine tools. It is considered as a system concept for medium volume manufacture with very restricted variety [44] [205] [218] [175] [274]. The flexible machining cell in the model is a simple flexible manufacturing cell, involving load/unload station, work stations, and an AGV (Figure 9.15).

To perform the required operations, the flexible machining cell should have the capability to deal with both workpiece flow and tool flow [205] [250]. Because of the purpose of the simulation model and the time limitation, the tool flow in the cell is not considered.

10.7.2 State Description

The flexible machining cell is defined in the model to enhance the flexibility and reliability on model applications. The operation and control of a flexible machining cell is a complex task. The research studies by other theses on this subject can be found in [175] [274] [286] [64], with concentration on different aspects and by using different design and analysis methods. In the following section, three major elements in the flexible machining cell are discussed.

Load/Unload Station. Load/unload station is one of the auxiliary stations in a cell. It fixes a batch of components onto a free pallet and unloads the pallet from the station off to the load buffer at the load/unload station [83]. In addition to this, is also loads a pallet from the local buffer of a load/unload station onto the station. The essential requirements of a load/unload station include a clean support for the pallet in a position accessible to the transporter, allowing access around the pallet to permit the loader to remove and load workpieces [274].

The pallet capacity is an important parameter which has to be defined in the model. The component can be sent to the work station only when it is palletised (pallet is available). The pallet capacity limits the component number moving within the cell. After a component
is completed at the cell, the pallet capacity is incremented by 1. When the component is palletised, the pallet capacity is reduced by 1 so that if the pallet capacity is equal to 0, it means that no pallet is available.

If the number of parts stored at load/unload station is less than the defined pallet capacity, the arriving component may be immediately sent to the work station. If the number of parts at load/unload station is greater than the pallet capacity, the component will be put into a list waiting for the pallet. When a pallet is available, the component will be palletised and sent to the work station.

**Work Station.** The capabilities of a flexible machining cell are uniquely identified by the machines it contains. For different types of parts, various types of machines may be found in a cell, such as horizontal-spindle machining centres, vertical machining centres, CNC machines and other special purpose machines [206].

Usually, a palletised part is sent by a transporter, such as a AGV, to the work station. The work station loads the workpiece onto the machine, manufactures the parts, changes tools if necessary and unloads the completed parts. Therefore, a work station has to contain a temporary buffer, a tool chain or magazine, and cutting facility. The work station in the model is assumed to have infinite local buffer capacity so that no blockage could occur. When a part is finished with its current processing, it can be unloaded immediately into its buffer [98].

When the work station is busy, the AGV cannot send any component to it. When it is inactive, the AGV can transfer component to it. However at that time, the AGV’s status has to be checked. AGV cannot send any component until it is free.

The processing time at each work station involves real cutting time, loading and unloading time.

**AGV.** AGV is a vehicle that transfers parts between stations or back to the load/unload station. It is one of three categories of pallet-movement facilities in a cell, inclusive AGV, roller conveyor and robot. The guidance and control of AGVs can be defined into three basic forms: rail-, antenna- and tow chain-guided [98] [83]. It is different from the transporter implemented in general shop level in the sense that an AGV cannot transfer any component until the work station is free. Therefore, when the component arrives, AGV first checks the status of the work station defined in the route plan. If the work station is idle, it immediately sends the component to that work station, otherwise it puts this component into a queue.

AGV also has to transfer completed components. Therefore, the work station has to release finished components to AGV when it is free.
10.8 Machine Level

10.8.1 Initial Comments

Machines are major and essential elements contained in manufacturing systems. Various types of machines exist for taking different operations, such as lathe, drilling machine, milling machine, grinding machine etc [214] [140].

The arrangement of machines in shop 1 are defined in a cNode (Figure 9.14) including four kinds of nodes, component dispatcher, machine, transporter, and exit. Nine machines are contained in the model. The new machines can be easily added into the model. The detailed procedure is found in Appendix V.

10.8.2 State Description

MASender. This node is a functional point in the system which doesn’t exist in a real manufacturing system. It has two functions. One is to keep the related information of each component passing through this shop. The other is to decide the destination of the component, based on the component type and routing plan information.

Machine. The operation of a machine usually includes setting up, cutting and changing tools. Setting up indicates the action of fixturing and positioning parts into fixtures. The setting up and changing tools is usually done manually in contrast to the operation in the flexible machining cell where this is done automatically. In the model, these actions are not separated, only processing time is defined to present the time spent in setting up and cutting.

There is also a storage space around the machine which stock piles the arriving components and completed components. Usually the quantity is not restricted but the finished components should be transported to other machines for further operation to reduce the level of work in progress.

For the components queued at each machine, certain scheduling rules have been implemented to sequence their processing order:

--first come first served
--last come first served
--shortest processing time
--longest processing time
--shortest total processing time
--longest total processing time
--early due date
--least slack
--smaller slack ratio.

When a sub-order arrives at a machine, the system first checks which schedule rule has been selected. Once a rule is selected, every machine should use the same rule.

The first two scheduling rules consider the arrival pattern of sub-orders. If first rule is used, the arriving sub-order is put into a queue. Each time, the first one in the queue is machined. If the second rule is selected, the sub-order is put in front of the queue so that when machine finishes processing one component, the one at the head of the queue will be machined. The method of these two rules are defined as follows:

(Method
((HMS-MACHINE PartRelease1) self)
"Method to release the part according to the arrival sequence."
(if (EQUAL (GET-ACTIVITY) 'BUSY)
  then (QINSERT package)
  else (PUT-ACTIVITY BUSY)
       (- self\ProcessToken package)).

(Method
((HMS-MACHINE PartRelease2) self)
"Method to release the part in the reverse sequence of arrival order."
(if (EQUAL (GET-ACTIVITY) 'BUSY)
  then (QPUSH package)
  else (PUT-ACTIVITY BUSY)
       (- self\ProcessToken package)).

The other seven rules are concerned with processing time or due date [283]. Therefore, the queue which contains sub-order type and related time variables is re-ordered according to certain time constraints. For example, if the shortest processing time rule is used, whenever the sub-order arrives at the machine, the queue (if exists) is re-ordered to put the sub-order with the shortest processing time in front of the queue. The method definition is summarised as following:

(Method
((HMS-MACHINE PartRelease3) self)
"Method to release the part with the smallest processing time first."
(if (EQUAL (GET-ACTIVITY) 'BUSY)
  then (QINSERT package)
       (for package in (REVERSE (QCONTENTS)))

164
do (if (EQUAL (EVAL (CONS 'MIN (for a in (QCONTENTS collect (@ (X-RAY-PACKAGE a) machiningtime)))) (@ (X-RAY-PACKAGE package) machiningtime)) then (QDELETE package) (QPUSH package) else (PUT-ACTIVITY BUSY) (- self \ProcessToken package))).

The processing time of the sub-order includes setting up time and cutting time. The tool management is not considered in the model and it is assumed that the tools are always available.

**MTransporter.** The machine level transporter is similar to the shop level transporter. It has to decide where to send the completed components according to certain information contained. The related information includes component type, the route plan, operation completed, and next operation. The difference is that it moves only within a shop not between shops.

In a process plan, the time provided usually contains processing time and moving time. Thus it takes some time for the transporter to transfer components.

The sub-orders arriving at the machine level transporter contain all the required information. Therefore, this node can decide where to send the sub-order.

*IF* sub-order first operation flag is T,

*THEN* the node sends the sub-order to its second operation machine.

After several operations, the completed sub-order has to be released to other shop.

*IF* sub-order destination is defined with name 'exit,' 

*THEN* the sub-order finishing time is now;

*the node sends the sub-order to the next destination (other shop).*

**S1-Exit.** Different from the machine nodes in the model, this node is defined to send the completed components according to the finishing time at the shop.
10.9 Discussion

This chapter describes manufacturing functionalities contained in the model. It identifies major activities happening at company, shop, cell and machine levels, it therefore provides an environment for total manufacturing organization modelling. However, the model is unable to simulate complex interactions within and between these levels.

The model is defined with two different levels with regard to the knowledge and data it contains and the functional levels it covers. This enables the user to use the model for different design stages. Based on the recognition of important knowledge and data contained in the system, the model allows the user to design his own system by selecting different operating strategies and entering different data (Figure 10.8).
<table>
<thead>
<tr>
<th>Order Number</th>
<th>Customer Name</th>
<th>Unique Product Number</th>
<th>Description</th>
<th>Quantity</th>
<th>Shipping Time</th>
<th>Time Now</th>
<th>Unit Price</th>
</tr>
</thead>
</table>

Figure 10.1  
**Customer Order Format**

<table>
<thead>
<tr>
<th>Unit Machine Volume for the Previous Year</th>
<th>Monthly Period</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine</td>
<td></td>
<td>901</td>
<td>902</td>
<td>905</td>
<td>910</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>901</td>
<td>40</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine</td>
<td></td>
<td>902</td>
<td>40</td>
<td>11</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine</td>
<td></td>
<td>903</td>
<td>40</td>
<td>8</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projected Unit Machine Volume for the Future Year</td>
<td>Monthly Period</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>Total</td>
</tr>
<tr>
<td>Machine</td>
<td></td>
<td>901</td>
<td>902</td>
<td>905</td>
<td>910</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>901</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>902</td>
<td>40</td>
<td>10</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>903</td>
<td>40</td>
<td>10</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 10.2</td>
<td><strong>Product Forecast Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Beginning inventory</td>
<td>0</td>
<td>100</td>
<td>150</td>
<td>300</td>
<td>270</td>
</tr>
<tr>
<td>2. Forecast demand</td>
<td>1,000</td>
<td>1,500</td>
<td>3,000</td>
<td>2,700</td>
<td>3,000</td>
</tr>
<tr>
<td>3. Safety stock</td>
<td>100</td>
<td>150</td>
<td>300</td>
<td>270</td>
<td>300</td>
</tr>
<tr>
<td>4. Production requirements</td>
<td>1,100</td>
<td>1,550</td>
<td>3,150</td>
<td>2,870</td>
<td>3,030</td>
</tr>
</tbody>
</table>

10% of a month's forecast is used as safety stock.

Figure 10.3  
**An Example of Net Production Requirements**

167
<table>
<thead>
<tr>
<th>Time Period</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Demand</td>
<td>10</td>
<td>15</td>
<td>30</td>
<td>27</td>
<td>30</td>
<td>16</td>
<td>10</td>
<td>18</td>
<td>26</td>
<td>30</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Maximum regular production/period: 19
Maximum overtime production/period: 4
Regular production cost: $30/unit
Overtime production cost: $35/unit
Subcontracting cost: $37/unit
Inventory holding cost/period: $1/unit

What is the optimum production plan for the next 12 months (assume beginning inventory is zero, desired ending inventory is zero, and no stockouts can be tolerated)?

<table>
<thead>
<tr>
<th>Period</th>
<th>Demand</th>
<th>Beginning inventory</th>
<th>Beginning production</th>
<th>Regular production</th>
<th>Overtime production</th>
<th>Subcontract</th>
<th>Ending inventory</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>10(1), 7(3), 2(4)</td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>9</td>
<td>15(2), 4(3)</td>
<td></td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>13</td>
<td>19(3)</td>
<td>2(4), 2(5)</td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>27</td>
<td>6</td>
<td>19(4)</td>
<td>4(4)</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>2</td>
<td>19(5)</td>
<td>4(5)</td>
<td>5(5)</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>0</td>
<td>16(6), 1(11)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>1</td>
<td>12(7), 7(11)</td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>8</td>
<td>10(8), 6(10), 3(11)</td>
<td></td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>17</td>
<td>18(9), 1(10)</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>26</td>
<td>18</td>
<td>19(10)</td>
<td></td>
<td></td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>11</td>
<td>19(11)</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>0</td>
<td>15(12)</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

\(a\) The quantity to be produced is entered in the row associated with the production time period. The parentheses immediately after the production quantity indicate the time period in which it will be demanded.

\(b\) The beginning and ending inventory columns are determined after all the strategy variables are assigned.

Figure 10.4

An Example of Production Plan
<table>
<thead>
<tr>
<th>Week</th>
<th>Days Per Week</th>
<th>Planned Quantity</th>
<th>Daily Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>76</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>95</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>95</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>95</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>103</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>105</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>96</td>
<td>19</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>94</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>94</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>94</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>94</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>75</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10.5  An Example of Master Production Schedule

Figure 10.6  An Example of Bill Of Materials
Figure 10.7

Hierarchical Modelling
Input Data and Rules for the Simulation Model

- **Customer Order**
- **Product Forecast**
- **Inventory**
- **Product**
- **Production Capacity**
- **Production Variability**
- **Component**

- **Customer Order Release**
- **Production Control**
- **Material Control**
- **Abstraction Level**
- **Scheduling**
- **Inspection**
- **Inventory Management**
Chapter 11
MODEL SPECIALISED INTERFACE

11.1 Introduction

This chapter overviews the user interface of the simulation model. The major topics include requirements for the user interface, STEM user interface facility, model defined interface and related menus contained.

11.2 Research Objective

The user interface is a major element contained in simulation models [130] [185] [229] [274]. It is a module for data acquisition and transmission capabilities with the simulation executive. Supported by LOOPS, STEM has provided powerful capability for model building, modification and execution [14]. However, for this specialised simulation model with manufacturing functionality contained in it, more user-friendly interface is needed to enable the user to focus on problem definition and analysis of results. Thus the purpose of this chapter is to identify the general requirements for an user interface and describe the user interface built in the simulation model.

11.3 Interface Requirements

Simulation, as a tool for problem analysis can be made more accessible and more efficient by developing an intelligent, flexible interface for building the model, manipulating the model, and analysing the model [130]. Thus interface is a critical feature of the modelling system. Therefore, requirements for the interface in creating a simulation model and selecting a simulation package should be considered carefully.

The first requirement for an interface is that it should enable the user to focus on problem definition, simulation execution, and analysis of results [130]. It provides the user with the capability to easily develop and modify the model himself without requiring more knowledge about the simulation language or programming environment [229]. The user also should have the control over the simulation running, suspending, and debugging.

Another important requirement for an interface is the flexibility. This allows the user easily to access any part of the simulation model at any time he wants to. Most interactions with the system are through menus, or pop-up menus [229].
Explanation capability is also a requirement for a user interface. Thus the selection of an option through the menu may bring up the meaningful prompts to the user. It should be realised that user friendliness is an important characteristics of any computer system, particular those providing intelligent decision support [32].

For the integrated knowledge base hierarchical manufacturing organizations modelling system, another significant feature is that the interface should allow the user to easily access to the specialised classes library. The parameters associated to the object have been defined in the simulation model. They can be modified with ease without affecting any other part in the model. The operating strategies applied in manufacturing systems may be selected from the available rules.

In addition, the interface should allow the user to easily get the output results. After every simulation run a minimum of statistical information is computed and presented to the user in a report [258], but additional analysis may be performed so that the model should provide more facilities to get those values which are worth attention. The user is of course free to choose the information he requires from the model.

Two major topics are included in the following sections, the discussion of STEM user interface facility and the model defined interface capability.

11.4 Structure of the Interface

To build a specialised manufacturing organization model, other features of the user interface have to be added thus enhancing the STEM interface capability, such as interactive data entry, interactive rule entry and output results presentation facility. Thus the aim of this section and the following sections is to describe this model defined user interface.

Provided by LOOPS environment, the interaction between the user and the modelling system can be of three types: menu driven, natural language based and graphic based [274] [241]. Among them, the menu driven interaction is the mostly widely used approach to guide the user to interact with the model [135] [50].

The interface facility developed in the model contains four menus, the global menu, data entry menu, rule entry menu, and output menu. Since it is not necessary to display all of them on the screen at one time and on the other hand this will mass the displayed layout on the screen, one menu is displayed at one time only. This is organised by a global menu.
11.5 The Global Menu

The global menu provides an access to all other interface facilities, shown in Figure 11.1. Typically these interfaces are tailored to support particular tasks that users wish to carry. In all cases, the interfaces are intended to give a user-friendly, convenient but flexible access to those facilities of the model that are particularly relevant to the input and output [50]. It contains three menus, input data menu, input rule menu, and output menu.

The global menu is opened by selecting "Create Company Menu" item on COMPANY object menu. Once the menu is displayed, the user is recommended to read model specifications provided by the menu by flick 'Model Specifications' on the menu to understand this model defined user interface and use it efficiently and accurately.

Two other options are offered by the global menu, saving input data and rule information into an input file and recording output results to an output file, which can be used for simulation model analysis.

11.6 Data Entry Menu

The interactive data entry editor of the modelling system has been built to allow the user to enter related data for the modelling system. The structural layout of the simulation model has been fixedly built and the related instance variables have been defined in the specialised classes, but they can be modified through this menu. The simulation model thus allows the user to deal with any particular case, With this facility, the user can re-define the initialised variables values. The value is obtained by selecting integers from a number pedal.

As shown in Figure 11.2, available data which can be entered include customer order data, product data, component data, and manufacturing related data. The explanation of each datum and the format used to enter the datum have been specialised in the window by selecting input help information in the menu. The conversion from realistic information to the data entered can be found in Appendix VII.

11.7 Rule Entry Menu

The simulation model includes a rule entry editor to allow the user to enter the rules with regard to both system behaviour and operation. This facility only allows selection of the available rules applied from the library of existing ones. If there are only two rules
pre-defined, the selection is made by typing T or NIL as indicated in the input window. If there are more than two rules available, the selection is made by choosing the related rule number through the number pedal.

Shown in Figure 11.3, the available rules mainly include customer orders generation, order shipping strategy, production type, scheduling rules etc. Once selected, the flag of these variables are valued. This updates the original variable value defined in the library. An rule entry example is shown in Figure 11.4 in which the available rules are listed and the user is asked to select a suitable one.

Another option available on the menu is the input rule help information. The specification on each item in the menu are displayed to give the explanation information on each item.

11.8 Data Output Menu

The STEM can provide graphical statistic outputs on certain variables and graphical textual outputs. However, the simulation model has to provide more information for serious post analysis. The output data editor has been developed to offer the facility allowing the user to get required output results.

The output results provided include general system performance output, customer order information, product information, component information, and resources information, summarised in Figure 11.5. The detailed discussion on the output results can be found in Chapter 12.

11.9 Discussion

This chapter describes the general requirements for an user interface and overviews the structure of the specialised user interface defined in the simulation model. Compared with interactive capability provide by STEM, this interface enhances the capability of STEM environment and permits the user to define the specialisation of a manufacturing system. It also allows the user to define and simulate at different levels of abstraction and provides facilities for model output collection and interpretation.
**Figure 11.1**
Model Defined Global Menu

<table>
<thead>
<tr>
<th>Model Specification</th>
<th>HardCopy Input Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Entry Menu</td>
<td>HardCopy Output Data</td>
</tr>
<tr>
<td>Rules Entry Menu</td>
<td></td>
</tr>
<tr>
<td>Output Data Menu</td>
<td>Close Menu</td>
</tr>
</tbody>
</table>

**Figure 11.2**
Model Defined Data Entry Menu

<table>
<thead>
<tr>
<th>Customer Order Data</th>
<th>Forecast Data</th>
<th>Order Release Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Status Data</td>
<td>Safety Stock Data</td>
<td>Lead Time Demand</td>
</tr>
<tr>
<td>EOQ</td>
<td>Production Capacity</td>
<td>Lead Time Data</td>
</tr>
<tr>
<td>Time in Production</td>
<td>Product Safety Stock</td>
<td>Input Stores Capacity</td>
</tr>
<tr>
<td>Work Areas Capacity</td>
<td>Output Store Capacity</td>
<td>Input Stores Variability</td>
</tr>
<tr>
<td>Work Areas Variability</td>
<td>Help</td>
<td>Model Master Menu</td>
</tr>
</tbody>
</table>

**Figure 11.3**
Model Defined Rule Entry Menu

<table>
<thead>
<tr>
<th>Subcontract Variability</th>
<th>Abstraction Level</th>
<th>Different Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Release</td>
<td>Shipping Rule</td>
<td>Production Type Rule</td>
</tr>
<tr>
<td>Material Supply</td>
<td>Material Movement</td>
<td>Information Delay</td>
</tr>
<tr>
<td>Shop Scheduling Rules</td>
<td>Scheduling Rules</td>
<td>Product Release Rule</td>
</tr>
<tr>
<td>Inspection Rule</td>
<td>Help</td>
<td>Model Master Menu</td>
</tr>
</tbody>
</table>
1> The model ignores the detailed machine level.
2> The model includes the detailed machine level.

Abstraction Level Rule Selection:
Please enter model abstraction level rule (T if is rule 1, NIL otherwise):

---

Figure 11.4
An Example of Rule Entry Environment

---

Figure 11.5
Model Defined Output Menu
Chapter 12
SIMULATION OUTPUTS

12.1 Introduction

This chapter describes the outputs of the simulation system. It first discusses the relationship between performance judgement of manufacturing organizations and outputs of the simulation model. Then it focuses on the discussions of output information and their use.

12.2 Research Objective

Output interpretation of the modelling results is another key issue that remained in the simulation research. The major problems of many current modelling tools lie in their lack of flexibility in providing customised outputs and the highly statistical non-engineering manner they are presented to the user [175] [234]. Thus the purpose of this chapter is to identify two major categories of outputs and explain these measurements to help the user understand the system performance.

12.3 System Performance versus Simulation Outputs

Manufacturing organizations are often of such size and complexity that traditional design techniques are inadequate to guarantee that the resulting design will have the required or desired cost and performance characteristics [234]. Simulation of a proposed system design provides the insights of manufacturing organizations behaviour (refer to Chapters 5 and 7).

This hierarchical simulation model has addressed some of the manufacturing issues stated in section 8.4, the need for and quantity of equipment, performance evaluation and evaluation of operational strategies (refer to Chapter 8). Therefore, carefully defined performance measurements have to be used to describe the behaviour of manufacturing organizations.

The primary requirement for outputs provided is that these measurements should be accurate enough to represent approximate system performance. However these figures sometime cannot be directly used in the actual design [243] [275].

178
The first reason is that these performance figures cannot simply provide a corresponding relation between those figures and design specifications from the practical point of view. It seems that two methods now have been studied to show the clear relation between them. One approach to doing this makes use of regression analysis [234]. The other method for predicting values of performance variables in simulation is named as perturbation analysis. However it is still unknown whether the accurate relationship can be defined [234] [274].

The second reason is that the accuracy of the collected statistics depends on the simulation execution period and the time period over which these statistics are collected [274]. Each model is run under certain initial conditions, with the introduction of the first order into steady states. Since no easy variable can indicate the termination condition, a certain time period has to be defined which is long enough to provide accurate performance.

The third reason is that the simulation model is a simplified model which is unable to contain every aspect of a real manufacturing organization and some of the required information for the simulation model cannot be directly provided by a particular manufacturing company.

In summary, the outputs of the simulation model can provide, to some extent, approximate system performance figures [109].

12.4 Simulation Outputs

The performance of this hierarchical simulation model can be evaluated in two ways. One is by adding dynamic monitors. These dynamic monitors, such as average monitor, trend monitor, and queue monitor provide means allowing the user to look at the running results. However they only help the user know those changes during simulation execution, they cannot provide serious post analysis [14] [109].

Another is through post analysis by providing certain manufacturing oriented output results. The purpose of this simulation output is to evaluate system performance through simulation experiments and provide decision making for the system design. In addition to ascertain the number of machines or other items of the system necessary to achieve a specified level of production, the control of manufacturing systems become of prime importance and the cost of holding work in progress is now recognized as a major element in a company's cost [109]. Therefore, investigating these different operation strategies and the level of work in progress can also be provided by the simulation outputs.
However, it should be realised that the performance output provided by a simulation model is constrained by the assumptions made in the modelling [274] [109]. The outputs provided by the simulation model can be obtained in two approaches. Thus according to the way they are collected, two categories of outputs have been identified. One is the collected manufacturing oriented outputs summarised in Figure 12.1. The other is the dynamic plot data obtained by the specified logger monitor and analyser in STEM. In the following sections, these two types of outputs are described.

12.5 Manufacturing Oriented Outputs

These manufacturing oriented outputs include overall system performance, product performance, component performance, inventory performance, working area performance, machine performance, work station performance and transporter performance.

The overall system performance concerns the production capacity of the system. It covers the following measurements figures:

--product order data
--product throughout
--average product lead time
--average utilisation.

Product performance is concerned with the flow patterns of each individual product in the system. The outputs which have been defined to measure this flow include the following:

--product lead time
--time at pre-production
--time at production.

Product lead time is defined as the time a product spends in manufacturing. Since a product order is divided into individual sub-orders by the material requirement planning, the total time of product in a company can be divided into two broad partition, time in the pre-production areas which as analysed later in Chapter 14 greatly influences the product finishing time and time at manufacturing which indicates the real machining time.

Component performance indicates the flow pattern of an individual part in the shops. For these two levels, statistics have been collected regarding to the following major activities:

--part at input store
--part at output store
--part at outside
Part at input and output store figures are simply defined as a list which contains all
components arriving at input store and output store in a shop. Part at outside is a record to
track parts through sub-contract, containing start time, finishing time, and lead time.

Part at working area, work station, and machine measurements record the performance
of parts at certain resources in the company. Part lead time is defined as the real processing
time, while waiting time is the time a part spends either at a machine or in a cell. At shop
level there is no queue in working area.

These figures can help determine the capacity of resources, i.e., input stores, working
areas and output stores. The time related measurements can reflect the real production
performance of the manufacturing system.

Inventory performance is concerned with stock status when a customer order arrives
at and leaves the sale department. This figure illustrates the dynamic changes of inventory
and is influenced by inventory management strategies.

Working area performance is concerned with the activities of each individual area
during the simulation experiment. Working area has been defined with infinite production
capacity. Therefore as long as the number of parts at working area is greater than 1, it means
that it is active. When the number is equal to 0, it is idle. Two categories of outputs have
been defined to indicate the working area performance:

--machining time
--idle time.

In contrast, machine performance is concerned with the activity at machine level,
which can only handle one part at a time. Therefore, when the status of a machine is busy,
it means that it is fully occupied. When it is idle, nothing is made at a machine as in working
area, it includes two types of statistics:

--machining time
--idle time.

Work station performance is concerned with the activities of work stations at cell
level. In the cell, the detailed pallitisation time, time at buffer, time at fixturing are not
concerned. The interested reader is recommended to reference [274]. Therefore it again
contains two types of outputs:
--machining time
--idle time.

Transportation performance is concerned with the utilisation of transport in a shop and an AGV in the cell during the simulation experiment. It is measured in two types of statistics:
--moving time
--idle time.

These resources performance figures represent their actual usage, therefore leading to the identification of their utilization and potential problems.

12.6 Standardised Outputs

This type of output is provided through the STEM defined logger monitor and analyser.

The logger can collect all the information about events and changes in variable values that happen during a simulation run. The stored information can be summarised or plotted by the analyser. It obtains the information by allowing the user to select from a menu of the log index files and provides a variety of utilities to allow plotting and extraction of the logged data [14]. The significant figures are:
--number of parts in input store
--number of parts in working areas
--number of parts in output store
--queue length
--time in queue.

Each shop at the approximate level contains three areas: input store, working area, and output store. With different simulation experiments, these figures can provide the changes of number of parts in each area. It is of importance to examine the maximum number in each area under different production inputs, this can help set the capacity of the input store, working area, and output store.

Queue length is a measurement figure which indicates the number of parts in each machine. The queue length will be increased when a part arrives at a busy machine. The figure is useful in measuring the machine performance and figuring out the potential bottle-neck resource.

The figures concerning time in queue can demonstrate the non-processing time of parts which shows the general flow patterns of the parts through the resources.
12.7 Discussion

This chapter describes major outputs provided by the simulation model. From the discussion above it is found that the model provides flexibility with configurable outputs to identify manufacturing performance. Different outputs can be obtained with different requests by different end users. It has enhanced the STEM post-analysis capability by monitoring more figures which can help users to understand how the system performs with respect to their initial requirements.
Figure 12.1 Outputs of the Simulation Model
Chapter 13
THE CASE STUDY

13.1 Introduction

This chapter reports an industrial case study to illustrate the use of the methodology established and to validate the integrated modelling system with the realistic industrial data and operating strategies. The company, GEC ALSTHOM Large Machines Ltd. is described in Appendix VI to provide the supporting background and knowledge. The input information used in the case study is discussed in Appendix VII. The following chapter reports the resulting output of the case study for the limited validation of the modelling system.

13.2 Objective of the Case Study

The overall objective of the case study is to test the methodology established and to validate the integrated modelling system with the realistic industrial data and operational strategies [23].

In Chapter 8, the methodology has been established for hierarchical modelling of manufacturing organizations. As the explanation for its principle, the case study should be carried out in a procedure defined in the methodology. Thus the whole process of the case study can illustrate the use of the methodology.

As a test-bed, the case study is used to examine the original principles. Due to the problems of size and complexity of manufacturing organizations, the available computer and necessary requirements for the detailed analysis, a certain degree of approximation is required. Through the case study, the integrated modelling system is challenged to show that it is an efficient approximate modelling system of a hierarchical manufacturing system to act as the design and analysis tool.

With the flexibility provided by the model to deal with different detailed levels, the emphasis is laid on the comparison of the system performance under alternative levels. This is another aim of the case study. It is believed that this flexibility can help users to quickly estimate the performance of the system or to carry out the detailed analysis as they wish. Under the same level, different input data and rules are used to test all scenarios and find out the promising one.
As a generalised modelling system, the simulation model offers the adaptability to allow the applications from different sectors. It is one of the aims of the case study to prove that this adaptability can provide the user with a general background and also specialisation capabilities. Thus the personality by different users can be achieved without much difficulty.

The company has already been involved in a simulation project [197], in which the interactions with the production control function and the manufacturing cells on the shop floor have been studied. This case study will use the integrated modelling system to describe in more detail the function of each major area in the company and the relationship between them.

The company is already looking to re-engineer the production control and purchasing system and to re-organise the manufacturing system. It is therefore believed that the application of the integrated simulation system will:

--to demonstrate the dynamics of these functions by modelling the time elements, which in itself will allow the re-engineered systems to recognize the time constraints involved which are so often overlooked,

--to help establish the manufacturing and pre-production resource capacities that are needed to fulfill a specified level of output,

--to help establish the levels of WIP that may be realistically experienced in achieving this level of output and the effects of constraining WIP,

--to test the implication of changes in the production management rules with the manufacturing system,

--to record the product lead time and spot bottle-necks in the system,

--to provide a documented base for system view leading to correct system selection based on the alternative system running.

The output results from the case study are expected to indicate the principle of the methodology is correct, the model is reliable and applicable. In addition, it is expected that they could help the company recognize remaining problems and find out the solutions.

13.3 The Case Study Experiment

As the assessment of the integrated modelling system, the test is critically carried out through the results analysis based on real industrial data and strategies. In order to do so, certain requirements for the case study have to be identified and taken into account such as realistic information, alternative tests with different information, realistic output and possible recommendation for the company to improve the production.
The case study was carried out to follow the procedure discussed in Chapter 8. As an example, it can explain how a real application is done during different phases defined in the methodology (refer to Figure 8.3), thus illustrating the use of the methodology. It should be noted that as a validation test, the case study did not follow exactly each step described in Chapter 8.

13.3.1 Problems Tackled

As discussed in Chapter 8, the issues addressed in the hierarchical simulation model are the capacity analysis of resources, evaluation of performance and operational procedures.

Based on the description of the company (refer to Appendix VI), it is found that the company is composed of 13 cells in which there are different machines to carry out certain operations. Thus the capacity analysis of resources mainly deals with the overall usage of each cell under certain level of input rates. These figures can be used either at the design stage to decide the resource requirements for the specified output level or at the operation stage to analyse the utilisation performance. For the system performance evaluation, the throughput, makespan and bottle-neck analysis can be done by comparing the output from the model with the real situation of GEC to indicate if the assumptions made in the model are reliable. Furthermore, operational strategies can be evaluated and this is done using alternative rules. Refer to Chapter 14 and Appendix VII.

13.3.2 Information Collection

As stated by Carrie [51] that any simulation needs data to work on. Thus information collection in the case study refers to the specific information collection step identified in Chapter 8.

According to the strategies identified in section 8.5, the procedure used in the information collection during the case study is to identify what information is needed for the input entry, to classify the resources of these information, to interview domain experts from different resources (key departments), to analyse the available information and to convert them into the format required by the model. As an illustration, the information collection activity in the case study has been discussed in section 8.5.
Since there are many types of products produced by the company, the first thing in determining the information required is to select candidate products. Seven small sized machines have been chosen for the case study with regard to the available information and common features found in manufacturing (refer to Appendix VII).

To classify what information can be obtained from which departments has been proved to be difficult. This depends on the overall knowledge about the company, such as the function performed by each department. It usually changes accordingly in different companies. The experience has shown that talking to the expert who has broad knowledge of the company can make this easier. In addition, it is helpful to get more information if possible to abstract useful data from them.

It should be noted that the information collection sequence used in the case study did not follow exactly the production sequence because of the availability of these experts. It has been proved that re-consulting some domain experts after obtaining the available information is helpful to fully understand the real meaning of these information, thus leading to the correct translation. The detailed discussion on information type and analysis can be found in Appendix VII.

13.3.3 The Experiment Design

As analysed in Appendix VI, two major entities flow in the company, i.e., information flow and workpiece flow. In the case study, the specific IDEF0 model of GEC has not been built since the IDEF0 representation discussed in Chapter 7 can help to identify the interaction between functional areas (refer to section 8.7.2).

As analysed in section 8.7 in Chapter 8, the model was built through two different design stages, preliminary design stage and expanded design stages. The model is therefore developed from simple to complex, containing more knowledge and information. These two design stages have no direct effects on the case study, but they are very important in model building and they provide the user a clear logic for model development.

It should be emphasised that if the model can be validated after having been verified at the preliminary design stage, the model can be proved to be more reliable. To follow this procedure, the case study was done from the approximate level to the detailed analysis level developed at these two stages respectively. Under each level, several running experiments have been planned. See section 13.3.4.
13.3.4 Strategic Planning

In order to meet objective of the case study both for the project and the company, one important issue is how to design an experiment that will yield the desired information [239]. Usually the experiment design is affected by the scope of the model, available information, computing environment and purpose of the experiment. As discussed in Chapter 10, the model is designed containing two levels, approximate level and the detailed analysis level. Under each level, there are alternative control and operational strategies available. Therefore, the case study is carried out with several running plans.

One is to compare different levels of modelling. This comparison illustrates the flexibility of the model in both quick estimation and detailed analysis and thus identifies the scope of different levels. For this comparison, input data and rules should be compatible. Furthermore the model can be run under two different conditions: random process and fixed process (refer to section 14.2). Thus under each level, two runs have been planned for the random condition and one run for the fixed condition.

The other is to highlight the effects of different input data and rules. It is aimed at testing alternatives and identifying the promising one. For this purpose, several runs have been carried out with the emphasis on key operational strategies and data since it is not possible to run all possible combinations of different inputs and rules because of the limitation of time and space. The overall experiment plans are summarised as follows:

1. Comparison of the same level under different conditions:
   --random process
   --fixed process.
2. Comparison of different levels.
3. Different strategies and input data:
   Approximate level
   --scheduling rule
   --material movement rule
   --other rules
   --time in pre-production area
   --different variabilities
   Detailed level
   --scheduling rule at machine level
   --inspection rule
   --subcontract variability.
A product contains many components, usually around 40. It takes longer computing time to complete all of them. The simulation model thus only chooses some of them to present the general performance of the system. The related component data is listed in Figures VIII.1 to VIII.7 according to the product information shown in Figures VII.1 to VII.7. Figure VIII.8 presents the process planning information for the detailed machine level.

13.3.5 Tactical Planning

As stated in [239], tactical planning involves questions of efficiency and deals with the determination of how each of the test runs specified in the experimental design is to be executed.

There are two problems concerned. One is the starting condition. Although a simulation model is built to represent a real system, the model is usually started with the system empty and idle. This is true only when the real system is activated for the first time. Therefore, it may take a certain period of time for the model to reach steady-state representative of the real world system operations.

One method to reduce the effect of this initial transient period is to run enough time to get to the steady-state. The other is to choose initial starting conditions that are more typical of the steady-state condition. As discussed in Chapter 9, the values of instance variables defined in the data base represent the initial status. Although they can be easily changed, the time cannot be eliminated for the model to approach the steady-state conditions since the model steady-state depends on not only the initial values but also the number of moving entities flowing in the system. Thus the potential solution is to run the model for longer time.

The second problem deals with the necessity to estimate the precision of experimental results and the confidence attributable to the conclusions or inferences drawn. This means that the model running times have to be determined to reduce the potential effects caused by the variability. The number of runs have been planned in the strategic planning in the consideration of available time and results accuracy. In general, more runs can lead to higher degree of precision. For more information, the reader is recommended refer to [239].
13.3.6 Running the Model

Once the running plan is determined and the input information is available, the model is ready to be tested. To run the model efficiently, the proper procedure should be followed. For the detail explanation for the model execution, refer to Appendix IV.

It has to be stated that the running of the model at this stage can help identifying some errors remaining in the model in terms of programming code or the accuracy of the input data. If there is such an error, the model has to be revised and the experiment should be repeated. This implies that changes would take place at any stage of the model development.

One decision has to be made during the running period, i.e., the steady-state of the model. This depends on the general production performance of the company. This has not been properly done in the case study for two reasons. One is that the company cannot provide the accurate data to indicate the steady-state condition. They only judge it according to the experience. The other is that the computing environment has removed the possibility because of the computing capacity. Thus the running is continued till the memory space is full.

13.3.7. Output Analysis

For each running, there are some outputs which can illustrate the system performance. The measurement figures have been identified and discussed in Chapter 12.

The output analysis mainly follows the experiments planned in section 13.3.4 to test the methodology, validate the simulation model and illustrate what the results mean. The sequence in analysing the result output is to collect the output from each running experiment, to interpret these output, to compare them with real figures from the company and to analyse the influence of the results on the assumptions made in the model and knowledge contained in the model. The detailed discussion on the output of the case study is found in Chapter 14.

13.4 Discussion

This chapter describes the industrial case study for the principle test of the methodology and model validation. The case study was done following the procedure defined in the methodology discussed in Chapter 8. The experiment explains the use of the methodology which illustrate the application capability of the methodology established.

The experience has shown that the methodology provides a systematic approach for model building and model validation. It also provides a guide-line for the user.
Chapter 14
RESULTS OF THE CASE STUDY

14.1 Introduction

This chapter analyses the results of modelling GEC ALSTHOM Large Machines Ltd. using the integrated knowledge based simulation model leading to the limited validation discussed in Chapter 15.

14.2 Initial Comments

The operation of each level in the model is influenced by the operating strategies and elements data. Therefore, the computational performance to some extent depends on the number and complexity of the rules contained in the model. The number and complexity of the rules in turn are mainly determined by the number of objects, parameters of each object.

In practice it is difficult for the company to define every aspect of the system in the fixed number quantity. A certain range of random processes in some cases can represent the realistic status of the system. The random changes contained in the model include: customer order generation interval, time in sales department, time in forecasting, time in Master Production Planning, time in Master Production Scheduling. The methods used to define these processes can be modified by using fixed number to change them into fixed processes.

The inputs to the system under these two processes are listed in Figures 14.1 and 14.2 (refer to 13.3.4). In these two figures, the presentation of input rules uses the text format with the consideration for easy and full understanding of the model operation. The computer codes of these input information are listed in Figures VIII.9 and VIII.10.

14.3 Discussion of Results

14.3.1 Introduction

The discussion of results follows the experiment plan designed in section 13.5. Each section concentrates on the results summary and implication of theses results with respect
of the overall structure and operating rules of the company, thus leading to the limited validation of the simulation model. The major aspects covered are overall performance, product performance, component performance, resources performance and work in progress.

14.3.2 Approximate Level

14.3.2.1 Introduction

The computer run time of the GEC model has been found to be about 35 minutes with seven to nine products completed for a 176 days planning horizon. Compared with half a year planning horizon, this indicates a promising potential for using this level to quickly estimate the performance of the system. It should be noted that this to some extent is not accurate since more products are completed in 176 days. The reason is that only 7 products are selected and the model is started empty.

14.3.2.2 Overall Performance

The results obtained under these two processes are summarised in Figures 14.3 to 14.5. Total product input means how many orders have been released into the system by the time of planning horizon (in days). Total product output represents the number of completed products from the system. Total product required shows the product quantity which should be finished at the end of the planning horizon. Total component input shows how many components have been released to manufacture and sub-orders to suppliers (sub-contract). Total component output is determined by the quantity of completed end products.

It is found that in the same running period, the number of products completed are changed. It is mainly influenced by the arrival time of the order into the system for each product and time in pre-production. In the same planning horizon as the random process, one more product has been finished under the certain condition. Since the fixed interval 10 is used which is usually less than the stochastic interval at random processes.

These results illustrate that the order arrival pattern considerably affect the system performance. Therefore, the selection of random processes should be carefully done to represent more realistic situation. Additionally, these figures can help establish the
manufacturing and pre-production resource capacities that are needed to fulfil a specified level of output. This means that if certain number of products is anticipated to be finished at certain time, the input data can be changed to provide useful results.

14.3.2.3 Product Performance

All products are finished nearly at the same time as their planned due date. The slight difference results from the input store variability, possible working area early or later variability and information delay variability. When product due date minus start time at production is greater than product lead time, the product can be finished on time. The completion time of products is also influenced by the scheduling rules applied at shop level.

Again this is not always the case in the company. Most of the time the product is finished later than its due date. This difference is mainly caused by the non-accurate data, time in the pre-production areas since the company cannot provide exact time for each product spent in the pre-production areas. However, from these figures possible time which can lead to the good performance may be identified.

14.3.2.4 Component Performance

The component performance under three runs is individually listed in Figures VIII.11 to VIII.17. For each component, the start time is calculated by the lead time of product, component processing time and of set. At shop level, the scheduling rule: the early start time has been selected for these three runs. In each shop, the material movement rule under three runs are the same (Figures 14.1 and 14.2). In shop 1, shop 2, shop 3, shop 4, and assembly, the component will be stored in output store if it is finished earlier than its due date. In shop 5 and shop 6, the component is transported to next shop immediately after finishing. At outside, the start time does not necessarily present the real start processing time. For example, for component E233, the planned start time is 100, finishing time is 133, the processing time is 10. Actually it was not processed until component E214 is released to outside at time 123 since E214 is the required component by E223.

From the above, the conclusion can be drawn that the finishing time of components is influenced by multiple factors, such as scheduling rule, material movement rule, capacity of working areas etc. The real production situation on the shop floor has been considered to use some variabilities to indicate the effects. It can be drawn that the model is reliable in the sense that it corresponds to the real system.
14.3.2.5 Resources Performance

The performance of working areas is found in Figures 14.3 to 14.5. Total working time represents the machining time of each working area in days. Since infinite capacity has been defined in the model, these figures can illustrate potential usage and determine the overall capacity of each shop but not real number of machines. It is found that the utilisation of some shops is not high mainly due to the input level of products, process planning and short computing time.

Thus it can be concluded that since the infinite capacity of working areas has been assumed, the resources output can only be used for the rough capacity plan at design stage.

14.3.2.6 Work in Progress

With certain level of input rate, the component number flowing in shops can be defined. Figures 14.6 to 14.8 are examples which present number of parts at shop 3 input store, working area and output store. These figures can be used to determine the area capacity under certain levels of production capacity. More plot data is found in Appendix VIII. It should be noted that the value of work in progress level is mainly influenced by the total component input rate, material movement rules, and scheduling rules.

Compared with the WIP level in GEC, these figures are lower than the real level. The incorrect figures result from the lower input rate and initial running condition.

14.3.2.7 Summary

The results under these two conditions are nearly the same. This can be found from both the general performance summary (Figures 14.3 to 14.5) and the dynamic changes in each area. One example of shop 4 input store plot data under three runs is separately presented in Figures 14.9 to 14.11. This implies that the random process at this level can be used to represent the system performance.

Since the company can only provide realistic and available data, the results are mainly determined by the input data. However, the expected output can in turn change the input data. No anticipated data is available from the company, this is not tested.

14.3.3 Detailed Analysis Level
14.3.3.1 Introduction

Since there is no flexible machining cell in GEC, only machine level is extended. The input information remains the same (Figures 14.1 and 14.2) with one exception: abstraction level selection.

14.3.3.2 Overall Performance

The results under two conditions at this level are summarised in Figures 14.12 to 14.14. It is found that two runs' results under the random process are the same in terms of processing time, finishing time, due date and arriving time. It should be noted that the same results do not mean only one case but the random processes stochastically generate the same quantity. If more runs were done, the results would change. This reflects that more runs should be planned. Under certain condition, OrderG's arriving time is 61, the due date of product G is therefore 161. Within 153 days planning time, only 6 products have been finished.

14.3.3.3 Product Performance

Again most products are finished nearly on time with the exception of product B. The delay of product B is caused by the lateness of its components, since there exist queues at some machines. These results however represent the realistic case in the company. Thus it can be concluded that the system performance varies based on different assumptions.

14.3.3.4 Component Performance

The component performance is summarised in Figures VIII.18 to VIII.24. It is found that some components are finished several days later than their due date because of the queue. From the result of waiting time of each component at each machine listed in Figures 14.12 to 14.14, one possible suggestion is to find alternative routing since there are no queues at machine 1, machine 6 and machine 7. It should be emphasised that although the delay really happens in the company, the queue is not the only reason, breakdown and delay of information are also the factors.
14.3.3.5 Resources Performance

At this level, each machine can only process one part at a time. The machining time represents the total working time not the utilisation. The waiting time represents the total waiting time of all parts at each machine. Figures 14.15 and 14.16 summarise the queue length and time in queue at machine 3 under run 2. These dynamic changes show the status of the queue and the time of each part spent in a queue. Furthermore, they can help the user to identify the potential bottle-necks in the system. It can be concluded that the utilisation of machines is not high. The reasons for this are: i) inaccurate input rate; ii) unavailable detail process planning for components and iii) short computing running time.

The transporter usage in the shop is mainly influenced by parts quantity in the shop and the moving time defined. The figure shows the total moving time and idle time. The transporter capacity is defined as infinite which means no delay happens because of unavailability.

14.3.3.6 Work in Progress

The total number of components arriving at each area in a shop is not changed at different levels, but the work in progress level is increased at detailed analysis level due to the queue at machines. In addition, the processing time of each component is another key factor. This illustrates that the results from this level is closer to the real situation in GEC.

14.3.4 Comparison of Different Levels

14.3.4.1 Introduction

This section focuses on the comparison of the results obtained from two different levels. To demonstrate the insight as provided by each study with regard to the behaviour of the system, the comparison is carried out as follows (Figure 14.17): overall performance, product performance, component performance, resources performance.

14.3.4.2 Overall Performance

The results obtained from two different levels are summarised in Figures 14.3 to 14.5 and 14.12 to 14.14. From these figures, it is found that the arrival pattern of customer orders is similar for most products, but the planning horizon at the approximate level is longer.
than that at detailed level. This is mainly determined by the Poisson process used in the model. This does not mean the production time at approximate level is longer than that at the detailed analysis level.

From the case study, it is also found that the input requirements at two levels are nearly the same. It indicates that further improvements for the detailed analysis level are needed to provide more powerful and realistic environment.

14.3.4.3 Product Performance

For most products, the difference between product finishing time and product due date is nearly the same for these two levels. However, the finishing time of product B under random process at detailed level is later than its due date. This is caused by the delay of components B153, B122 and B111.

It can be concluded that if early due date scheduling rule at shop level is applied, more products are expected to be delayed at detailed level. Furthermore, if variabilities are defined at the machine level to consider any possible reasons causing the finishing delay of the component, the difference will be greater.

14.3.4.4 Component Performance

Most components are finished earlier than or at their due dates (Figures VIII.11 to VIII.24). In Figure VIII.19, it is found that the delay of component B153 causes the delay of components B122 and B111. Figures 14.12 and 14.13 have illustrated that B153 has been queued at machines 3, 5 and 8. At approximate level, B153 is finished about 15 days earlier than its due date, but the total waiting time at detailed analysis level \((5.5 + 6.800004 + 5.200001)\) is already greater than 15. It is found that if more shops contain a detailed machine level, a greater difference could exist. Since the delay of components happens in the company, the conclusion can be drawn that the detailed analysis level can provide more reliable and realistic results than the approximate level.

14.3.4.5 Resources Performance

Theoretically, the total machining time of each machine contained in shop 1 should be similar to the total machining time of shop 1 at approximate level. However, the calculation of processing time at each machine is the finish time minus start time, thus the waiting time at each machine has been taken into account. Therefore, the total machining
time of all machines in shop 1 at detailed level is much greater than the total machining time of shop 1 at approximate level. This can be improved by only considering the machining time at each machine.

Since it is supposed that at shop level no time is wasted by the transporter, no comparison can be made between different levels.

14.3.5 Different Strategies and Input Data at Approximate Level

14.3.5.1 Introduction

With the available information provided by GEC, only a certain range of changes may represent the real case. Therefore, this case study has not experimented with every available rule. Only those which are consistent with the company and can significantly influence the system performance have been selected.

14.3.5.2 Different Scheduling Rules

The results under four rules have been obtained: early due date, early start time, first come first served and shortest processing time (Figures 14.18 to 14.21). The material movement rule under these four runs is that the part will be released out immediately after it is completed.

Under early due date rule, all products have been finished at nearly the same time of their due dates. Under early start time, all products are finished earlier than their due date. When first come first served rule is applied, only product A is finished early than its due date, others are later. With shortest processing time rule, only product A is finished within the planning horizon.

From the above figures, it is found that outputs under two scheduling rules (early due date and early start time) have given promising results. It should be realised that if the later variability at working area is greater, information delay and shortage of raw materials happen, the early due date may cause production delay. This means that the Just-In-Time philosophy cannot always provide good performance unless all the resources are available and no breakdowns or delays occur. On the other hand, the early start time rule can lead to good performance: finishing on time. Again this does not always mean making profit to the company unless the finished products can be directly shipped to the customer instead of being put into stock. The results under shortest processing time indicate that it is not a proper scheduling rule used at shop level.
As a result, it is necessary to consider both forward scheduling and backward scheduling [99]. This depends on the general customer service performance. If delay always happens, the early start time is recommended otherwise the early due date rule is the right rule to be used in order to reduce the work in progress level.

### 14.3.5.3 Different Material Movement Rules

Two experiments have been carried out with these two rules under two different scheduling rules: early due date and early start time (Figures 14.22 to 14.25). It is found that when the early due date scheduling rule is applied, the performance under two material movement rules is nearly the same (Figures 14.22 and 14.23). This is due to the fact that all components are released to the working areas according to their due date. The finishing time of each component is nearly the same as its due date.

When the early start time rule is used, there exists a dramatic performance difference (Figures 14.24 and 14.25) between these two material movement rules. Under the first rule, all components are finished earlier than their due date, therefore, the end products are finished early as well. Under the second rule, some components are finished earlier than their due date. However, they cannot be released to the next shop until their due date. As a result, the rest of the components are finished at their due date.

When another scheduling rule: first come first served rule is used, the performance under the first material movement rule has been summarised in Figure 14.20. From this figure, the conclusion can be drawn that when the second material movement rule is applied, only product A is finished at its due date, the performance of all other products remains the same. The performance under shortest processing time is the same under these two material movement rules. As discussed in section 14.7.2, the selection of material movement rules depends on not only the general performance under each strategy, but also the consideration of other operating rules and the production performance of the company.

### 14.3.5.4 Other Rules

Other rules which can influence the system performance have been implemented in the simulation model. However, the real production situation of GEC has decided that only one operating rule could best represent the company, thus, the experiment is restricted under only one strategy, such as production type, customer order shipping rule etc. Even on experimental runs have been carried out, the results tendency under certain rules can be estimated.
14.3.5.4 Time in the Pre-Production Areas

This run is compared with the experiment summarised in Figure 14.5 with input listed in Figure 14.1. It is found that when the time range spent in the pre-production area is increased (time in bracket in Figure 14.26), there is no difference if the product due date minus start time in production is greater than its lead time in production area (Figures 14.27 to 14.28). For product D, the lead time is 65, the difference between its due date and start time in production is 64, therefore, its finish time is later than its due date.

Thus it can be concluded that time in pre-production can affect the product performance. Since the real machining time of each component cannot be changed, it is thus the key consideration to reduce the time spent in pre-production.

14.3.5.6 Different Variabilities

One experiment was planned with the different variability (Figure 14.29) used in run 1 under random process at approximate level (Figure 14.1). The results of these two runs are summarised in Figures 14.30 and 14.31.

Under the second run, it is found that product A is finished later than its due date. When the variability is increased, the real processing time is longer. Thus, components A211 and A111 are completed later than their due date. As a result, the end product is delayed. For other products, they are finished nearly at their due date. If greater variabilities are used, more products are expected to be finished later than their due dates.

As an example test of GEC, it is concluded that the model has considered most possible reasons which may cause the change of machining time.

14.3.6 Different Strategies and Input Data at Detailed Level

14.3.6.1 Introduction

This section will concentrate on the comparison of different scheduling rules applied at detailed machine level.
14.3.6.2 Scheduling Rules at Machine Level

Nine runs have been carried out for each of the scheduling rules (Figures 14.32 to 14.40). From these figures, it is found that different scheduling rules may influence machine performance. For example, when first come last served rule is used, components B124, B233 and B153 are finished much later than their due dates.

The effect can be analysed through two variables associated with machines: queue length and time in queue. Figures in 14.41 to 14.46 display queue length at machine 3 under alternative rules. It was found that when the maximum queue length was 2, the queue length change is similar. On the other hand, the machine performance is also influenced by the component flow in the system, especially the usage depends on the computational time. It is expected that when more components are released into shop 1, different scheduling rules would lead to different results. In addition, different scheduling rules applied at shop level can affect the system performance.

In conclusion, which rule can lead to the promising performance depends on several factors. Therefore, the selection of operating rules should be made with regard to all the possible factors and practical strategies applied in the company.

14.3.6.3 Inspection Rules

After products have been assembled, they are released to test shop for routine testing or special testing according to customers' specifications. Inspection can be considered as another operation like machining. Therefore, a longer time is expected if the end product is released to the test shop.

14.3.6.4 Subcontract Variability

One variable which indicates the purchasing performance is subcontract variability. In GEC, all material and components through subcontract are always available and the value is defined as 0 in the case study. However, it can be estimated that this value may delay the completion of the end products.

14.4 Concluding Remarks

The above sections report the output results from the model. The conclusion drawn from the case study will be discussed in the following chapter for limited validation.
### Key Input Information

<table>
<thead>
<tr>
<th>Key Input Information</th>
<th>10 (POSSION 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Release Interval:</td>
<td>Early start time</td>
</tr>
<tr>
<td>Shop Level Scheduling Rule:</td>
<td>First come first served</td>
</tr>
<tr>
<td>Machine Level Scheduling Rule:</td>
<td>Approximate level</td>
</tr>
<tr>
<td>Abstraction Level Selection:</td>
<td>To shipping</td>
</tr>
<tr>
<td>Product Release Selection:</td>
<td>Make-To-Order</td>
</tr>
<tr>
<td>Production Type Rule:</td>
<td>Shipping out</td>
</tr>
<tr>
<td>Product Inspection Rule:</td>
<td>No variability</td>
</tr>
<tr>
<td>Subcontract Variability Fig:</td>
<td>0</td>
</tr>
<tr>
<td>Product Safety Stock:</td>
<td>Early Later</td>
</tr>
<tr>
<td>Time in Production Area sales:</td>
<td>1 3</td>
</tr>
<tr>
<td>forecasting:</td>
<td>1 3</td>
</tr>
<tr>
<td>engineering:</td>
<td>1 5</td>
</tr>
<tr>
<td>MPP:</td>
<td>1 4</td>
</tr>
<tr>
<td>MPS:</td>
<td>1 3</td>
</tr>
</tbody>
</table>

#### Figure 14.1
**Input Information**
**Under Random Process**

### Key Input Information

<table>
<thead>
<tr>
<th>Key Input Information</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order Release Interval:</td>
<td>Early start time</td>
</tr>
<tr>
<td>Shop Level Scheduling Rule:</td>
<td>First come first served</td>
</tr>
<tr>
<td>Machine Level Scheduling Rule:</td>
<td>Approximate level</td>
</tr>
<tr>
<td>Abstraction Level Selection:</td>
<td>To shipping</td>
</tr>
<tr>
<td>Product Release Selection:</td>
<td>Make-To-Order</td>
</tr>
<tr>
<td>Production Type Rule:</td>
<td>Shipping out</td>
</tr>
<tr>
<td>Product Inspection Rule:</td>
<td>No variability</td>
</tr>
<tr>
<td>Subcontract Variability Fig:</td>
<td>0</td>
</tr>
<tr>
<td>Product Safety Stock:</td>
<td>product leadtime * 5%</td>
</tr>
<tr>
<td>Time in Production Area sales:</td>
<td>1</td>
</tr>
<tr>
<td>forecasting:</td>
<td>0</td>
</tr>
<tr>
<td>engineering:</td>
<td>product leadtime * 5%</td>
</tr>
<tr>
<td>MPP:</td>
<td>3</td>
</tr>
<tr>
<td>MPS:</td>
<td>3</td>
</tr>
</tbody>
</table>

#### Figure 14.2
**Input Information**
**Under Fixed Process**
### Total Product Input
- 16

### Total Product Output
- 6

### Total Product Required
- 6

### Total Component Input
- 68.48

### Total Component Output
- 66

### Planning Horizon
- 173

---

**Figure 14.3** Results Summary of Random Process (run 1)

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.54538</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.10048</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.6827</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.38215</td>
</tr>
</tbody>
</table>

**Performance of Working Area**

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.54538</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.10048</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.6827</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.38215</td>
</tr>
</tbody>
</table>

### Total Product Input
- 16

### Total Product Output
- 7

### Total Product Required
- 7

### Total Component Input
- 70.48

### Total Component Output
- 76

### Planning Horizon
- 174

---

**Figure 14.4** Results Summary of Random Process (run 2)

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>345.1485</td>
</tr>
<tr>
<td>Shop2</td>
<td>171.81493</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>55.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.50601</td>
</tr>
<tr>
<td>Shop6</td>
<td>145.67775</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.41103</td>
</tr>
</tbody>
</table>

**Performance of Working Area**

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>345.1485</td>
</tr>
<tr>
<td>Shop2</td>
<td>171.81493</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>55.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.50601</td>
</tr>
<tr>
<td>Shop6</td>
<td>145.67775</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.41103</td>
</tr>
</tbody>
</table>

### Total Product Input
- 18

### Total Product Output
- 8

### Total Product Required
- 8

### Total Component Input
- 81.51

### Total Component Output
- 80

### Planning Horizon
- 178

---

**Figure 14.5** Results Summary of Fixed Process (run 3)

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>375.1958</td>
</tr>
<tr>
<td>Shop2</td>
<td>191.47635</td>
</tr>
<tr>
<td>Shop3</td>
<td>80.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>70.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>90.82667</td>
</tr>
<tr>
<td>Shop6</td>
<td>180.60808</td>
</tr>
<tr>
<td>Assembly</td>
<td>80.40139</td>
</tr>
</tbody>
</table>

**Performance of Working Area**

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>375.1958</td>
</tr>
<tr>
<td>Shop2</td>
<td>191.47635</td>
</tr>
<tr>
<td>Shop3</td>
<td>80.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>70.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>90.82667</td>
</tr>
<tr>
<td>Shop6</td>
<td>180.60808</td>
</tr>
<tr>
<td>Assembly</td>
<td>80.40139</td>
</tr>
</tbody>
</table>

**Performance of Product**

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.149796</td>
<td>101</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>107.15442</td>
<td>107</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.2599</td>
<td>119</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Product D</td>
<td>153.8953</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>Product E</td>
<td>180.0</td>
<td>180</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>Product F</td>
<td>174.10732</td>
<td>174</td>
<td>7</td>
<td>74</td>
</tr>
</tbody>
</table>

**Figure 14.4**

**Results Summary of Random Process (run 2)**

**Figure 14.5**

**Results Summary of Fixed Process (run 3)**
Figure 14.6  Number of Parts in Shop3 Input Store

Figure 14.7  Number of Parts in Shop3 Working Area

Figure 14.8  Number of Parts in Shop3 Output Store
Figure 14.9  Number of Parts in Shop4 Input Store (run 1)

Figure 14.10  Number of Parts in Shop4 Input Store (run 2)

Figure 14.11  Number of Parts in Shop4 Input Store (run 3)
<table>
<thead>
<tr>
<th>Total Product Input:</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Product Output:</td>
<td>7</td>
</tr>
<tr>
<td>Total Product Required:</td>
<td>7</td>
</tr>
<tr>
<td>Total Component Input:</td>
<td>74+52</td>
</tr>
<tr>
<td>Total Component Output:</td>
<td>76</td>
</tr>
<tr>
<td>Planning Horizon:</td>
<td>153</td>
</tr>
</tbody>
</table>

**Performance of Working Area**

<table>
<thead>
<tr>
<th>Shop1</th>
<th>191.72076</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop2</td>
<td>65.6</td>
</tr>
<tr>
<td>Shop3</td>
<td>60.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>70.65784</td>
</tr>
<tr>
<td>Shop5</td>
<td>180.03407</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.20265</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>Performance Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine1</td>
<td>22.0</td>
</tr>
<tr>
<td>Machine2</td>
<td>88.0</td>
</tr>
<tr>
<td>Machine3</td>
<td>100.100006</td>
</tr>
<tr>
<td>Machine4</td>
<td>52.0</td>
</tr>
<tr>
<td>Machine5</td>
<td>78.900024</td>
</tr>
<tr>
<td>Machine6</td>
<td>32.0</td>
</tr>
<tr>
<td>Machine7</td>
<td>14.0</td>
</tr>
<tr>
<td>Machine8</td>
<td>61.899994</td>
</tr>
</tbody>
</table>

**Results Summary of Random Process (run 1)**

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Actual Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.04873</td>
<td>101</td>
<td>1.5</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>106.640076</td>
<td>107</td>
<td>10.0</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.14978</td>
<td>119</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Product D</td>
<td>115.0173</td>
<td>118</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Product E</td>
<td>150.11148</td>
<td>150</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product F</td>
<td>146.0</td>
<td>146</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product G</td>
<td>152.1529</td>
<td>152</td>
<td>6</td>
<td>52</td>
</tr>
</tbody>
</table>

**Performance of Transporter**

<table>
<thead>
<tr>
<th>Transhiper in Shop</th>
<th>60.399996</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance of Component in Shop</td>
<td>Waiting Time</td>
</tr>
<tr>
<td>$S_1$</td>
<td>21.2001</td>
</tr>
</tbody>
</table>

**Results Summary of Random Process (run 2)**

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Actual Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.05483</td>
<td>101</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>130.640076</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.14978</td>
<td>119</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Product D</td>
<td>115.0173</td>
<td>118</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Product E</td>
<td>150.11148</td>
<td>150</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product F</td>
<td>146.0</td>
<td>146</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product G</td>
<td>152.1529</td>
<td>152</td>
<td>6</td>
<td>52</td>
</tr>
</tbody>
</table>

**Performance of Transporter**

<table>
<thead>
<tr>
<th>Transhiper in Shop</th>
<th>60.399996</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance of Component in Shop</td>
<td>Waiting Time</td>
</tr>
<tr>
<td>$S_1$</td>
<td>21.2001</td>
</tr>
</tbody>
</table>

**Results Summary of Certain Process (run 3)**

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Actual Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.04873</td>
<td>101</td>
<td>1.5</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>106.640076</td>
<td>107</td>
<td>10.0</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.14978</td>
<td>119</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Product D</td>
<td>115.0173</td>
<td>118</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Product E</td>
<td>150.11148</td>
<td>150</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product F</td>
<td>146.0</td>
<td>146</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>Product G</td>
<td>151.05717</td>
<td>151</td>
<td>12</td>
<td>41</td>
</tr>
</tbody>
</table>

**Performance of Transporter**

<table>
<thead>
<tr>
<th>Transhiper in Shop</th>
<th>41.399996</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance of Component in Shop</td>
<td>Waiting Time</td>
</tr>
<tr>
<td>$S_1$</td>
<td>102.3</td>
</tr>
</tbody>
</table>
Figure 14.15  
Queue Length at Machine3 (run2)

Figure 14.16  
Time at Queue at Machine3 (run 2)
<table>
<thead>
<tr>
<th>Product</th>
<th>Level</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Finish Time</td>
<td>Due Date</td>
<td>Finish Time</td>
</tr>
<tr>
<td>A</td>
<td>Approximate</td>
<td>101.1491</td>
<td>101</td>
<td>101.14976</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>101.09493</td>
<td>101</td>
<td>101.09493</td>
</tr>
<tr>
<td>B</td>
<td>Approximate</td>
<td>107.15442</td>
<td>107</td>
<td>107.15442</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>109.640076</td>
<td>107</td>
<td>109.640076</td>
</tr>
<tr>
<td>C</td>
<td>Approximate</td>
<td>119.95297</td>
<td>119</td>
<td>119.26103</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>119.14917</td>
<td>119</td>
<td>119.14947</td>
</tr>
<tr>
<td>D</td>
<td>Approximate</td>
<td>118.05757</td>
<td>118</td>
<td>118.05748</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>118.05173</td>
<td>118</td>
<td>118.05173</td>
</tr>
<tr>
<td>E</td>
<td>Approximate</td>
<td>153.06063</td>
<td>153</td>
<td>153.06063</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>150.11148</td>
<td>150</td>
<td>150.11148</td>
</tr>
<tr>
<td>F</td>
<td>Approximate</td>
<td>150.0</td>
<td>150</td>
<td>150.0</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>140.0</td>
<td>140</td>
<td>140.0</td>
</tr>
<tr>
<td>G</td>
<td>Approximate</td>
<td>152.1529</td>
<td>152</td>
<td>152.1529</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>152.1529</td>
<td>152</td>
<td>152.1529</td>
</tr>
</tbody>
</table>

Figure 14.17

Comparison of Different Levels
| Total Product Input: | 17 |
| Total Product Output: | 7 |
| Total Product Required: | 7 |
| Total Component Input: | 65+45 |
| Total Component Output: | 76 |
| Planning Horizon: | 173 |

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>324.42554</td>
</tr>
<tr>
<td>Shop2</td>
<td>126.17635</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>60.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.7161</td>
</tr>
<tr>
<td>Shop6</td>
<td>115.52065</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.29249</td>
</tr>
</tbody>
</table>

Performance of Product

<table>
<thead>
<tr>
<th>Finish</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.46453</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>Product B</td>
<td>107.41706</td>
<td>107</td>
<td>10</td>
</tr>
<tr>
<td>Product C</td>
<td>118.49382</td>
<td>119</td>
<td>7</td>
</tr>
<tr>
<td>Product D</td>
<td>118.31611</td>
<td>118</td>
<td>10</td>
</tr>
<tr>
<td>Product E</td>
<td>150.19254</td>
<td>150</td>
<td>8</td>
</tr>
<tr>
<td>Product F</td>
<td>151.08345</td>
<td>151</td>
<td>8</td>
</tr>
<tr>
<td>Product G</td>
<td>165.97223</td>
<td>165</td>
<td>8</td>
</tr>
</tbody>
</table>

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>248.96384</td>
</tr>
<tr>
<td>Shop2</td>
<td>126.35426</td>
</tr>
<tr>
<td>Shop3</td>
<td>45.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>45.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>55.62358</td>
</tr>
<tr>
<td>Shop6</td>
<td>100.373856</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.31374</td>
</tr>
</tbody>
</table>

Performance of Product

<table>
<thead>
<tr>
<th>Finish</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>88.15665</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>Product B</td>
<td>109.51813</td>
<td>107</td>
<td>10</td>
</tr>
<tr>
<td>Product C</td>
<td>127.26381</td>
<td>123</td>
<td>8</td>
</tr>
<tr>
<td>Product D</td>
<td>148.13045</td>
<td>122</td>
<td>7</td>
</tr>
<tr>
<td>Product E</td>
<td>151.16173</td>
<td>154</td>
<td>7</td>
</tr>
<tr>
<td>Product F</td>
<td>200.09206</td>
<td>152</td>
<td>7</td>
</tr>
<tr>
<td>Product G</td>
<td>213.05516</td>
<td>159</td>
<td>5</td>
</tr>
</tbody>
</table>

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>139.22122</td>
</tr>
<tr>
<td>Shop2</td>
<td>87.04245</td>
</tr>
<tr>
<td>Shop3</td>
<td>20.19045</td>
</tr>
<tr>
<td>Shop4</td>
<td>20.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>15.0</td>
</tr>
<tr>
<td>Shop6</td>
<td>20.20401</td>
</tr>
<tr>
<td>Assembly</td>
<td>15.063965</td>
</tr>
</tbody>
</table>

Performance of Product

<table>
<thead>
<tr>
<th>Finish</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>139.22122</td>
<td>101</td>
<td>11.5</td>
</tr>
</tbody>
</table>

Figure 14.18 Results Summary Under Early Due Date Rule

Figure 14.19 Results Summary Under Early Start Time Rule

Figure 14.20 Results Summary Under First Come First Served Rule

Figure 14.21 Results Summary Under Shortest Processing Time Rule
Total Product Input: 17
Total Product Output: 7
Total Product Required: 7
Total Component Input: 65
Total Component Output: 76
Planning Horizon: 173

Performance of Working Area
<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop 1</td>
<td>324.42554</td>
</tr>
<tr>
<td>Shop 2</td>
<td>126.17635</td>
</tr>
<tr>
<td>Shop 3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop 4</td>
<td>60.0</td>
</tr>
<tr>
<td>Shop 5</td>
<td>70.7161</td>
</tr>
<tr>
<td>Shop 6</td>
<td>115.520065</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.29249</td>
</tr>
</tbody>
</table>

Performance of Product
<table>
<thead>
<tr>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.48453</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>Product B</td>
<td>107.41706</td>
<td>107</td>
<td>10</td>
</tr>
<tr>
<td>Product C</td>
<td>119.49832</td>
<td>119</td>
<td>7</td>
</tr>
<tr>
<td>Product D</td>
<td>118.31611</td>
<td>118</td>
<td>10</td>
</tr>
<tr>
<td>Product E</td>
<td>150.15254</td>
<td>150</td>
<td>8</td>
</tr>
<tr>
<td>Product F</td>
<td>151.08345</td>
<td>151</td>
<td>9</td>
</tr>
<tr>
<td>Product G</td>
<td>155.97223</td>
<td>155</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 14.22 Results Summary (rule 1)
Under Early Due Date Rule

Total Product Input: 14
Total Product Output: 8
Total Product Required: 7
Total Component Input: 74.44
Total Component Output: 90
Planning Horizon: 173

Performance of Working Area
<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop 1</td>
<td>345.1909</td>
</tr>
<tr>
<td>Shop 2</td>
<td>146.25798</td>
</tr>
<tr>
<td>Shop 3</td>
<td>60.0</td>
</tr>
<tr>
<td>Shop 4</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop 5</td>
<td>65.17221</td>
</tr>
<tr>
<td>Shop 6</td>
<td>140.54813</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.432655</td>
</tr>
</tbody>
</table>

Performance of Product
<table>
<thead>
<tr>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>85.31573</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>Product B</td>
<td>97.09749</td>
<td>107</td>
<td>10</td>
</tr>
<tr>
<td>Product C</td>
<td>101.52122</td>
<td>119</td>
<td>7</td>
</tr>
<tr>
<td>Product D</td>
<td>107.20537</td>
<td>118</td>
<td>9</td>
</tr>
<tr>
<td>Product E</td>
<td>130.1294</td>
<td>153</td>
<td>7</td>
</tr>
<tr>
<td>Product F</td>
<td>141.06334</td>
<td>160</td>
<td>6</td>
</tr>
<tr>
<td>Product G</td>
<td>155.23979</td>
<td>176</td>
<td>9</td>
</tr>
<tr>
<td>Product H</td>
<td>167.34406</td>
<td>165</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 14.23 Results Summary (rule 2)
Under Early Due Start Time Rule
Key Input Information

Order Release Interval: 10 (POSSION 10)
Shop Level Scheduling Rule: Early start time
Machine Level Scheduling Rule: First come first served
Abstraction Level Selection: Approximate level
Product Release Selection: To shipping
Production Type Rule: Make-To-Order
Product Inspection Rule: Shipping out
Subcontract Variability Fig: No variability
Product Safety Stock: 0
Time in Production Area

<table>
<thead>
<tr>
<th>sales:</th>
<th>Early</th>
<th>Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>forecasting:</td>
<td>1 (3)</td>
<td>3 (6)</td>
</tr>
<tr>
<td>engineering:</td>
<td>1 (3)</td>
<td>3 (6)</td>
</tr>
<tr>
<td>MPP:</td>
<td>1 (5)</td>
<td>5 (10)</td>
</tr>
<tr>
<td>MPS:</td>
<td>1 (4)</td>
<td>4 (8)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Release Selection: To shipping</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Release</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Arrivals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101.1491</td>
<td>101</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>107.15442</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119.32599</td>
<td>119</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>D</td>
<td>118.0634</td>
<td>118</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>E</td>
<td>153.06053</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>F</td>
<td>160.0</td>
<td>160</td>
<td>6</td>
<td>60</td>
</tr>
</tbody>
</table>

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.54358</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.10048</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.68827</td>
</tr>
<tr>
<td>Shop6</td>
<td>145.62129</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.38215</td>
</tr>
</tbody>
</table>

Performance of Product

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101.1491</td>
<td>101</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>107.15442</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119.32599</td>
<td>119</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>D</td>
<td>118.0634</td>
<td>118</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>E</td>
<td>153.06053</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>F</td>
<td>160.0</td>
<td>160</td>
<td>6</td>
<td>60</td>
</tr>
</tbody>
</table>

Figure 14.26

Input Information

Under Random Process

Total Product Input: 16
Total Product Output: 6
Total Product Required: 6
Total Component Input: 68+48
Total Component Output: 66
Planning Horizon: 173

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.54358</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.10048</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.68827</td>
</tr>
<tr>
<td>Shop6</td>
<td>145.62129</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.38215</td>
</tr>
</tbody>
</table>

Total Component Output: 66 Total Component Output: 76 Planning Horizon: 157

Performance of Product

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101.1491</td>
<td>101</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>107.15442</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119.32599</td>
<td>119</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>D</td>
<td>118.0634</td>
<td>118</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>E</td>
<td>153.06053</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>F</td>
<td>160.0</td>
<td>160</td>
<td>6</td>
<td>60</td>
</tr>
</tbody>
</table>

Figure 14.27

Results Summary of Random Process (run 1)

Total Product Input: 19
Total Product Output: 7
Total Product Required: 7
Total Component Input: 72+48
Total Component Output: 76 Planning Horizon: 157

Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.74344</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.59398</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.7378</td>
</tr>
<tr>
<td>Shop6</td>
<td>150.83722</td>
</tr>
<tr>
<td>Assembly</td>
<td>70.28879</td>
</tr>
</tbody>
</table>

Performance of Product

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Production</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101.21127</td>
<td>101</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>107.15442</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>120.31091</td>
<td>120</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>D</td>
<td>123.38743</td>
<td>123</td>
<td>21</td>
<td>37</td>
</tr>
<tr>
<td>E</td>
<td>154.16469</td>
<td>154</td>
<td>19</td>
<td>44</td>
</tr>
<tr>
<td>F</td>
<td>150.0</td>
<td>150</td>
<td>19</td>
<td>50</td>
</tr>
<tr>
<td>G</td>
<td>155.07954</td>
<td>155</td>
<td>20</td>
<td>55</td>
</tr>
</tbody>
</table>

Figure 14.28

Results Summary of Random Process
### Key Input Information

<table>
<thead>
<tr>
<th>Order Release Interval:</th>
<th>10 (POSSION 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early start time</td>
<td></td>
</tr>
<tr>
<td>First come first served</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine Level Scheduling Rule:</th>
<th>Early Later</th>
</tr>
</thead>
<tbody>
<tr>
<td>To shipping</td>
<td>1</td>
</tr>
<tr>
<td>Make-To-Order</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstraction Level Selection:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Release Selection:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Inspection Rule:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subcontract Variability Flag:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Safety Stock:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time in Production Area:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sales:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>forecasting:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>engineering:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPP:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPS:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Information Delay Variability:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working Area Early Variability:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working Area Later Variability:</th>
<th>Estimated level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Results Summary of Random Process (run 1)

<table>
<thead>
<tr>
<th>Total Product Input:</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Product Output:</td>
<td>6</td>
</tr>
<tr>
<td>Total Product Required:</td>
<td>6</td>
</tr>
<tr>
<td>Total Component Input:</td>
<td>5648</td>
</tr>
<tr>
<td>Total Component Output:</td>
<td>65</td>
</tr>
<tr>
<td>Planning Horizon:</td>
<td>173</td>
</tr>
</tbody>
</table>

#### Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>344.5658</td>
</tr>
<tr>
<td>Shop2</td>
<td>172.10048</td>
</tr>
<tr>
<td>Shop3</td>
<td>65.0</td>
</tr>
<tr>
<td>Shop4</td>
<td>50.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>70.68287</td>
</tr>
<tr>
<td>Shop6</td>
<td>145.62129</td>
</tr>
<tr>
<td>Assembly</td>
<td>60.38215</td>
</tr>
</tbody>
</table>

#### Performance of Product

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.1491</td>
<td>101</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>197.15442</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.32599</td>
<td>119</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>Product D</td>
<td>118.9654</td>
<td>118</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>Product E</td>
<td>159.06553</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>Product F</td>
<td>190.0</td>
<td>160</td>
<td>6</td>
<td>60</td>
</tr>
</tbody>
</table>

---

### Results Summary of Random Process (run 2)

<table>
<thead>
<tr>
<th>Total Product Input:</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Product Output:</td>
<td>7</td>
</tr>
<tr>
<td>Total Product Required:</td>
<td>7</td>
</tr>
<tr>
<td>Total Component Input:</td>
<td>7047</td>
</tr>
<tr>
<td>Total Component Output:</td>
<td>76</td>
</tr>
<tr>
<td>Planning Horizon:</td>
<td>175</td>
</tr>
</tbody>
</table>

#### Performance of Working Area

<table>
<thead>
<tr>
<th>Working Area</th>
<th>Total Working Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shop1</td>
<td>356.87903</td>
</tr>
<tr>
<td>Shop2</td>
<td>176.90845</td>
</tr>
<tr>
<td>Shop3</td>
<td>68.94397</td>
</tr>
<tr>
<td>Shop4</td>
<td>55.0</td>
</tr>
<tr>
<td>Shop5</td>
<td>72.23933</td>
</tr>
<tr>
<td>Shop6</td>
<td>147.28908</td>
</tr>
<tr>
<td>Assembly</td>
<td>72.00867</td>
</tr>
</tbody>
</table>

#### Performance of Product

<table>
<thead>
<tr>
<th>Product</th>
<th>Finish Time</th>
<th>Due Date</th>
<th>Time in Pre Production</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>102.25149</td>
<td>101</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Product B</td>
<td>107.99725</td>
<td>107</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Product C</td>
<td>119.86538</td>
<td>119</td>
<td>7</td>
<td>19</td>
</tr>
<tr>
<td>Product D</td>
<td>118.407265</td>
<td>118</td>
<td>9</td>
<td>33</td>
</tr>
<tr>
<td>Product E</td>
<td>153.52214</td>
<td>153</td>
<td>7</td>
<td>43</td>
</tr>
<tr>
<td>Product F</td>
<td>160.0</td>
<td>160</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>Product G</td>
<td>174.43585</td>
<td>174</td>
<td>5</td>
<td>74</td>
</tr>
</tbody>
</table>
### Results Summary of Random Process – FCFS Rule

<table>
<thead>
<tr>
<th>Product</th>
<th>Production</th>
<th>Total Comp. Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>G</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Due Time</th>
<th>Production Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>136</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119</td>
<td>9</td>
</tr>
<tr>
<td>D</td>
<td>118</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>151</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>146</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>152</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Shop in</th>
<th>60.399998</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Comp. Input</td>
<td>91.200001</td>
</tr>
</tbody>
</table>

### Results Summary of Random Process – FCLS Rule

<table>
<thead>
<tr>
<th>Product</th>
<th>Production</th>
<th>Total Comp. Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>G</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Due Time</th>
<th>Production Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>136</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119</td>
<td>9</td>
</tr>
<tr>
<td>D</td>
<td>118</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>151</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>146</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>152</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Shop in</th>
<th>57.399998</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Comp. Input</td>
<td>94.600008</td>
</tr>
</tbody>
</table>

### Results Summary of Random Process – SPT Rule

<table>
<thead>
<tr>
<th>Product</th>
<th>Production</th>
<th>Total Comp. Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>D</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>G</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Product</th>
<th>Due Time</th>
<th>Production Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>101</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>136</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>119</td>
<td>9</td>
</tr>
<tr>
<td>D</td>
<td>118</td>
<td>10</td>
</tr>
<tr>
<td>E</td>
<td>151</td>
<td>9</td>
</tr>
<tr>
<td>F</td>
<td>146</td>
<td>9</td>
</tr>
<tr>
<td>G</td>
<td>152</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance of Shop in</th>
<th>57.399998</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Comp. Input</td>
<td>94.600008</td>
</tr>
</tbody>
</table>

---

### Footnotes

- *Figures 14.32–14.34* Results Summary of Random Process, showing performance under different scheduling rules: FCFS, FCLS, and SPT, respectively.
### Figure 14.35
Results Summary of Random Process -- LPT Rule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Figure 14.36
Results Summary of Random Process -- STPT Rule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

### Figure 14.37
Results Summary of Random Process -- LTPT Rule

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Figure 14.38
Results Summary of Random Process — EDD Rule

<table>
<thead>
<tr>
<th>Total Product Input</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Product Output:</td>
<td>7</td>
</tr>
<tr>
<td>Total Product Required:</td>
<td>7</td>
</tr>
<tr>
<td>Total Component Input:</td>
<td>74.52</td>
</tr>
<tr>
<td>Total Component Output:</td>
<td>76</td>
</tr>
<tr>
<td>Planning Horizon:</td>
<td>153</td>
</tr>
</tbody>
</table>

Performance of Machine

<table>
<thead>
<tr>
<th>Machine</th>
<th>22.0</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine2</td>
<td>88.0</td>
<td>18.0</td>
</tr>
<tr>
<td>Machine3</td>
<td>106.100006</td>
<td>50.100002</td>
</tr>
<tr>
<td>Machine4</td>
<td>63.0</td>
<td>33.0</td>
</tr>
<tr>
<td>Machine5</td>
<td>32.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Machine6</td>
<td>14.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Machine7</td>
<td>98.800004</td>
<td>14.800004</td>
</tr>
</tbody>
</table>

Performance of Transporter

<table>
<thead>
<tr>
<th>Transporter in Shop</th>
<th>S7.598998</th>
<th>$1.200091</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.09493</td>
<td>B153 9</td>
</tr>
<tr>
<td>Product B</td>
<td>109.640076</td>
<td>107</td>
</tr>
<tr>
<td>Product C</td>
<td>119.14918</td>
<td>119</td>
</tr>
<tr>
<td>Product D</td>
<td>119.14918</td>
<td>119</td>
</tr>
<tr>
<td>Product E</td>
<td>150.11148</td>
<td>150</td>
</tr>
<tr>
<td>Product F</td>
<td>150.11148</td>
<td>150</td>
</tr>
<tr>
<td>Product G</td>
<td>152.1529</td>
<td>152</td>
</tr>
</tbody>
</table>

Figure 14.39
Results Summary of Random Process — LS Rule

<table>
<thead>
<tr>
<th>Total Product Input</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Product Output:</td>
<td>7</td>
</tr>
<tr>
<td>Total Product Required:</td>
<td>7</td>
</tr>
<tr>
<td>Total Component Input:</td>
<td>74.52</td>
</tr>
<tr>
<td>Total Component Output:</td>
<td>76</td>
</tr>
<tr>
<td>Planning Horizon:</td>
<td>153</td>
</tr>
</tbody>
</table>

Performance of Machine

<table>
<thead>
<tr>
<th>Machine</th>
<th>22.0</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine2</td>
<td>88.0</td>
<td>18.0</td>
</tr>
<tr>
<td>Machine3</td>
<td>106.100006</td>
<td>50.100002</td>
</tr>
<tr>
<td>Machine4</td>
<td>63.0</td>
<td>33.0</td>
</tr>
<tr>
<td>Machine5</td>
<td>32.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Machine6</td>
<td>14.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Machine7</td>
<td>98.800004</td>
<td>14.800004</td>
</tr>
</tbody>
</table>

Performance of Transporter

<table>
<thead>
<tr>
<th>Transporter in Shop</th>
<th>S7.598998</th>
<th>$1.200091</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product A</td>
<td>101.09493</td>
<td>B153 9</td>
</tr>
<tr>
<td>Product B</td>
<td>109.640076</td>
<td>107</td>
</tr>
<tr>
<td>Product C</td>
<td>119.14918</td>
<td>119</td>
</tr>
<tr>
<td>Product D</td>
<td>119.14918</td>
<td>119</td>
</tr>
<tr>
<td>Product E</td>
<td>150.11148</td>
<td>150</td>
</tr>
<tr>
<td>Product F</td>
<td>150.11148</td>
<td>150</td>
</tr>
<tr>
<td>Product G</td>
<td>152.1529</td>
<td>152</td>
</tr>
</tbody>
</table>

Figure 14.40
Results Summary of Random Process — SSR Rule
Figure 14.41  Queue Length at Machine3 -- FCLS Rule

Figure 14.42  Queue Length at Machine3 -- LPT Rule

Figure 14.43  Queue Length at Machine3 -- STPT Rule
Figure 14.44  
Queue Length at Machine 3 -- EDD Rule

Figure 14.45  
Queue Length at Machine 3 -- LS Rule

Figure 14.46  
Queue Length at Machine 3 -- SSR Rule
Chapter 15
CONCLUDING DISCUSSION

15.1 Introduction

This chapter discusses the contributions and limitations of the integrated knowledge based simulation model based on the case study. The purpose of this chapter is to bring together major issues to formulate conclusions.

15.2 The Efficient Integrated Methodology

The literature surveyed in Chapter 2 has illustrated the potential trend towards the provision of powerful modelling tools for manufacturing systems design [177]. The study of manufacturing organizations in Chapter 3 has highlighted their hierarchical and complex structure and indicated the way they are organised. Chapter 4 has identified the key issues with regard to the efficient operation of a company. Recognition and discussion on contributions and limitations of currently available modelling tools and the requirement for the design of large scale manufacturing organizations lead to the need for a new modelling method. This provides the impetus and foundation for research into the integrated knowledge based hierarchical modelling of manufacturing organizations, Chapter 5.

Supported by the widely recognized IDEF0 modeling technique and STEM simulation package, an integrated methodology has been established to provide a novel approach for modelling manufacturing organizations, Chapter 6. The major aspects of the integrated modelling system include the IDEF0 representation of manufacturing organizations, hierarchical modelling approach, IDEF0-like configuration of the simulation model, specialised data base, specialised knowledge base and specialised user interface. Different from single level modelling, hierarchical modelling allows modelling of large and complex systems. The simulation model is organised in just the same way as it is structured, thus providing a realistic model.

15.3 Static Hierarchical Modelling of Manufacturing Organizations

The IDEF0 system analysis method has been employed in the research to provide analysis of static performance by inserting the functional activities of a manufacturing organization into a hierarchical model. The work reported in Chapter 7 has demonstrated
that the IDEF0 representation of manufacturing organizations provides an useful structure based on which the simulation model is built. It enhances simulation system which only offer dynamic modelling and provides a potential interface for the simulation model. However, a more detailed IDEF0 model should be considered to provide more information and enhance the model capability.

15.4 Hierarchical Modelling Approach

Based on the requirements from industries and limitations of single level modelling methods, The study in Chapter 8 has established an original methodology for hierarchical modelling. This methodology reveals general procedures used in the development of the simulation model and identifies key issues at each stage. It also illustrates the decisions made at different stages and provides the foundation for building the simulation model.

15.5 Simulation System Structure

The hierarchical representation of the simulation model discussed in section 9.3 in Chapter 9 has illustrated a novel approach to present the physical structure of the model -- IDEF0-like representation. This overcomes the major problems of many simulation tools that either they cannot provide a physical configuration or they are restricted in the limited types of manufacturing resources [175] [274] [181]. Additionally it provides the user with a clearly defined structure of the system and the capability to be easily understood, but it requires further development in order to realistically model the system by involving more activities (refer to Chapter 17).

15.6 Simulation System Operation

The knowledge based simulation system has been structured like other AI modelling systems [274] using an integrated knowledge representation scheme within a discrete event simulation package. The major elements in this system include data base, knowledge base and inference engine, section 9.3 in Chapter 9. Based on the STEM, the concentration has been placed on the specialisation of knowledge base and data base which define different application domain. This hierarchical organization of objects overcomes the limitation of conventional languages [62] [175] and provide easy data management system.
15.7 Modelling Knowledge of the Simulation Model

The significant partition in the simulation model lies in the knowledge base of the system, which contains the modelling knowledge to decide the unique characteristics of the system. It uses rules and procedures to describe the simulated objects in manufacturing organizations [216]. Since a simulation object differs from its real world counterpart in detail but not in form [216], the conversion from the real system to a computer program requires a certain degree of assumptions. The first requirement for clear representation of modelling knowledge in the simulation model is to state the assumptions made in the model (refer to Chapter 10).

Although major functional areas have been included in the model, more interaction between them needs to be considered to offer the user more realistic insight into the system.

15.8 Model with Different Levels

According to the design sequence and the decision making level, the simulation model has been designed to contain two different levels for different purposes, section 10.6 in Chapter 10. The distinction of these two levels highlights the different usage of the model and provides the capability for efficient and effective application. Although two different levels have been involved in the model, more knowledge need to be identified at the detailed analysis level to consider more decision making and thus lead to the creation of a more powerful model.

15.9 Data and Rule Driven Environment

One major problem which can be found in many simulation tools is the level of user involvement with the model. One tendency is that the user can only enter data but he cannot change decision making contained in the model [175]. This integrated simulation model allows the user to enter both data and rules to enhance its applicability.

According to the effects of decision making, those decisions which can greatly influence the system performance and present different possible ways can be selected by the user. The study in Chapter 10 has identified those rules and outlined the scope of input data. The need to allow the user to enter new rules has been identified as a further requirement.
15.10 User Interface

The user interface provided by the simulation model has made it easier to specialise different systems. Control strategies, scheduling rules and required data can be menu-selected from an integral library, or individually specified. Thus the user is able to progressively develop a structured description of the system into a simulation model (refer to Chapter 11). Furthermore, it enhances the STEM environment and provides flexibility to allow the user to obtain different results in which the user is interested.

15.11 Model Output

The study of output interpretation is a major research area of its own. The key issue is to provide flexibility with configurable and meaningful outputs to different users. Chapter 12 identifies two categories of outputs based on the way they are obtained, manufacturing-oriented outputs and standardised outputs. These outputs help the user understand how the system is performing when certain rules and data are applied.

15.12 Computational Environment

The computing environment provided by the SUN and Xerox work stations has made it possible to building a specialised manufacturing organization simulation model. However, experience has shown that the computational environment has restricted the development of more complex models because of limited computer storage and memory capacity. This to some extent has limited the creation of a more realistic simulation model. The potential solution is to extend the computing capability.

15.13 Case Study Experience

Based on the results discussion in Chapter 14, this section carries out the limited validation with regard to the scope of the case study established in section 13.2.

15.13.1 Aggregated Manufacturing Organizations Model

GEC ALSTHOM Large Machines Ltd. has been effectively modelled using the simulation model. The overall results illustrate that the model covers major areas which are
found the key departments in the company. However, some limitations have been identified: more areas should be included such as estimating and stock control departments and realistic relationship between them should be established.

15.13.2 Adaptability

The adaptability of the model was successfully demonstrated. As a test of make-to-order and one-off type company, the GEC case study has proved that the model contains alternative options which can cope with different types of production and operating strategies. However, more rules should be added to be able to model batch production, such as batch size decision and effects of setting-up etc.

15.13.3 Approximate Modelling and Detailed Modelling

The results (sections 14.3) demonstrate the flexibility of the model. They illustrate that the approximate level is most suitable for design study at the early stage determining working areas capacity and work in progress level. It can also help to quickly analyse the system’s overall performance. At the detailed analysis level, the general cell, workstation and machine performance can be analysed and the component flow at machine level can be effectively assessed. The results show that the detailed analysis level produces nearly the same results as the approximate level, which implies that further improvement is required in order to enhance the flexibility.

15.13.4 Easy Data Base Management

The case study experience has shown that the management of the data base in the model is flexible and efficient. Since not all of the decision rules and data can be entered through the user interface, some have to be modified in the data base. The representation of each object in the class definition helps locate the corresponding object easily and the initialised instance variables can be changed. Furthermore, the addition and deleting of object instances can also be made without affecting the rest of the model.

15.13.5 Influence of Assumptions

The discussion in section 14.3 has demonstrated that the assumptions made in the model can greatly influence the accuracy of the results. The more assumption the less
accurate of the results. Therefore the output from detailed analysis level is more reliable than that of the approximate level. One solution to this is to add more rules to reduce the incorrect level of results.

15.13.6 Influence of Different Rules

The case study has illustrated that different rules can influence greatly the system performance, for example, the scheduling rules at shop level and machine level can affect the performance of components, especially finishing time. The order shipping rule may change the flow of production orders. However, the results show that the effect of their combination matters a lot. Therefore, careful considerations have to be made before selecting them. Usually the decision depends on the effect of each rule and the real environment.

15.13.7 Influence of Initial Running Condition

The case study has shown that the initial running condition cannot influence the results of output but the confidence level of output provided. The longer the computing time, the higher level the output provides. Thus the longer running time is recommended if it is possible. The computing capacity in this project has indicated that further improvement is required (refer to Chapter 17).

15.13.8 Accuracy of the Results

The output results have been proved to be accurate to represent a real manufacturing system in the sense that the running results closely correspond to the real system. Some limitations have been identified due to: i) assumptions; ii) initial running conditions; iii) data unavailable from the company and iv) the random process. This can be improved by putting more rules, running longer time and using correct input data and find the proper random process to best represent the real case.

15.13.9 Potential Improvements

From the case study, some gaps have been identified between the real company and the simulation model. Therefore, the modelling system should be extended and enhanced to improve the capability of the system. Furthermore, the improvements on the simulation environment have been identified, refer to Chapter 17.
16.1 Introduction

Chapter 15 has drawn some conclusions on the major features of the modelling system based on the case study. However, it should be emphasised that it is also important to identify limitations of the research in order to justify the research work, indicate the requirements for the further development and point out the potential direction of the research. Thus the purpose of this chapter is to critically assess the project based on major criteria such as original objectives, generic and specialised nature of the model, justification of hierarchical modelling, linkage between IDEF0 and STEM, improvement of the user interface and practical use of the modelling tool in respect of ability to handle realistic problems and company wide models.

16.2 Research Objectives

This section examines the original objectives established in comparison with the major contributions of the research. It also states some impacts of the project on other research work which were not originally identified when this work started. It finally identifies the remaining problems in achieving the original research objectives and discusses how to improve them.

This project aims to research into an integrated methodology consisting of static system description (Chapter 7) [102] [141] and dynamic simulation (Chapter 9) [239] [264] for modelling manufacturing organizations based on the literature survey and discussion in Chapter 5. The focus of the research is on the integration of the use of IDEF0 methodology into the building of a simulation model. The major outcome from the research is a top-down approach to modelling manufacturing organizations and as a by-product a generic hierarchical simulation model has been built for tackling certain problems identified (refer to Chapters 8 and 13) [234] [109].

This research has achieved some notable results. It has derived the following:

--why an integrated methodology is required (Chapter 5)
--the real purpose of an integrated methodology (Chapter 5)
--major requirements of an integrated methodology (Chapter 5)
--major requirements of a system description method (Chapter 5)
--major requirements of the dynamic simulation method (Chapter 5)
--selection of a system description (Chapter 5)
--initial study of the selection of a dynamic simulation method (Chapter 5)
--integration of IDEF0 and STEM (refer to sections 16.3 and 16.5).

The achievement in building the generic hierarchical model of the manufacturing organization has been accomplished by applying the integrated methodology in the context of company operation. This company wide model covers four organizational layers of the reference model proposed by the ISO [112] (refer to Chapter 9). The model can be run under different approximation levels for quick estimation and detailed analysis [274] (refer to Chapter 10). A user-friendly interface has been developed for the model to allow both data and rule entry, assisting the user in inputting to and collecting output from the model (refer to Chapter 11). In addition any node or CNode defined in the model can be re-used in other models.

The manufacturing organization model developed in this research contains domains of decision and information as well as physical domains (refer to Chapter 10) which have been identified as key substructures in planning and implementing CIM systems but are usually studied separately [8] [66] [73]. A strength of the model is that it may be used as a test bed for evaluating and testing such CIM systems [8] [66] [217]. This test bed aspect is an area which has not been fully explored at this time.

Furthermore, the work in this thesis has served as the preliminary study for a new research project funded by SERC. The overall aim of this new project is to develop an aid for the design and analysis of complex and hierarchical manufacturing organizations. The following areas will be explored in this new research:
--formal description of hierarchical systems
--dynamic modelling of activities within the system description
--study of the problems of approximate modelling
--expert interpretation of modelling results.

The above has been based on the major contributions of the research reported in this thesis (refer to Chapters 14 and 15 and section 16.7).

Potential improvement needed to the integrated methodology and the hierarchical simulation model. During the research, some limitations were identified in both the integrated methodology and the hierarchical simulation model. For the integrated methodology, two major issues are the integration of the static system description and dynamic simulation method in one single computing environment and the need for information modelling.
Integration of the system description and simulation method. The key issue in the research is the integration of the use of IDEF0 methodology into model building. The work from this integrated methodology is not completed in the sense that the simulation model is built within the scope bounded by IDEF0 representation, but the transition from IDEF0 to the simulation model is indirect (refer to section 16.4). The Figures 7.1 to 7.8 and 9.2 to 9.15 illustrate this. Since the simulation model is not represented in the same way as used in IDEF0, this may decrease the comprehensive capability which is one of the objectives of the integrated methodology (refer to Chapter 5).

Lack of information modelling. Another problem which has been identified is the lack of information modelling between functional modelling and dynamic modelling [251] [73]. Although a functional representation of the system has provided the user a clear structure of the system and design aid for the modeller to build the simulation model (refer to Chapter 8), the information required for the dynamic modelling and data flow between different functional areas are not identified and are only known by the modeller. Many researchers conclude that these three modelling aspects contribute to the complete study of a system, such as IDEF0, IDEF1 and IDEF2 developed for this purpose [18] [208]. One possible solution is to use data flow diagrams to complement IDEF0 diagrams by showing the information flow within the system [18] [208].

Software constraints. In respect of the hierarchical simulation model, many issues should be discussed but this section deals with the constraints of the simulation package STEM. The reason why STEM was originally selected is stated in section 16.5. The nature of the model is discussed in section 16.3, fixed configuration of the model is tackled in section 16.5 and problem solving capability of the model is discussed in section 16.7.

One limitation related to the simulation tool is the constraints of the STEM and LISP environment. In STEM, one important advantage is that it provides a facility to hierarchically build the model, especially the repeated use of a composite node which represents different layers in the hierarchy. A manufacturing organization is always represented in a hierarchical structure (refer to Chapter 3). Each composite node in STEM is defined as a composite class. When a model contains many composite nodes, it takes a lot of space to save them. In addition one disadvantage of the LISP environment is the limited memory capacity. Thus STEM is mostly suitable for modelling a system which contains similar processes or sub-processes, such as the manufacturing of printed circuit board (PCB) systems. These constraints of STEM have made it very difficult to create a detailed model of manufacturing organizations and have reduced the accuracy level of the output.
Although the research has met the original objective in the broad sense, limitations identified above imply that more work has to be done in order to develop the integrated methodology and to enhance the simulation model.

**Improvement to the integrated methodology.** As a new piece of work in the field, the following issues related to the integrated methodology have to be studied:

--detailed feasibility study
--one single environment for both static system description method and dynamic simulation method
--if information modelling is required between functional modelling and dynamic modelling what information should be involved?

**Improvement to the hierarchical simulation model.** These improvements have been discussed in sections 16.3 and 16.7 respectively.

From the discussion the conclusion is that the research has carried out the first step towards the development of an integrated methodology and hierarchical modelling of manufacturing organizations. However a lot of work has to be done to improve the integrated methodology and the software for the integrated modelling of manufacturing organizations.

### 16.3 Generic Nature of the Model versus Specialised Modelling

The aim of this section is to study the nature of this hierarchical simulation model. It first briefly states the principal requirements of a manufacturing organization model (refer to Chapter 6). It then moves onto the discussion of two types of models applied in industry in respect of their application capability, generic model and specialised model. Finally the nature of the model built from this research is assessed and the potential improvement is pointed out.

At present, the majority of simulation systems in manufacturing industry are designed only for a particular type of lower level manufacturing systems such as flexible machining facilities, assembly line and transportation systems [250] [109] [23] [175]. However the design and detailed analysis of a manufacturing organization require that a company wide model should be created (refer to Chapter 5) and in many respects this model is different from these lower level manufacturing system models (refer to Chapter 6).

The simulation model is used as a working environment, the kernel of the integrated methodology (refer to Chapter 9). This model should be adequately and efficiently built from a functional perspective, i.e., IDEF0 representation, in the way the system is organised and operated. It should capture key issues modelling both the relationship between the departmental areas within each layer and the interaction between different layers to provide
the user with a reasonably realistic and effective output in as short a period of time as possible and inexpensively. The principal requirements are as follows:

--adequate representation of key system features
--effective outputs
--user interface for efficient model building, modification and manipulation
--powerful facility for representing system performance
--capability to be extended for a further detailed model.

Two types of models may be classified in manufacturing, generic and specialised. The distinction between a generic and a specialised model is not a black and white issue, therefore the degree of generality and speciality is only a relative measure. For example, for particular types of manufacturing systems such as FMS, some models can be considered as generic [274] [175], others as specialised [4]. In a more broad sense, they are all specialised because they are restricted only to FMS systems.

Generally speaking a generic model is a model of a specific type of system written in such a way that certain parameters can be altered by the user through the data [51]. A generic model usually avoids any computer programming and it is easy for a non-specialist to quickly get the results within an acceptable cost. Although a large number of systems may be studied and many features captured, the capability of modelling a particular system still depends on the incorporation of special features in the model (refer to Chapter 10). A generic model may not be capable of modelling special applications and is not suitable for detailed studies [51] [196].

A specialised model is tailored to meet particular requirements of a specific system or a special situation. Generally this type of model which is interesting to industrial engineers can provide detailed studies of the system and solve real problems since the model is specially developed for a particular purpose. Consequently it is difficult to apply it to other systems or it requires special expertise to modify the model. In addition it takes a longer time to develop the model and to learn it [274] [175].

The simulation model built as a part of the research reported in this thesis is considered to be a generic model for a range of factories in terms of type and organizational structure. It is not dedicated to any particular manufacturing organization and the one chosen to be the case study depicted in Appendix VI is just a representative example. At company and shop levels, the model contains those areas which are found essential in many companies and perform the basic functions. Composite nodes defined in the model may be applied in other models that might be written for particular factory studies (refer to Chapters 9 and 10).
The relative merits in the building of a generic model or a specialised model is influenced by various factors. First of all because of the restricted skill in modelling required and the nature of generic and specialised models discussed above, a generic model can be expanded more readily than a specialised model.

Considering the time spent in building the simulation model, the first stage is to build the initial static model. The second stage is to create the simulation model and the third is to run and obtain adequate and approximate modelling data. At the present time the period of time taken to define the generic model is less than to build a specialised model because more detailed specific features have to be contained in a specialised model. Coupled with time, the cost involved in developing the generic model is less than in building a specialised model.

In the respect of application capability, a specialised model is more applicable to a particular situation than the generic model. Therefore it can solve application problems that a company may have. In general, decision making in industry is such that the speed of the modelling process must take account of the speed of changes of company objectives. Thus this perception of the relative ease of use of the generic model or a specialised model is to obtain an adequately sound dynamic simulation output as quickly and cheaply as possible. To pursue to a very detailed level, modelling is only a serious proposition in a very limited number of cases and the penalty is a longer time taken in the production of the model.

As a result, the decision made at the beginning of the project was to build a generic model. However there is some conflict among the industrial users that will perceive a different purpose in using a simulation model. Thus further research must produce methods of building the generic model which meets the requirement of most manufacturing organizations for initial studies, in a short period of time within an acceptable cost. This generic model developed in this research allows the easy extension to a specialised model. In order to do so, several issues have been addressed.

Identification of generic nature and specialised functions of manufacturing organizations. Different companies are usually organised and operated differently but a basic classification can be made as discussed in Chapter 3. Therefore the model must be built only for one type. In order to abstract a model containing common features found in that type of manufacturing organization, there is a need to reveal the functionality of each area, analyse different issues encountered according to their roles and then classify them according to their functionality. This should be done through collaboration with the industrial companies.
Investigation of the appropriate level of generality in the model. Based on the study above, it is important to develop standard sub-libraries from standard STEM libraries with common manufacturing features not only for the physical manufacturing operations [76] but also higher level decision making (refer to Chapter 10). The arrangement of these features should be done hierarchically in the data and knowledge base to provide as much generality as possible and at the same time reduce the difficulty in making them specialised (refer to Chapter 9 and section 16.4).

**Easy expansion.** Coupled with the last two issues is the need to identify an easy approach to build a specialised model to meet particular requirements based on these generic sub-libraries. The technique of object oriented programming is the proper approach for this application (refer to Chapter 9 and section 16.4) [14] [274]. Starting from specialised classes built from either standard STEM classes or new classes from scratch, sub-libraries are built by specialising standard STEM methods or writing new methods around these specialised classes to describe specific situations. Thus these methods are organised in layers of increasing detail to represent state transformations in a hierarchical way [14] [274] [174]. Both the explanation facility (refer to section 16.6) and the documentation on the methodology and techniques (refer to Chapters 9 and 10) used for the expansion can contribute and guide the user to quickly build a specialised model. However this requires knowledge and experience in LISP, LOOPS and STEM.

In this way, a model with special requirements can be easily built using the available generic sub-models defined and the methodology provided from the research.

The generic model developed from the research allows estimation of system performance covering company and shop levels. For the more detailed analysis, the model can be run containing two more lower levels, cell and machine levels. This gives the user certain flexibility for different purposes. However as an exploratory investigation into building a company wide model, some issues such as what nature a company wide model should have, generic, specialised or somewhere between, techniques used to easily build a model with required configuration for a particular purpose and how they may influence the quality of a model have to be further studied.

16.4 Justification of Hierarchical Modelling and Discussion of Alternatives

This section justifies the hierarchical modelling used in this work in comparison with alternative approaches. It identifies the reason why the hierarchical modelling approach is used from a modelling principle point of view.
In simulation modelling, there are various ways to build a model in terms of object arrangement in the model, graphical representation of the model, modelling procedure and model abstraction level. Review of recent literature and reconsideration of some of the old publications have lead to the conclusion that there is some degree of confusion in the use of the word hierarchical. A number of authors [274] [76a] [197] have chosen to define their approaches as hierarchical modelling just as in this thesis where a particular stand has been taken.

Object arrangement represents the way how objects contained in a model based on object oriented programming [14] [76a] are organized. Graphical representation of a model illustrates the diagrammatic display of system elements and their relationship in the model [274] [14]. The modelling procedure indicates the process used to build a model: top-down, bottom-up or flat [14] [76] [274]. The model abstraction level defines the transformational actions for the system elements based on the operational assumptions of the system [274].

Considering the way to organise objects, one example reported by Fegan et al [76a] is ISI. This method concentrates on the hierarchical abstraction of objects based on the principle of object oriented programming. ISI was developed as total simulation support environment for SIMAN. A very important feature of ISI is that it offers hierarchical modelling facilities which allows the development of libraries of sub-models, macros. In this approach, higher level building blocks (macros) are constructed from the basic SIMAN building blocks and then used subsequently to build models. In this way, the model is finally built from libraries of hierarchical macros.

One advantage of this method is that repeated sub-systems only need to be modelled once in a hierarchical system and the libraries of sub-systems models from one model can be used in other models. Another is that these sub-systems can be built and tested independently, expediting model debugging and validation. The final one is that it allows the operation of the model and the results to be represented at any level. However one disadvantage is that it is only suitable for modelling a system with similar sub-systems [76a] [14] [197].

The STEM approach is different from the ISI method in the sense that the model can be developed both top-down and bottom-up. It is also based on the principle of object oriented programming therefore has the advantages found in ISI. Furthermore the model can be represented in a single root map or in different composite nodes hierarchically (refer to Chapters 9 and 10) [196] [197] [14]. Therefore this approach covers aspects of object arrangement, graphical representation of the model and modelling procedure. However it
has the limitation that although a system with non-similar processes can be modelled, the size and execution of the model is restricted because of the limited memory capacity and LISP computing environment (refer to section 16.2).

From a graphical representation point of view, the contrast to the hierarchical modelling is "flat" modelling which means the whole system is modelled in one layer. With this approach, the basic principle of this approach is to arrange each basic element of a system graphically within a single layer representation. This approach allows easy modification because any internal change to each element does not affect others but only its own performance. Therefore it is suitable for lower level modelling such as flexible machining cells or machine level where design and analysis of physical manufacturing systems are the central issue. The simulation models reported in [175] [274] belong to this type of modelling. However objects representing these machining facilities in a simulation model can be organised in a hierarchy applying object oriented programming principle like the approach used in [274]. Modelling a company wide system by flat modelling makes it difficult to develop and handle models because the testing of a whole model takes a longer time and the massive layout of the system may cause confusion to the user. In addition, it is disadvantageous when a modelling team is working on the project [76a].

The position taken by Wang [274] mainly represents the abstraction level of the model in terms of the modelling details contained. It was assumed that there can be different levels of decision and description knowledge associated with different level of transformational actions, thus the model is operated at three different levels with gradually more detailed information. Level 1 aims at the basic sizing of a system design by providing a quick estimation of the system performance; level 2 is designed to study the flexible integration effects associated with work flow in flexible machining systems; and level 3 can be used to analyse and predict the influence of both work and tool flow on the system. This method allows the model to be used for different purposes and provides a smooth transition between different design stages.

Hierarchical modelling of manufacturing organizations in this research project denotes the process used to build a model in a top-down manner. It allows the level by level modelling of the organizational structure of a company, properly representing the activity relationships, the information flow and the flow of materials. As the model evolves from general to specific, activities and associated information and material flows are decomposed into more and more detail. This modelling technique allows simulation to provide both static system analysis inferred from IDEF0 representation and data on dynamic performance of a factory. At the present time, the work is restricted to dynamic data, the static system analysis is available but they are not co-ordinated (refer to Chapters 7 and 9 and section 16.2).
The technique of this approach is influenced by the IDEF0 method since decomposition of activities leads to the hierarchical model naturally. In this approach, the model is built from its highest level and is constructed in a form of IDEF0 representation. Any box at one level may be decomposed to a lower level showing the detail. Again nodes at this lower level may be decomposed to a further level down. This decomposition process is continued until the desired detailed level (refer to Chapter 9). Therefore the essential role of hierarchical modelling is the guidance in the building of a model, to provide a systematic and efficient way for modelling a complex system since it guides where to start building a model and in what sequence.

Hierarchical modelling employed in this research allows building of a model following the organizational and operational structure thus corresponding to the real systems. Since it is very close to the way people perceive systems [174], the development of complex multi-level models becomes easier. It also provides modular capability which means that sub-systems can be re-used in other models. Coupled with the structure hierarchy, objects are arranged in hierarchies as well [1] [14], providing the easy environment for knowledge and data management. However one disadvantage is that any change at one level will affect upper levels. Thus it is time consuming to keep the consistency of the model, especially a lot of time wasted in updating the model.

Hierarchical modelling has been proved to be a useful way to model manufacturing organizations because it allows a company to be modelled in a systematic way and represented in a graphical context. However, the model’s scope is mainly determined by IDEF0 representation in terms of levelling and contents of the model. Therefore there is a need to enhance this method by identifying criteria to determine how many layers and what should be contained in each layer (refer to sections 16.4).

16.5 Rationale for Linking IDEF0 and STEM

This section aims to state the reason for linking IDEF0 and STEM. It first states briefly why a static system description is required in the integrated methodology and identifies reasons why IDEF0 is used for this task in comparison with other system description methods. It then discusses how to link IDEF0 and STEM and what should be involved. Finally it draws the conclusion based on the identification of advantages and disadvantages of linking these two modelling methods.

The modelling of a large and complex manufacturing organization requires an integrated methodology because of the limitation of current modelling methods applied in industry, difficulties met in modelling such a system (refer to Chapters 2, 5 and 6) [208]. The integrated methodology must be designed in such a way that individual modelling
approaches making up the overall system have to be fitted together, checked and operated so as to achieve the overall objective in the most efficient way [275a]. Therefore it was decided to use two modelling methods to design and analyse manufacturing organizations, a static system description [102] [141] and a dynamic simulation method [239] [264].

From a modelling principle point of view, the static system description is the first stage in building a model (refer to Chapter 8) [75]. The development of a simulation model requires a system description which is combined with timing and procedure requirements to form a dynamic computer simulation model [239] [75]. However for most current simulation models there is not a complete understanding between the model designer and the user as to the purpose and structure of the simulation system [75]. This is because the structure and operation of a system is only embedded in coding. The relationship between elements contained in the model and how they are operated are not easy to understand [175] [274].

Because of the size and complexity of manufacturing organizations and the large number of complex interactions, it is difficult to build a model without following a certain formal method. Consequently a static system description is needed, combined with dynamic simulation to provide an integrated methodology. The major requirements (refer to Figure 5.5) are identified as following:

--a systematic, highly structured and top-down technique [75]
--graphical representation [102]
--easy to understand
--easy transition to dynamic simulation
--thoroughly documented and widely used and recognized.

IDEFO technique is selected because it meets the above requirements with key features: hierarchical and consistent system description, graphical representation and widely accepted and used (refer to Chapter 7) [102] [157] [141].

Other alternative methods (refer to Figure 2.27) discussed in Chapters 2 and 7 such as process flow chart [239] and GRAI method [13], but the process flow chart is restricted to the representation of software logic and GRAI is for description of production management. Input/output analysis is a method of generating all possible ingredients required to give all possible results [7] but it does not concern the conditions and mechanisms to carry out that process.

The major competitors to IDEFO are data flow diagrams, structure charts, structured analysis and design technique (SADT) and control requirements expression (CORE). However, the data flow diagram [208] [7] [18] only shows the flow of data through a system and there is no description of processes. Compared to data flow diagrams, IDEFO models
can provide a more consistent, multi-layered system specification and design [208]. A structure chart [275a] is a representation of a system in hierarchical format but it does not show the sequence in which the system components will be executed and which conditions cause activity. Figure 3.4 is a typical example of a structure chart. SADT [18] [7] originally developed by D. T. Ross of SofTech is the initial form of the IDEF0 technique but IDEF0 is supported with detailed guidance such as a description of the people and roles involved and how to communicate the various tasks, various means of interviewing and constructing lists, matrices, glossaries etc. Therefore compared with IDEF0 it is not fully developed and not widely used. CORE [7] developed by British Aerospace in conjunction with an associated software company, System Designers plc, is a development of SADT but with a number of additions. It is not suitable for system description because of additional tabular collections and complex diagrams required. It is not as popular as IDEF0 in industry.

As a result, IDEF0 has been considered as a preferred choice of structured system description and design methodology supported with the significant experience in the use of IDEF0 methodology in the department.

STEM was originally selected to link with IDEF0 because the investment had already been made in the past, the system was available even in its development stage in the department and there was an intention to explore the usage of STEM for simulation modelling. Another reason is that STEM can meet the requirements for dynamic modelling discussed in Chapter 5 with two superficial features [14] [196] [197]: capability to develop a model hierarchy using its graphical capability.

The main competitor to STEM may be SIMKIT [234] [76]. Both can be extended to contain major areas in a company, providing a powerful environment for model design, modification, and execution. The standard libraries contained provide the right level of generality and flexibility and any specialised sub-libraries can be re-used in other simulation systems. Another advantage of SIMKIT is that it has developed standard sub-libraries containing common manufacturing functionalities based on the standard SIMKIT libraries. However one important disadvantage of SIMKIT is that it does not have the hierarchical model development facility.

Other available simulation tools and simulation language such as WITNESS [113], MAST [139] and SIMAN [243] [76] [188] are not suitable for this research project because of the difficulty for higher level decision making and hierarchical modelling. In addition they are not easily linked with IDEF0[59] in terms of structure representation and modelling environment (refer to section 16.3).
The next issue is how to link IDEF0 and STEM. Since graphical and hierarchical representation of activities is the essential feature of IDEF0 methodology, the linkage between them is through the graphical representation (refer to sections 16.2 and 16.4). IDEF0 representation sets the boundary to the dynamic simulation model and also illustrates the configuration of the system modelled from the functional perspective (refer to Chapter 8), not information, resource and decision view [13] [208].

What should be linked is another important issue tackled in considering the principle of linking IDEF0 and STEM. Transition from IDEF0 to the model building involves transformation from graphical representation in IDEF0 to simulation model configuration and addition of decision making for performing these activities since IDEF0 only captures the activities involved and the relationship between them but not how they are performed. Therefore the link between IDEF0 and model building only provides an aid for the modeller to build the model and help the user to easily understand the structure of the model (refer to Chapter 8). The complete modelling system requires others such as key features captured in the model, powerful interface, efficiently manageable data base and knowledge base (refer to section 16.4).

The linkage of IDEF0 and STEM is suitable for the complete analysis of a complex system (section 16.2). The IDEF0 representation makes it easier for the user to understand the structure of the system modelled and the internal relationship between activities. IDEF0 representation is a design aid for the modeller and an interface between the user and the simulation model (refer to Chapter 8). It provides two important characteristics which represent good system analysis and design methodology. One is the degree to which they assist the analyst and the other is the ease of understanding both the methodology and the documentation produced by it [239] [234].

From the discussion of Chapters 7 and 8, it is evident that the structure of both IDEF0 representation and hierarchical simulation model are designed to a particular situation. This to some extent limits the wide application (refer to section 16.4).

Although the IDEF0 representation was drawn up based on some preparatory work such as information gathering, studying systems, interviewing experts etc, and certain methodology has been used in the diagraming process. It sometime needs to be changed during the progress of the project. As IDEF0 is a hierarchical decomposition method any change made at any level can lead to syntax inconsistency at other levels. This can be time-consuming when the inconsistency causes a new diagram or diagrams to be drawn [59] [208] (refer to section 16.4).
Comparing Figures 7.1 to 7.8 with Figures 9.2 to 9.15, it is seen that there is no direct correspondence between the IDEF0 representation and the STEM model because the linkage between nodes in STEM only shows the connection relationship not each individual information flow (refer to section 16.2).

Therefore further work is required to improve the above problems. First of all, the challenge is to investigate the methodology to guide modellers all the way through the system description process in order to help them to justify the IDEF0 representation. Therefore it is needed to identify key factors which influence the functional representation. These factors can be gained through the detail of study of manufacturing organizations by visiting industrial companies and references (refer to section 16.3).

Reducing the difference between IDEF0 representation and the STEM model has been partially achieved by the new research group by building a simulation model based on the IDEF0 representation implemented in the same software environment. This can remove the obstacle of indirect correspondence between IDEF0 representation and STEM simulation model and ensure that the hierarchical modelling technique is fully implemented (refer to section 16.3).

It should be emphasised that the linking of IDEF0 and STEM is only a medium illustrating that the integrated methodology works and how it can be applied. It does not necessarily mean that they are the only choices among different system description and modelling methods (refer to Chapter 5). As a result, the research has found that STEM in many aspects needs to be improved in order to properly fulfil the task (refer to section 16.2) or other simulation package should be selected.

In conclusion the combination of different modelling methods has developed a novel approach for modelling manufacturing organizations and moved one step in the potential research direction for the future [208] [217] [75] but more work as identified above has to be done to provide a sound foundation for the integrated methodology.

16.6 Requirements to Improve the User Interface

This section discusses the function of the user interface in the modelling system, identifies reasons why the model specialisation interface needs to be improved from the user's point of view and illustrate how these improvements are achieved.

As discussed in Chapter 8, a methodology has been established for hierarchical modelling of manufacturing organizations. Briefly the modelling process starts with the specification of manufacturing goals or identification of problems being solved by simulation. Once the analysis objectives have been stated [274], the modelling procedure includes
model building, experiments with the model and analysis of output results from the experiments. It is evident that one significant element of the modelling system which plays an important role during all stages of this modelling process is the user interface.

Considering the potential end users, industrial engineers who have manufacturing knowledge but little simulation and programming experience and analysts who have both manufacturing and programming knowledge, the user interface must provide the facility of guiding them in carrying out a complete analysis of manufacturing organizations. The following user interface requirements are identified as important and necessary for efficient and effective application of the modelling system [274] [258] [32] (refer to Chapter 11):

--right level of user involvement
--easy to use
--explanation capability
--easy access to the simulation system
--flexibility (efficient for both industrial engineer and analyst).

The user interface of this research project refers to the facility specialised for data and rule entry, output results collection and explanation because STEM provides a powerful and convenient facility for model building, modification, and model execution (refer to Chapters 9 and 10). This user interface has been developed with the following features (refer to Chapter 11).

**Four menus.** Four menus have been built to provide facility required for the user’s interaction with the system, the global menu to manipulate other menus, data entry menu for data input, rule entry menu for rule selection and output menu for collecting outputs.

**User involvement.** This knowledge based simulation system is capable of handling the variables in two ways. One is to set the initial value in the data base, the other is through the user interface. The first method requires programming experience in LOOPS and LISP and is not suitable for an industrial engineer. The latter method is easy for a user with little programming experience but the challenge is that not all values can be managed in this way because a lot of information is contained in the model and it is impossible to enter every single data and rule in this way. Through the identification of key numerical data and operating strategies which will significantly influence the system performance, the interface only provides a facility for those information entry [274]. For rules, options are available for different situations.

**Easy to use.** For data and rules entry, there is no required input sequence and the new entry will overwrite the old value but the recommendation is to enter the input information
in a logical order, displaying one menu at a time by selecting from the global menu, a master menu as a control centre for accessing other menus. After running for a period, the user can easily access to any output information at any time.

**Simple explanation facility.** On each menu, there is a simple help information facility explaining the meaning of each item contained and the influence of different input value (refer to Figures 11.1 to 11.5). These can help the user understand the simulation system and prompt the user to enter the right input information.

**Only for rule selection.** For rule manipulation, the interface at present only allows the user to select one from the available rules. This seems very efficient for an industrial engineer to control the simulation system. The user can select a rule from a library of existing ones which are displayed on the rule entry menu. For each rule option, different entry influences the action considering the decision point [274].

With regard to the flexibility and user-friendliness of the interface, the following limitations have been identified (refer to Chapter 11) [274] [185] [229]:

--no data format check
--no explanation against selected item
--no facility for new rule entry.

The interface has no data format check facility to guarantee the right data and rule entry format in the first place. This may cause more time wasted running on the wrong data or rules. Coupled with this is that the model does not possess an on-line explanation environment which is a facility to display the meaning against the item selected, explain impacts of different entry and prompt the user with the right input format. Although potential end users may be either industrial engineers or analysts, the user interface does not have options to allow the experienced users to manipulate the modelling system [152] by entering new rules.

For the syntax check, more work needs to be done on the programming. According to the format of input data and input rule, the model should be able to recognize the right form which can be accepted by LISP or LOOPS and prompt the user where there is the error and what is the right input format (refer to Chapter 11). This syntax check only tests the input format such as a list, a numerical datum or a string. Thus the content of information entered has to be checked by the user himself.

The explanation facility should be provided against the selection of an option from a menu. For rule selection, usually several options are available. For example, scheduling rules at machine level may involve first in first served, shortest processing time and shortest total processing time etc (refer to Chapters 10 and 11). Therefore when selecting a rule, the model should display that rule with the related interpretation and it should also prompt the
user to take certain action against different options (refer to Figure 11.4). For data entry, it should provide guidance for input format and briefly explain the impacts of different data value. As a result, this can keep the errors arising during the entry process to the minimum [185].

New rule entry has to be considered and done carefully. First of all, a feasibility study is required to investigate whether rules are needed and identify potential new ones. A great challenge is that there should be a way to edit this new rule, insert it into the knowledge base and use it. In the STEM environment, all rules for decision making are built around classes either in rule set or LISP procedure format [274] [14] [9]. Being able to express a new rule, the user has to determine to which class this new rule should be added and know the variables of the decision point class [274]. Once the class is identified, all rules built around that class have to be displayed. By selecting a new rule entry, a window should be opened for rule editing. Other auxiliary rules may be needed or existing rules have to be modified to allow the new rule to be called. After a new rule is entered, it should be automatically added to the existing rule library and can be easily modified and even deleted.

All these mean that on the one hand, more menus should be added to manipulate a new rule entry. On the other hand, more work should be done to improve the structure of existing rules to make it easier to enter a new rule. One possible solution is to divide rules into groups according to their role in the model. For those rules which should be manipulated by the user, either because they greatly influence the system performance or they represent different options of, the model should be as simple as possible. Thus it is easier for them to be modified and simpler for the user to understand.

The above discussions have addressed issues on how to improve the user interface to provide more flexibility, allow wide application and offer an easy way for the user to efficiently control the simulation system. The first two improvements, input data format check and on-line explanation are not difficult and only require more programming. The last one is considered as the most difficult one because it is also influenced by other issues which should be fully studied, such as important strategies, possible options, arrangement of rules, classification of different types of rules etc.

In summary the modelling environment has provided a simple but efficient and friendly interface to control the simulation and get the results, but limitations identified above need to be improved to offer a more powerful user interface to ease the running process and provide more flexibility.
16.7 Practical Use of the Modelling Tool

This section mainly discusses the applicability of the modelling tool through an industrial case study (Chapters 13 and 14) in order to assess and validate the model from a practical use point of view.

The overall scope of the simulation model has been set up in Chapter 8 to specify what problems are going to be tackled. It only covers those issues closely related to design and operation (refer to Chapter 8) [144] [51] [234] [84]: resource requirements, evaluation of operational procedures, performance evaluation, and analysis of WIP level.

As discussed in Chapter 14, utilisation of working areas and machines indicate their usage under certain production requirements during a period of time. These figures can help the company find out some problems, such as is the resource capacity sufficient for the production requirements; if there is any bottle-neck which actions should be taken, introducing new machines into the system, changing the part route, or re-scheduling parts into the system; what is maximum and minimum resource requirements etc. However, these resource requirements can only be used for the rough capacity analysis because of the assumption of infinite capacity of working areas, assumed input level, limited number of machines contained, simplified bill of material, short computing time etc.

Different operating rules have been defined in the model, discussion in section 14.3.5 and 14.3.6 have investigated the influence of scheduling rules, material movement rules, inspection rules and other rules etc. These results indicate that the overall system performance is determined by many factors. The change of a strategy only illustrates one aspect; the combination of different strategies significantly influences system performance. Therefore the output results provide guidance for the company to identify which rule can lead to promising results under a particular situation such as an urgent order and help them to implement some strategies which are not used in practice. It has to say that the case study only concentrates on these important and potential changed rules, for other rules only one case which best represents the real operation of the company has been tested (refer to Chapter 14).

The system performance was illustrated by the overall performance displaying total production requirements, total production throughput and total products required in a defined horizon. By comparing them, it could be found whether the company meet customers' requirement or not. Product, component and resource performance from different aspects indicate the overall performance. However, the number of products going through the system is far different from practice, thus the results can only be the estimated analysis (refer to Chapter 14 and Appendix VIII).
From the discussion in Chapter 14, it is evident that the output results of the model are mostly suitable for design study at the early stage roughly determining resource requirements estimating work in progress and overall performance and evaluating the impact of different operating rules because of the defined scope of the model and accuracy level of outputs (refer to Chapter 14).

The decision was made to restrict the scale of the generic model used in this research. As this was the founding piece of work done in this area, it was considered essential to have a generic model which has the scope to be applied to industrial problems so that it could give an adequate test of the integrated methodology (refer to Chapters 8 and 13). The key factors significantly influencing the output results are assumptions made in the model, short running time and accuracy of the data from the company.

The first factor is a two-fold issue. On the one hand, approximation is always required because it is impossible to create a model which exactly represents the reality due to the difficulties met in building a complex model; constraints of computing environment; time, cost and effort spent. Thus the level of approximation should be dealt with carefully. It is mainly influenced by the objective of the model and computing capacity. On the other hand, the model is of value only if the insights generated can be used to impact reality. Since the model ignores activities in other departments, such as engineering, stock control, finance and purchasing, the problems which can be dealt with are limited.

The second factor is the short running time. In Chapter 14, it has been stated that since the computer is unable to run longer due to the limited memory capacity, the output is not the steady state results because the initial condition of the model is always assumed empty. The third reason is that some data required by the model is not available from the company. Therefore the accuracy of assumed data is limited such as actual production requirements, work area capacity, time spent in pre-production etc (refer to Appendix VII).

As a result, a lot of work has to be done in order to improve the above limitations to enhance the model to solve real problems. This covers two areas of work. For the computing environment, the suggested work has been discussed in section 16.2. For the input data required, one possibility is to investigate what information a company uses to manage the production in order to reduce the difference between the reality and the model. Therefore another important improvement is related to the scope of the model.

The improvement of the model includes many issues such as flexible configuration (refer to section 16.5), nature of the model (section 16.3), enhancement of the user interface (section 16.6), applicability. The last requirement can be achieved by enlarging the model in terms of departmental areas involved and activities and operating rules contained.
However, issues such as what should be involved, how much detail have to be considered, what real problems are going to be solved etc have to be fully investigated in order to provide a company wide model for the industry solving their problems.

16.8 Concluding Remarks

Pursuing the global aim of researching into the integrated methodology for modelling manufacturing organizations has involved many facets of study and this chapter assesses those key aspects which may influence the decision making during the project and reflect the quality of the work.

This research has set a new research direction in the community of manufacturing organizations design and analysis and developed the static system description -- IDEF0 representation and a hierarchical simulation model to provide an integrated environment for modelling the system in both static and dynamic points of view.

On the academic front, the research has identified those factors significantly indicating why the integrated methodology is needed, what are general requirements for it and for each modelling method, how these two modelling methods should be linked. In industry the research has provided a hierarchical simulation model which can assist companies to identify their problems such as bottle-necks, influence of different operational rules, potential changes necessary to improve the system performance in a cost-effective way and in a short period of time.

It is expected that the scope of the integrated methodology will be progressively widened to investigate more aspects related and encompass as many aspects of a system as possible to provide a more realistic and adequate model.
Chapter 17
CONCLUSIONS

The conclusions formulated from this research project are as follows:

1. This integrated methodology brings a novel approach for integration of modelling tools, it thus enhances their efficiency, effectiveness and usefulness. It provides a new method for modelling complex and hierarchical manufacturing organizations in a systematic way and makes an early contribution to hierarchical modelling of manufacturing organizations. It also fills the gap caused by different modelling methods applying to different areas in the same organization.

2. The introduction of standardised system representation method -- IDEFO has been proved to be a valuable medium prior to the development of the simulation model, thus providing an interface with regard to the scope and structure of the model and interaction between activities.

3. The work performed in this research on the dynamic modelling has resulted in the design and development of the hierarchical simulation model. The hierarchical IDEFO-like structure of the model provides a new way for model configuration. Its modular architecture enables any sub-systems or a single functional node to be re-used in other simulation models. The specialised data base provides an easy and effective method to organise the objects and functional areas found in manufacturing organizations. The specialised user interface allows the user to easily interact with the model and govern the modelling system.

4. The manufacturing functionalities have been demonstrated through the modelling knowledge put in the model. A comprehensive set of rules regarding the major decision makings in key areas and real manufacture of products has been embodied in the model. This enhances the capability of the STEM simulation environment and provides a specialised manufacturing simulator for manufacturing engineers which can handle a wide range of manufacturing environments.

5. The capability of the model to be used for both quick estimation and detailed analysis has been demonstrated. This has proved the flexibility provided by the model capable of being applied at design and operation stages. Additionally, both data and rule driven environment allows the recognition of promising system performance through applying alternative data and rules.
6. The case study has demonstrated the use of the methodology and illustrated the applicability of the simulation model. The experience shows that the original methodology is not only essential in developing the hierarchical simulation model but also provides useful guide-lines for the user in applying the model. The limited evaluation of the model illustrates the proper approximation made in the model and identifies the influence of alternative data and rules.

7. The STEM general simulation package used in the research project has proved to be an effective and powerful facility for manufacturing organizations modelling. It provides efficient means for easy model building, modification, debugging and graphic output. However, further improvements on STEM are needed to cope with large scale system and to provide efficient model execution environment.
Chapter 18
FURTHER WORK

18.1 Introduction

Based on the concluding discussion in Chapter 15, critique in Chapter 16 and conclusions in Chapter 16, the recommendation on the further work is stated in this chapter.

18.2 Integrated Environment

As analysed in Chapter 7, there is a close relationship between IDEF0 representation and the simulation structure in SIM. These two parts of the work at present are done separately. It is intended that a SIM simulation environment be developed to include, through additional libraries, such as an integrated environment. An initial model description will be input in IDEF0 format, and whilst an IDEF0 description is inadequate for dynamic modelling, methods of consistency checking of information flows will be investigated. The IDEF0 description will then accept additional specification of dynamic features of the system elements so that dynamic modelling at a variety of levels of detail. The creation of the IDEF0 representation can be developed directly on SIM by using the powerful graphic capability. Although the available software [248] has made the building of IDEF0 representation much easier, the direct development if IDEF0 presentation on SIM is believed to be more efficient.

18.3 Modelling of Interactions

To enable the integrated method to be used in all manufacturing organizations correctly and to minimise the difference between the real performance of the system and the performance at the design phase, more detailed activities should be modelled, especially the interaction between these activities should be considered. For example, information flow between engineering and production and within production should be involved, thus the operation status can be sent back to production planning to generate a realistic plan.

18.4 Modelling of the Engineering Function

With the emphasis put on the analysis of the production activity, more effort has been placed in the production areas in a company. However, engineering is also an important
function in terms of productivity, various range of product types, and lead time reduction. The engineering node in the model can be changed to be a composite node, including product engineering, manufacturing engineering, and production engineering. This engineering function will generate drawings of the components and relative specifications, according to customer specifications, company strategies and tactics. The ease with which nodes can be defined and their functionality modified makes it possible to develop nodes representing subsystems by modelling paradigms other than simulation.

18.5 Interpretation Facility of Results

The sophisticated manufacturing organization modelling environments are intended for use by those experienced in the application domain rather than in the field of systems modelling. Therefore, the simulation model should give assistance in interpretation of the results of running the model. It is intended that an analyser object is developed embedding an expert system to provide such assistance and possible guidance on the need for further experiment with the model.

18.6 Improvement on User Interface

As described in Chapter 11, the user interface of the model at present can only handle pre-defined information entry. To be more flexible and application oriented, the model should allow the user to enter new rules. The new rule entered has to use the specialised classes defined in the model, the associated instance variables can be newly added or can use the pre-defined ones. This will enhance the application capability and provide more flexibility for the user to support the particular application domain.

18.7 Linking with Available Packages

With more workstations available in the department and the related research work done in parallel with this thesis, it is believed that they can be integrated to provide a more powerful capability. Based on this distributed network, an available software -- knowledge based modelling system of flexible machining cell conducted by [274] can get the information from this model and make the detailed performance analysis of the cell. This somehow can compensate the disadvantages of both models and provide the correct and detail analysis tool.
18.8 Global Data Base System

As a general design and analysis tool of manufacturing organizations, the model requires a lot of information and operational strategies. Meanwhile, the information continues changing the status of a system, thus updating the information all the time. To support this simulation model and other potential packages and to get consistent results at the overall company level [203], a global data base system is needed to store and provide the required and newest information. This will ensure the accuracy capability of the modelling system and offer an analysis tool of real performance for manufacturing organizations. Furthermore, it is the potential development tendency of CIM systems.

18.9 Improvement on Computing Environment

In order to create more realistic and complex model and increase the running speed, the computing environment needs to be improved especially on the model size and memory capacity. The project undergoing in the department has made some achievements by modifying the STEM CNode concept and structure.
REFERENCES


2. Adelsberger, H. H., Pooch, U. W., Shannon, R. E. and Williams, G. H.

3. Adler, R. M.

4. Afentakis, P.

5. Amrine, H. T., Ritchey, J. A. and Hulley, O. S.

6. Ando, K., Takeshige, A. and Yoshikawa, H.

7. Anon.

8. Anon.

LOOPS (User Manual), Xerox Corporation, 1988

10. Anon.

11. Anon.


14. Artificial Intelligence Ltd.  

15. Azar, I.  


17. Ballakur, A. and Steudel, H. J.  


20. Banks, J. and Carson II, J. S.  


22. Baskett, F.  
23. Bastos, J. M. and Shires, N.

24. Beerel, A. C.

25. Begg, L. M. and Worsley, C. P.


27. Ben-Arieh, D.

28. Ben-Arieh, D.

29. Ben-Arieh, D.

30. Bernemann, S., Hellingrath, B. and Joemann, J.


32. Bewley, W. L. and Rosenberg, D. A.
33. Bilalis, N. G.

34. Birtwistle, G.

35. Birtwistle, G., Lomow, G. and Unger, B. et al.

36. Birtwistle, G. and Kendall, J.

37. Birtwistle, G., Lomow, G. and Unger, B.


40. Boaden, R. J. and Dale, B. G.

41. Boling, B. A. and Laymon, M. A.

42. Borgen, E. and Strandhagen, J. O.
"SIMMEK: an object-oriented analysis tool based on discrete event simulation",

43. Bovone, M., De Ferrari, V. and Manuell, R.

44. Bowers, E. A.
"Defining your system needs to select the proper material handling system", IE, Feb. 1987, pp 34-41.

45. Boykin III, R. E.

46. Browne, J., Harhen, J. and Shivnan, J.

47. Bullinger, H. J., Warnecke, H. J. and Lentes, H. P.

48. Bunce, P.

49. Burgoine, J.
"Beyond MRPII", Production Engineer, Jan. 1988, pp 31-33.

50. Burrow, L.
"The design to product Alvey demonstrator -- an integrated knowledge based system to support design and manufacture", Proceedings of the 1st Int. Conf. on Artificial Intelligence and Expert Systems in Manufacturing, IFS, pp 63-74.

51. Carrie, A.

52. Cellier, F. E.

53. Chaharbaghi, K., Davis, B. L. and Minett, S. C.

55. Churchill, G. A.

56. Clark, M. and Farhoodi, F.

57. Cleary, J., Goh, K. S. and Unger, B.

58. Cobbin, P.

59. Colquhoun, G. J., Gamble, J. D. and Baines, R. W.

60. Conway, R.

61. COSIMA Project Team

62. Crookes, J. G.

63. Davis, W. J. and Jones, A. T.

64. De Souza, R.
65. Denning, P. J. and Buzen, J. P.  

66. Doumeingts, G., Erkes, K. and Krezinski, P.  
"Integration of decisional and physical systems in the design of manufacturing systems", Proc. of 25th IEEE Conf. on Decision and Control, Dec. 10-12, 1986, Athens, Greece, pp 2212-2217.


68. Dubois, D. and Stecke, K. E.  

69. Ebrahimpour, M. and Schonberger, R. J.  

70. Eisner, H.  

71. Elsayed, E. A. and Boucher, T. O.  

72. Elzas, M. S.  

73. Erkes, K. F.  

74. Eschenbacher, P., Langer, K. J. and Dornhofer, K.  

75. Evers, K. H., Bachert, R. F. and Santucci, P. R.  
76. Faugnt, W. S.

77. Fisk, J. C. and Ballou, D. P.

78. FitzGerald, J. and FitzGerald, A.

79. Fox, M. S. and Smith, S. F.

80. Futo, I. and Gergely, T.

81. Futo, I. and Gergely, T.

82. Gaines, B. R. and Shaw, M. L. G.

83. Gaines, B. R. and Shaw, M. L. G.

84. Galante, G. et al.

85. Gaskins, R. J. and Tanchoco, J. M. A.

86. Gerelle, E. G. R. and Stark, J.

87. Gessner, R. A.
88. Globerson, S.

89. Gorden, W. J. and Newell, G. F.

90. Graefe, P. W. U., Chan, A. W. and Levi, M.
"A production control aid for managers of manufacturing plants", SIM-1, 1985, pp 55-64.

91. Graham, G. S.

92. Grant, H.

93. Groover, M. P.

94. Hadavi, K., Shahraray, M. S. and Voigt, K.

95. Haddock, J.

96. Haider, S. W. et al.

97. Harmon, P. and King, D.

98. Hartley, J.
FMS at Work, IFS, 1984, pp 30-90.
99. Hay, E. J.
The Just-In-Time Breakthrough Implementing the New Manufacturing Basics, John Wiley & Sons, 1990, pp 1-125.

100. Hayes-Roth, F., Watermann, D. A. and Lenat, D. B.

101. Heinz-Fischer, H., Los, R., Poles, H., Pohl, C. and Stienen, H.

102. Hicks, C., Braiden, P. M. and Simmons, J. E. L.

103. Hitomi, K.

104. Hitt, M. and Middlemist, R. D.


106. Holland, D.
"Strategic benefits gained from CIM", The Production Engineer, Vol. 63, No. 6, 1984, pp 14-16.

107. Hollingum, J.

108. Hollocks, B.

109. Hook, G. M.
110. Hooper, J. W.  

111. Hurrion, R. D.  

112. ISO.  

113. Iset Ltd.  

114. Iwata, K. F., Murotsu, A. and Yasuda, K.  

115. Jackson, J. R.  

116. Jackson, J. R.  

117. Jackson, P. and King, M.  

118. Jiao, H. and Popplewell, K.  

119. Johri, P. K.  

120. Jones, A. T. and McLean, C. R.  

121. Jones, D. A.  
122. Jones, V. C.

123. Joshi, S., Wysk, R. and Jones, A.

124. Kamper, S.

125. Kasturia, E., Dicesare, F. and Desrochers, A.
"Real time control of multilevel manufacturing systems", Int. Conf. on Robotics & Automation, 1988, Philadelphia, pp 1114-1119.

126. Kay, J. M.


128. Khoshnevis, B. and Chen, A. P.

129. Khoshnevis, B. and Chen, A. P.

130. Khoshnevis, B. and Austin, W. M.

131. Kimemia, J. and Gershwin, S. B.
132. Kiran, A. S. and Smith, M. L.

133. Klahr, P., Faugnt, W. S. and Martins, G. R.

134. Klahr, P. and Watermann, D. A.

135. Kleine, B.

136. Klittich, M.

137. Knapp, V.

138. Kochan, A.
"Hocus takes the risk out of planning FMS", *The FMS Magazine*, April, 1984, pp 91-93.

139. Kochhar, A. K.

140. Koenig, D. T.

141. Koriba, M. and Eyres, P. K.
142. Kornell, J.

143. Krajewski, L., Ritzman, L. and Wong, D.
"The simulation of real manufacturing systems for the improvement of inventory, productivity and service", Proc. IIF Fall Industrial Engineering Conf. 1982, Cincinnati, Ohio, USA, pp 241-249.

144. Kreutzer, W.

145. Lal, H. and Peart, R. M.

146. Lane, G. M., Fegan, J. M. and Nolan, P. J. et al.

147. Langen, P. A.

148. Lavery, R. G.

149. Law, A. M. and McComas, M. G.

150. Lawrence, A.

151. Lemoine, A. J.
152. Lenz, J. E.

153. Lenz, J. E.


155. Lockyer, K.

156. Lubben, R. T.

157. Mackulak, G. T.
"An examination of the IDEF0 approach used as a potential industry standard for production control system design", Automated Manufacturing (Ed. Gardner, L. B.), A Symposium Sponsored by ASTM Committee E-31 on Computerised Systems, 5-6 April, 1983, San Diego, California, pp 136-149.

158. Malloy, B. and Soffa, M. L.

159. Marshall, G. R.

160. Martinez, J., Alla, H. and Silva, M.

161. McFall, M. E. and Klahr, P.
162. McLean, C. R.

163. McManus, J. J.

164. McRoberts, M., Fox, M. and Husain, N.


166. Merabet, A. A.


168. Meyer, W. and Isenberg, R.

169. Mittra, S. S.
"Discrete system simulation concepts", Simulation, Vol. 43, No. 3, 1984, pp 142-144.

170. Monks, J. G.

171. Monks, J. G.

172. Montan, V. and Reddy, Y. V.

173. Mullins, L. J.
174. Nadoli, G. and Biegel, J.

175. Newman, S. T.


177. Norman, V. B. and Norman, T. A.

178. Nyen, P. A.

179. O'Grady, P. J.


181. Olgen, O. M. and Thomasma, T.

182. Oren, T. I. and Zeigler, B. P.

183. O'Shea, T., Self, J. and Thomas, G.

184. Painter, C. W.
"Last rites for MRP", Production Engineer, April, 1987, pp 33-34.

186. Parsei, H. R., Ward, T. L. and Karwowski, W.  

187. Peck, A. C.  

188. Pegden, C. D.  


190. Peterson, J. L.  

191. Pidd, M.  

192. Pierreval, H., Guinet, A. and Dussauchoy, A.  

193. Pivi, G. and Righini, G.  

194. Plossl, K. R.  
195. Pooley, R.

196. Popplewell, K.

197. Popplewell, K. and Jiao, H.

198. Pourbabai, B.

199. Powers, Jr. J. H.

200. Prabhu, V. and Baker, M.

201. Pritsker, A. A. B.

202. Raczynski, S.


204. Ram, S., Carlson, D. and Jones, A.

205. Randhawa, S. U. and Bedworth, D. P. E.
"Factors identified for use in comparing conventional & flexible manufacturing systems", IE, June, 1985, pp 40-44.
206. Ranky, P. G.
The Design and Operation of FMS, North-Holland, 1983.

207. Ranky, P. G.

208. Ranky, P. G.

209. Rayner, K.

210. Reddy, Y. V., Fox, M. S. and Husain, N.

211. Reddy, Y. V., Fox, M. S. and Husain, N.

212. Reisig, W.

213. Rembold, U., Blume, C. and Dillmann, R.

214. Riggs, J. L.

215. Roberts, S. D.

216. Robertson, P.

217. Roboam, M. et al
218. Rogers, P., Williams, D. and Wesley, P.

219. Rolston, L. J.

220. Rose, C. A.

221. Ross, D. T.

222. Rothenberg, J.

223. Roukens, J. and Renuart, J. F.

224. Rui, A. P.

225. Russell, E. C.

226. Ruzicka, R.
"Environment for SIMUL-R", ESC 89 (Eds. Murray-Smith, D., Stephenson, J. and Zobel, R. N.), Sept. 5-8, 1989, Edinburgh, Scotland, pp 268-274.

227. Salmon, M.

228. Salvendy, G.

230. Sarker, B. R.

231. Sartori, L. G.


233. Scheer, A.-W.

234. Schriber, T. J.

235. Schriber, T. J.

236. Schwetman, H.


238. Shannon, R. E., Mayer, R. and Adelsberger, H. H.

239. Shannon, R. E.
Shannon, R. E.

Shannon, R. E.

Shivnan, J. and Browne, J.

Skooglund, N. S.

Smith, S. F., Ow, P. S., Lepape, C., McLaren, B. and Muscettola, N.
"Integrating multiple scheduling perspectives to generate detailed production plannings", SME Technical Paper.

Smith, S. F., Fox, M. S. and Ow, P. S.

Smolik, D. P.

Smunt, T. L.

SofTech Inc.

Solberg, J. J.
250. Solberg, J. J.

251. Spur, G.

252. Stadtler, H.

253. Standridge, C. R.
"Performing simulation projects with the extended simulation system (TESS)", *Simulation*, Dec. 1985, pp 283-291.


255. Stecke, K. E. and Talbot, F. B.

256. Stinson, J. P. and Khumawala, B. M.

257. Stokking, E. J.

258. Strandhagen, J. O.

259. Sumichrast, R. T. and Baker, J. R.

260. Suri, R.

273
261. Suri, R. and Hildebrout, R. R.

262. Suri, R.

263. Suri, R. and Diehl, G. W.

264. Szymankiewicz, J., McDonald, J. and Turner, K.

265. Tanimoto, S. L.

266. Tersine, R. J.

267. Tersine, R. J.

268. Tzafestas, S.

269. Vaucher, J. G.

270. Villa, A.

271. Vinod, B. and Solberg, J. J.
272. Voss, C. and Clutterbuck, D.

273. Voss, C. A.

273a. Wang, M.
The development of an integrated methodology for the design and implementation of computer integrated manufacturing (CIM) systems, Proc. of 1st Conf. on CIM (Ed. Lim, B. S.), World Scientific, Singapore, 1991, pp 141-144.

274. Wang, W.

275. Warneck, H. J. et al.


277. Waterman, D. A.

278. Weatherall, A.

279. Wichmann, K. E.

280. Wichmann, K. E.

281. Widman, L. E., Loparo, K. A. and Nielsen, N. R.
282. Wild, R.

283. Wu, B.

284. Wu, N. L.

285. Young, R. E. and Vesterager, J.

286. Zeiglar, B. P. and De Wael, L.

287. Zhang, P.
Appendix I
MAIN FEATURES OF STEM

I.1 Introduction

STEM is a tool for building models of processes, simulating those processes and analysing the results.

In STEM, the following concepts are used. A process is made up of a flow of information between significant points in the process. What flows between the points in the process is called data, while the points in the process are called nodes. Nodes are connected by paths. The arrangement of nodes and paths is called a map. The data that flows from one node to the next is called a token. Figure I.1 shows these basic concepts.

All parts of a STEM simulation are referred to as objects, such as nodes, tokens, and paths etc. Objects have a behaviour, which determines how they act in the model, and attributes which are variables that hold data about the object. Attributes have values which can change during the simulation [14].

I.2 Model Formulation

STEM provides a number of classes of objects. Subclasses can be derived from classes. These inherit behaviour and attributes from their superclass, but can have their own behaviour and attributes that override or enhance their inherited characteristics [196]. Therefore, for any simulation, the model building is actually to specialise these existing classes or define new classes from scratch if necessary. These specialised classes can be stored in library files, which can be loaded incrementally on top of STEM, providing a specialised application 'personality' of the simulation environment. To compare the effects of trying different constraints or varying the process itself, STEM provides support for the development of variants and for the incorporation of separately developed simulation components.

In order to start and stop the simulation, two types of classes are always needed in any model, originator which generates the token required and terminator which holds the token and does not send any token to the other nodes. An example is shown in Figure I.2.
I.3 Hierarchical Structure Building

One important feature of STEM, is its ability to create, store, maintain and re-use composite entities. A composite node is a single node composed of other nodes and the paths linking them. It behaves as if it were a single node. It appears in a model as a single node, unless the user chooses to open a screen window showing the internal structure and its animated behaviour (Figure I.3). With the capability of creating composite nodes, the following benefits can be got:

Hierarchical Presentation. Any system can be represented in a hierarchy, consisting of subsystems. It is usually easier to deal with subsystems, especially when the system is complex, e.g., Manufacturing system [120]. Composite nodes provide the facility to arrange objects in the simulation model in hierarchy, presenting the real situation.

Easier Model Development. Both top-down and bottom-up development can be used. A concept model can be built and tested quickly then the model can be replaced by composite nodes and developed in detail. Composite nodes can be treated like other simple nodes. They can be stored in a library and re-used in the same or different models. Therefore, parts of the simulation can be developed and tested in isolation and then combined in a more complex simulation.

I.4 Knowledge Presentation

STEM is based on the LOOPS and LISP environment. LOOPS integrates several programming paradigms, such as object-oriented programming, procedure-oriented programming, access-oriented programming and rule-oriented programming. With object-oriented programming and rule-based programming, LOOPS allows the embedding of expert or knowledge-based system in a simulation and is the simplest and most intuitive to represent simulation knowledge and behaviour [211].

Each object in a model has its own data and behavioural rules. These rules are usually presented in an IF-THEN scheme and can be defined as methods or rule sets. They provide a convenient way to describe a flexible response to a wide range of events, make the control knowledge of objects easy to understand. They can be easily modified as well.
I.5 Interactive Program Execution

The most tedious work, and perhaps requiring most of the model building time when using conventional simulation is the debugging of the simulation program. Usually the time to develop and verify the models has taken so long that the design teams are delayed waiting for simulation results [177].

STEM, however, provides the facility to help quickly debug the simulation programme. Besides the run time animation which can help debugging in the first place. STEM provides a break package for debugging, which will record the dynamic status of the objects in the model and provide clue for inspection. This inspection of expected object attributes helps determine the behaviour of the object. STEM, on the other hand, allows you to suspend and continue or stop running the simulation at any time.

I.6 Graphic Capability

STEM provides powerful graphic capabilities for model building, editing and run-time animation.

Creating a simulation model in STEM, the first step is to identify the points in the process to be simulated and to choose initial STEM nodes to match each of the process points, and then to position these nodes on the map and connect the nodes by data paths. Therefore, the model can be physically shown on the screen, providing clarity.

The simulation model can be run with the animation on or off. Animation enhances the credibility of simulation models and improves the utility of simulation results [240] [216].

From a practical viewpoint, animation capability has the following advantages:
--It can be used to convince the engineers or managers about the validity and usefulness of the model.
--It also provides an excellent mechanism for obtaining feedback from personnel who understand the problem or system under study, but do not have any detailed understanding about the use of simulation techniques.
--It can help the modeller recognize the problems in the simulation model when the model is first created and tested.
--The user, while the model is running, is presented with a dynamic display, and can modify the value of attributes and immediately see the results of these changes. Thus, for a manufacturing system simulation model, the engineer becomes part of the simulation process and can learn and master the model quickly.
I.7 Output Analysis

The purpose of building a simulation model is to identify problems in a system and evaluate different scenarios by analysing the simulation results [250]. There are two entities that change during a simulation run in STEM and are therefore, suitable for monitoring and for producing results. These are the node attributes and the simulation events.

In STEM, monitoring is accomplished by attaching monitors to attributes or events. According to different requests, there are several types of monitors available. All monitors have an iconic form on the simulation map and will be saved and retrieved with the simulation. One type of the monitors is called a logger. It accepts multiple inputs and collect them in a time ordered manner when the run is complete or a discontinuity in time is selected, the data may be processed or stored.

Animation and dynamic monitors are used during development of the simulation to produce the 'best' model. However, for serious results analysis, the post-analysis is needed [210].

I.8 Time Manager

STEM uses the discrete event simulation approach, which manages a calender of events [37]. A calender is a set of event notices ordered by execution time [164]. It provides a time management object, so providing the minimum necessary in a simulation environment. Running the simulation model simply means executing the next event queued in the calender until there are no more events or until some criteria to stop is encountered.

At any time, there may be one or more activities happening. Once all activity is completed the event is removed from the event queue and the next event is processed. This time manager makes the model building much easier. The status window can display the time and the events happening at that time, making understanding of simulation much easier.
Figure 1.1

Basic STEM Concept

Token movement along path

Sending Node  Token  Token  Receiving Node

Figure 1.2

A STEM Example
Figure 1.3

A Composite STEM Example
II.1 Introduction

This appendix introduces the STEM operating environment based on the knowledge discussed in Appendix I. The main aspects covered involve the language manipulation environment, menu operating system and other facilities in STEM.

II.2 Operating Environment

STEM is an object-oriented package, based on LOOPS and LISP. It can be run on a Xerox workstation or a SUN workstation. Figure II.1 shows the interaction between the major components of STEM.

II.2.1 LOOPS

The programming system running in LISP is called Interlisp-D [10] [11] [12]. It consists of a programming language and a large number of pre-defined functions using LISP terminology. The language and pre-defined functions of Interlisp-D are rich, but similar to those of other modern programming languages [3]. The Interlisp-D programming environment, on the other hand, also provides a completely self-contained world for creating, debugging and manipulating Interlisp-D programmes. In addition to these basic programming tools, Interlisp-D also provides a wide variety of programming support mechanisms, such as a list structure editor, break package, record/datatypes package, file package, masterscope, windows and the inspector.

LOOPS [9] developed at Xerox and based on Interlisp-D, is an integrated knowledge engineering language, which integrates several programming paradigms to facilitate the design of AI applications. The major characteristics of LOOPS are the integration of four different programming schemes to provide a powerful environment for knowledge system building.

Object-Oriented Programming. In this paradigm, the information is organised in terms of objects. The objects with the same features and behaviour [243] are built into a class, and the classes are arranged in hierarchy which allows complex objects to be simply described. Objects in the system can interact by 'sending messages' to other objects, causing the methods (functions) associated with an object to be executed. These methods may
include sending messages to other objects. This feature results in significant flexibility and extensibility of models [243]. The detailed discussion of object-oriented programming can be found in Appendix III.

**Procedure-Oriented Programming.** This paradigm is the most widely used language system. The large procedures are built from smaller subroutines. In this environment, the data and instructions are kept separate. Interlisp-D itself plays this part in the LOOPS environment.

**Access-Oriented Programming.** In this paradigm, accessing a value can trigger an action. This is done by placing active values on the object variables and probing additional computations when data is changed or read [274]. It is very useful to monitor certain values.

**Rule-Oriented Programming.** AI-based simulation languages and models need to have the capability to change the system based upon state variables and to dynamically modify the flow of entities through the system [185]. The capability entails the use of pattern-directed inference systems and production rules. Rules, in this paradigm are organised into rule-sets which specify the rules, a control structure and other descriptions of rules. These rules provide a convenient way to describe flexible responses to a wide range of events.

### II.2.2 Libraries

**STEM** is designed for general application. These standard object classes are used to supply a generic range of functionality. The general standard object classes of **STEM** are listed in Figure II.2. They are built based on LOOPS.

For the specific applications, **STEM** allows the user to modify the behaviour and nomenclature of standard classes to suit particular application needs by specialising these classes or creating new classes from scratch. Based on these standard classes stored in **STEM** library, the simulation libraries contain these specialisations which provide the specific application area.

### II.2.3 Monitoring

Monitors in **STEM** are designed to track the variables of the objects and events during the simulation. They all have an iconic form on the simulation map and will be saved and retrieved with the simulation. There are two main types of monitor: single attribute monitor and multiple attribute monitor.
Single Attribute Monitor. They can only be attached to a single variable on a single node on both root map and the composite node map. There are four main types of single attribute monitor: average monitor, trend monitor, queue monitor and text monitor.

Multiple Attribute Monitor. In contrast to single attribute monitors, the multiple attribute monitors can monitor more than one attribute of the nodes to which they are attached and/or they can monitor events. There are three types of multiple attribute monitors: quick tracer, logger and status window.

Animation and dynamic monitors are used during development of the simulation to produce the "best" model. However, for serious analysis, post processing on a large number of points is needed [243]. STEM provides this via a specialised logger monitor and analyser.

A logger can be added to the root map or any of the composite node maps. Both variables and events can be logged. Having logged the data, STEM provides methods for displaying the data and inserting it into the document. The analysis is made by creating an analyser node selected from the title menu of the root map.

In addition to analysing node attributes, the model can also be analysed by seeing which events are caused by which. A caused event browse is used to build up and display a hierarchical structure of events.

II.3 Menu Operating System

STEM provides powerful facilities for building, running, and changing simulations and for collecting results by extensive use of menus, three-button mouse, multiple windows and interactive graphics.

In STEM, some of the menus may be permanent, while other menus are "pop up" when a selection is made it disappears. But they all operate in a similar manner. Some options available on the menu change according to the states of the system.

There are five main menus in STEM, the global menu, the map title menu, the text menu, the controller menu and the object menu. With the exception of the global menu, all of the menus refer to a particular simulation.

The global menu is the central control point for using STEM. Options on this menu give the user access to all existing simulations and their files.

The map title menu is used to add objects to the simulation and manipulate the simulation map. CNode map title menus have slightly different options because they also refer to the CNode editing behaviour.
The edit menu provides options for editing those parts of the simulation which are
not accessible through the graphic map interface. This includes saving a simulation and
editing of the text associated with a simulation.

The controller menu provides options for running and modifying a simulation.

The object menu provides specific options for manipulating simulation objects on the
map. There are also generic class options for changing the general characteristics of the
class of objects.

II.4 Other Facilities

Some of the options in the STEM menus give the user direct access to other system
packages, so that these facilities are available within STEM.

The inspector package is selected for inspecting and editing individual objects and
the structure editor for examining source code of functions and gate handles and for
debugging during simulation. The LOOPS browsers can be used for seeing the class
inheritance lattices and the break package for debugging and the support of break windows.

II.5 Modelling Logic

Most simulation studies are concerned with a system's performance over a period of
time. The method used for time keeping is therefore an important issue in simulation system
design [239]. Timekeeping in a simulation has two aspects: that of advancing time or
updating the time status of the system and that of providing synchronization of various
elements and occurrence of events. Two basic timekeeping mechanisms are usually used:
fixed time increment which is used by continuous change models and variable time
increment by discrete change models. Simulation applied in manufacturing industry implies
the use of the later type: discrete event modelling [239] [144] [139].

II.5.1 Discrete Event Simulation

A discrete event simulation is one which employs variable time increment approach
to control the behaviour of the model [191]. It views models as structured collections of
objects bound into webs of relations and transformations [169].

For any discrete event simulation model, there exist five components in the system
[144]: model structure, time manager, random process environment, statistical output
environment, and simulation executive. The relation between them is listed in Figure II.3.
The simulation structure environment is the facility for model structuring and execution. The most natural view is that of a collection of interacting objects moving through sequences of actions [144].

Any dynamic models need a clock from time management. Discrete event models typically use 'next event' strategy for advancing model time [243] [25]. Here the clock's value is advanced to the time of the next event on the monitor's agenda. Once it has been established that no further actions can be performed at the current time instance.

Random processes are used to represent less essential aspects of the simulation model, otherwise all interesting phenomena are too complex to be modelled in all their details [144].

One objective of the simulation is to analyse the system performance through the simulation results analysis. Facilities for statistical instrumentation and reporting are therefore essential. They must provide for means, minimum, maximum, frequency tables and time plots.

A simulation executive is responsible for sequencing state transitions. It usually needs an event list to keep track of pending events.

Three components, clock, distribution, and data collection devices are the basic elements in a simulation system. Nearly all simulation packages can provide these facilities [3] [211] [80], but by using different languages and approaches.

In the environment for model structuring and execution, entities are used to present the elements of the system being simulated and can be individually identified and processed. Actions leading to state transitions are typically described through functions and procedures and their dynamic interactions [144]. There are two ways to organise these entities and their actions. One is to employ the object-oriented programming paradigm. Here, a model must encapsulate all relevant aspects of an entity; its attributes, actions, and life cycle. Communication between objects is allowed only through well-defined interfaces, provided by those messages an object is willing to respond to (refer to Appendix III). The other is to use the procedural programming paradigm, in which the procedure is composed of small procedures.

Sequencing state transitions is the responsibility of a simulation executive. The event lists used to keep tracks of pending events can be organised in a number of ways. Most event list implementations are designed so that the next imminent event can very quickly be determined and removed from the event list. The sequencing strategy used by a simulation execution depends on the way descriptions of a model's behaviour are modularised.
Activity, event, process and three phase based world views may be distinguished [3] [32]. The executive based on each view can be found in Figures II.4, II.5, II.6 and II.7. The detailed discussion can be found in [144] [239].

II.5.2 Modelling Logic in STEM

STEM belongs to event based modelling, in which two basic classes have been defined, EVENT and EVENT-BROWSER. EVENT class describes events which may occur. An event happens when the specified message is sent to the specified object so it can be viewed as a delayed message sent. EVENT-BROWSER is a specialised form of browser which allows the browsing of events.

An event is an object which contains related information, such as datum, gate, time, destination etc. Thus at each step of modelling, the clock is advanced to the time of the event at the head of the event queue. This event sends messages to all objects which it affects, which respond according to their defined methods, and which may issue messages to further objects, or add new events to the queue [196].

The inference engine of the knowledge based simulation environment is embedded in STEM. The execution of events on the event list is manipulated by this inference engine defined in a class called SIMULATION-CONTROLLER.
Figure 11.1 STEM Environment
Main Components in Discrete Event Simulation
Appendix III
OBJECT-ORIENTED PROGRAMMING TECHNIQUES

III.1 Introduction

This appendix introduces the object-oriented programming paradigm applied in LISP. It is aimed at helping interested readers and simulation model users use this technique effectively.

III.2 Object-Oriented Programming

In an object oriented programming style, an object is defined as a symbol associated with a unique database of properties and operations which represent the object [281]. Objects communicate with each other by the passing of messages, which define a protocol of all the requests an object will responded to. Other data structures, functions, procedures are private and can only be used from within an object [144].

One of the characteristics of object-oriented programming is the use of some inheritance mechanism. Objects are organised into hierarchical classes in which subclasses inherit properties and message patterns from their superclasses. Hierarchical structures are most commonly supported, where classes can be further specialised into subclasses, sub-subclasses and so on [144].

Another feature of object-oriented programming is that messages specify only which operation should be performed, not how the operation should be performed [281]. This means the principle of strong localization of information in closed modules is strictly adhered to [145] [243].

Object-oriented programming paradigms are becoming attractive vehicles for constructing simulation models of real world systems [172]. The popularity of object-oriented programming stems form the close relationship between entities in the simulation world, and program objects [196]

Complicated models in object-oriented paradigms can be built rapidly and reliably with the abstraction capability [105]. This provides enough flexibility so that the user can specify the behaviour of individual instances [187] [25]. The following section will introduce the object-oriented programming technique in the LOOPS environment.
An object is usually declared in a piece of structured code which enumerates the attributes of the object class: Instance Variables (IVs), Class Variables (CVs), methods and relations with other classes (the supers and meta-classes to which it relates). The following is an example which shows the specification of the object LCD in the LOOPS environment [14].

(Class LCD

((Mate Classes Class doc (* A gauge which gives an alphanumeric display of a Loops value))
(Supers Gauge etc.)
(Class Variables
(Font (Gacha 12 Bold) doc (* Font used to display readings on all class numbers)))
(Instance Variables
(reading 0 doc (* value shown on gauge))
(height 14 doc (* height of gauge))
(width 30 doc (* width of gauge))
(readingY 7 doc (* y position of centre of reading))
(precision 4 doc (* precision is the number of characters in the reading))
(readingRegion NIL doc (* reading region is area which holds reading)))
(ComputeScaleLCD. ComputeScale
args (min max)
(SetLCD. Set
args (reading)
doc (* set value of reading and display it))
(SetParametersLCD. SetParameters
args NIL
doc (* set the font for the window ))))))

Every member of the class may have a different value for an instance variable, such as the reading instance variable. However each instance of the class has the same class variable, like Font in the class variable.

In any traditional programming language, procedural code operating over data structures is applied. For example, in modelling the behaviour of a logic gate, a set of procedures would be defined which would operate over some data structure (a state vector) which represents a particular gate.

In an object oriented system, the set of procedures applicable to a particular state vector are associated with that state vector. The conventional call and return function protocol is replaced by a slightly different "message send and return" protocol. Here a
message is sent to an object consisting of a selector identifying what the user wants to take place and message arguments. Thus, in LOOPS the behaviour to compute the state of a gate in a logic model would be defined separately for each type of gate and it would be involved in sending a message to the relevant gate. In LOOPS code, this would be:

(- gate ComputeState).

Thus an immediate advantage of object oriented programming is the reduction in the number of decision points present in the code. In an object oriented system, code is located in a manner that clearly associates it with the data structure over which the code is intended to operate [145]. The functional system does not need to be changed when new types of gate are introduced or when the data representations are altered. This makes the system much easier to modify and thus supports a prototyping approach to software development.

The behaviour of objects are specified by the functions that are invoked as a result of the message sent. Objects of different types may use different functions in response to the same message. For example

(- gate ComputeState)

could call on one of the following:

OR.ComputeState
AND.ComputeState

depending on whether the gate is an OR gate or an AND gate.

Objects of the same type will always use the same functions in response to the same selector but they may have different data over which the function operates. For example, two OR gates could have different inputs so (- gate ComputeState) could result in one setting its output high and another setting its output low.

Although the data values on different objects of similar type may vary, the actual data structure needs to be identical otherwise the functions applicable to that type of object might attempt to access non-existent data. For example, all gates have inputs and outputs together with a state, although the individual values on particular gates may vary.
Appendix IV
SIMULATION MODEL EXECUTION SPECIFICATION

For efficient and accurate use of the simulation model, it is of importance to describe the instructions on the model running process clearly. The user should follow the steps described in order to use the simulation model correctly. The next section discusses the step-by-step description. The simulation model is run in the LISP environment. Therefore, the LISP environment on the SUN or Xerox workstation should be implemented and be ready to be used before starting the simulation execution.

1. Connect the directory to {DSK}<LISPFILES>STEM>jh>PROJECT> on the Xerox or {DSK}<home/jelly/hmsm/jiao/project> on the SUN.

2. Load the class library files. HMS-SYSTEM-START.THE is loaded on the Xerox and HMS-SYSTEM-START.STEM-USER-LIBRARY on the SUN. This will load all the related class library files defined in the simulation model. It usually takes several minutes. The files include:
   - HMS-SYSTEM-PRINTINTERFACE.STEM-USER-LIBRARY
   - HMS-SYSTEM-CELLS.STEM-USER-LIBRARY
   - HMS-SYSTEM-MACHINES.STEM-USER-LIBRARY
   - HMS-SYSTEM-SHOPS.STEM-USER-LIBRARY
   - HMS-SYSTEM-MANUFACTURE.STEM-USER-LIBRARY
   - HMS-SYSTEM-PROPLANNING.STEM-USER-LIBRARY
   - HMS-SYSTEM-PRODUCTION.STEM-USER-LIBRARY
   - HMS-SYSTEM-ENGINEERING.STEM-USER-LIBRARY
   - HMS-SYSTEM-SALES.STEM-USER-LIBRARY
   - HMS-SYSTEM.STEM-USER-LIBRARY.

3. Load the simulation file.
   --Lclick on the global menu choose STEM simulation to get a pop-up menu, choose simulation and select HMS-COMPANY from the menu.
   --Place the window and click the left button.

4. Display the Controller menu for HMS-COMPANY.
   --Lclick the title bar of root map of HMS-COMPANY choose Display control menu from the STEM Title Menu.
   --Position the Controller menu.

296
5. Enter input data and rules if required.
--L-click the HMS-COMPANY options menu choose Create Company Menu. This opens the Company Model Master Menu.
--Select Model Specification from the Company Model Master Menu to read the instructions carefully.
--Select Data Entry Menu from the Company Model Master Menu to open Company Model Data Entry Menu.
--Select Help from the Company Data Entry Menu to look at the input data explanation.
--Enter related data by selecting different items from the menu.
--Select Model Master Menu from the Company Model Data Entry Menu to get the Company Model Master Menu.
--Select Rules Entry Menu from the Company Model Master Menu and enter the rules by using the same method described as entering the data.

6. Record the input data and rules information into a file.
--Select Model Master Menu from the Company Model Rules Entry Menu.
--Select Hardcopy Input Data. This will save all of the input data and rules into a file: HMS-SYSTEM.INPUT.

7. Run the simulation. STEM not only provides facilities for running a simulation, but also facilities for determining some of the characteristics of the run, either before the run starts or during a run. All of these facilities are available through the Controller menu [14].
--Ensure that the Controller menu for HMS-COMPANY is displayed.
--Select Initialise Simulation from the Controller menu.
--Select Animation on from the Controller menu.
--Select Run Simulation from the Controller menu.
--To slow down the animation Select Set animation interval from the Controller menu.
--Select Suspend Simulation from the Controller menu if the user wants to stop the simulation run and look at the results and then continue to run the simulation.
--Select Model Master Menu from the Company Model Rules Entry Menu.
--Select Output Data Menu from the Company Model Master Menu to open the Company Model Output Menu.
--Select different items from the menu to look at different output data.
--Select Continue run from the Controller menu.
--Repeat several times until satisfactory outputs have been obtained.
8. Record the output data.
   --Select Suspend simulation from the Controller menu.
   --Select Hardcopy Output Data from the Company Model Master Menu to save all the output data into a file: HMS-SYSTEM.OUTPUT.

9. Stop the simulation run and restart if required.
   --Select Suspend simulation from the Controller menu.
   --After step 8 Select Reset simulation from the Controller menu.

10. Monitor the simulation. There are several monitors to choose from to monitor the node attributes and the simulation events. The detail description on these monitors can be found in Appendix II. The next section describes the steps taken in using a single-attribute monitor and a multi-attribute monitor. It is recommended that this step is taken prior to step 7: run the simulation.

    Attach an AVERAGE-MONITOR to the numberininputstore attribute of the s1-input.
    --Drag the mouse through Create node and select Monitor from the HMS-SHOP! title menu.
    --Drag the mouse through *NUMERIC-ATTRIBUTE-MONITOR* and select AVERAGE-MONITOR.
    --Place the monitor on the map.
    --Select Attach from the AVERAGE-MONITOR attribute menu.
    --Lclick on the S1-input and select the attribute numberininputstore.

    Attach the logger to the numberininputstore attribute of the inputstore1.
    It should be realised that the logger can be used to monitor more than one attribute.
    --Drag the mouse through Create node and select Monitor from the root map HMS-COMPANY title menu.
    --Select logger and place it on the root map without naming it.
    --The logger can log attributes of any node, including those within a cNode.
    Left click on the LOGGER and select Attach to Attribute.
    --Lclick on the HMS-COMPANY node, you are now given the option of selecting attributes of the cNodes by left clicking or components of the cNode by right clicking.
    --Right click and select COMPANY cNode from the root map.
    --Right click and select PRODUCTION cNode on the HMS-COMPANY map.
    --Repeat the same procedure several times until the HMS-SHOP1 cNode has been selected.
    --Right click and select s1-input node from HMS-SHOP1 cNode map.
--Select number in input store from the attribute list.
--Lclick on the logger and select Switch on, the eyes of the elephant will
now open showing that the logger is recording data.
--Run the simulation for a while, then select Switch off.
--Reset the simulation, this closes the logged files. You are prompted to enter
a text string that is used to identify what the log was recorded for.

It should be realised that the simulation output results which are obtained through the
model defined interface Company Model Menu can only be recorded after suspending and
before resetting the simulation. Therefore the sequence of recording simulation output
results is to save output results onto a file and reset the simulation to get the logged data
through the analyser.

11. Present the logged results. Having logged the data, STEM provides methods for
displaying the data and inserting it into a TEdit document [14].
--Drag the mouse through Create node and select Create Analyser.
--Place it on the root map without naming it.
--Lclick on the analyser node select Select log from ANALYSER
options and choose HMS-SYSTEM.STEM-LOG-INDEX.
--Select plot data from the ANALYSER options to look at the results
and get various plots on these output data.
Appendix V
SPECIFICATIONS ON CNODE MODIFICATION

V.1 Introduction

In the simulation system, the structural layout of the model has been set up to provide a generic manufacturing organization modelling system. However, it should be considered that the model must provide the capability to allow the user to change the layout by adding some process points which are important to him/her or modifying the model. Therefore it is necessary to offer the user specifications for cNode modification. The following sections focus on describing the instructions for several occasions.

V.2 Add a Simple Node on a CNode

Adding another simple node on a cNode map, the first task is to choose the right node type which matches the process point of interested and place it in the right place. To add a new machine in the DETAILED LAYOUT cNode for example, the step-by-step instructions are as follows:

1. Open cNode HMS-MACHINES.
2. Create the new machine node.
   --Select 'Create node' from the STEM Title Menu and choose HMS-MACHINE from options provided.
   --Enter new-machine in the prompt window and place the node at the top of the map.
3. Connect the new-machine with s1-ms and transporter node.
   --Lclick (Left click) the s1-ms node.
   --Select 'Add Path' from s1-ms options.
   --Lclick new-machine stage process.
   --The path does not require a name, press return when prompted for the path name.
   --By using the same way, the connection between new-machine and transporter (from new-machine to transporter) and the connection between transporter to new-machine (from transporter to new-machine) can be done.
4. Change the image of new-machine if required by copying the machine image or creating a new one.
5. Update the cNode map: HMS-MACHINES.
   --Select 'Update CNODE class' menu from the STEM Title Menu.
6. Replace the old cNode with the newly updated one.
--Lclick the DETAILED LAYOUT cNode in the HMS-SHOP1 window.
--Select 'Edit class' from class options from HMS-MACHINES options.
--Change the input and output gate value if required in the class variable
gateDescriptions, in order to position the cNode map at the right position.
--Lclick the DETAILED LAYOUT cNode in HMS-SHOP1 window.
--Select 'Replace node' from the HMS-MACHINES options and choose the same
cNode: HMS-MACHINES.
--Open DETAILED LAYOUT cNode again, the newly defined HMS-MACHINES
cNode should be displayed.
7. Update all the other related cNodes in a sequence: HMS-SHOP1,
HMS-MANUFACTURE, HMS-PRODUCTION, HMS-COMPANY till the
root map of HMS-COMPANY.
8. Save the related files and simulation.
--Select 'Save this simulation' from the HMS-COMPANY edit menu. This will
save all the related files and the simulation file.

The newly modified layout of the detailed machine level is shown in Figure V.1.

V.3 Add a CNode on a CNode

The principle theory of adding a cNode on to a cNode map is similar to that of adding
a simple node on a cNode. The only difference is that a cNode is originally constructed
from a simple node. All the input and output gates required by the new cNode should be
added to that simple node. cNode only changes the internal relation between nodes contained
in the cNode. In this section, suppose that the newly added cNode is connected to two simple
nodes on each side. The next section discusses the steps for changing cNode external links.

The example adds a new shop to the MANUFACTURE cNode. If the steps used are
the same as those in section V.2, they are not repeated. The descriptions are as follows:
1. Open HMS-COMPANY cNode map of: HMS-MANUFACTURE.
2. Create a router process node on the HMS-MANUFACTURE cNode in the same
way as described in step 2 in section V.2 with the name new-shop.
3. Connect JOB-SCHEDULER node with new-shop node, new-shop node with
transporter, and transporter with new-shop.
4. Change the simple node into a cNode.
--Select 'Create cNode class' from HMS-MANUFACTURE Title menu.
--Draw the box well around the "new-shop", do not include any other nodes.
--Enter HMS-NEW-SHOP when prompted for the class name.
--Enter HMS-SYSTEM-SHOPS when prompted for the file name.
5. Edit the cNode of new-shop.
   --Delete the path between router process and the output tag.
   --Replace the router process node with HMS-INPUTSTORES.
   --Add HMS-WORKAREAS router process node, HMS-OUTPUTSTORES, and STAGE node.
   --Connect these nodes in the same linking relationship as in other shops.
6. Update the newly defined cNode class: NEW-SHOP in the same way as introduced in step 5 in section V.2.
7. Replace the old cNode NEW-SHOP with the updated cNode class (Refer to step 6 in section V.2).
8. Update all the related cNodes, HMS-MANUFACTURE, HMS-PRODUCTION, HMS-BUSINESS, till the root map of HMS-BUSINESS.
9. Save the related class library files and simulation file.

The newly modified HMS-MANUFACTURE and HMS-NEW-SHOP cNode maps are shown in Figure V.2 and Figure V.3.

V.4 Modify a CNode

It often happens that a defined model needs to be modified to meet the user’s special requirements and to offer wide application areas. Therefore, the simulation environment should provide such a capability. This section describes the steps in detail used in the modification process in order to provide guidance to the user for efficient and accurate use for his/her simulation model.

An example is used to present a step-by-step description of how to modify the cNode HMS-COMPANY by adding a simple node "finance" into the cNode. This node is connected with SALES and PRODUCTION cNodes.

1. Open HMS-COMPANY cNode map of: HMS-COMPANY.
2. Create a router process node on the HMS-COMPANY cNode in the same way described in step 2 in section V.2 with the name finance.
3. Open HMS-COMPANY cNode map of: HMS-SALES.
4. Create a tag on the HMS-SALES cNode.
   --Select 'Create node' from the STEM Title menu and choose Create tag from options provided.
   --Link the tag with the output gate of PO-SENDER node when prompted.
   --Update the cNode map of: HMS-SALES (Refer to step 5 in section V.2).
--- Replace HMS-SALES cNode with newly modified class.
5. Link SALES cNode and finance node on the HMS-COMPANY cNode map.
6. Link finance node with PRODUCTION cNode in the same way as described in steps 3 and 4 above.
7. Update HMS-PRODUCTION cNode.
8. Replace PRODUCTION cNode on the HMS-COMPANY with the newly modified HMS-PRODUCTION cNode.
10. Replace COMPANY cNode on the root map of HMS-COMPANY with the newly defined HMS-COMPANY cNode.
11. Save all the related class library files and simulation file.

The newly modified cNodes are presented in Figures V.4, V.5, and V.6.
Figure V.1

Modified CNode Map of HMS-MANUFACTURE
Figure V.2

Modified CNode Map of HMS-MANUFACTURE
Figure V.3
Modified CNode Map of new-shop

Figure V.4
Modified CNode Map of HMS-COMPANY
Figure V.5
Modified CNode Map of HMS-SALES

Figure V.6
Modified CNode Map of HMS-PRODUCTION
Appendix VI
GEC ALSTHOM LARGE MACHINES LTD

VI.1 Introduction

A company, GEC ALSTHOM Large Machines Ltd. is introduced in this appendix as an industrial case study for the preliminary test. The system configuration of the company is first described. Then the operation system of the company in hierarchy is stated. Finally the products and the key cells involved are discussed.

VI.2 System Configuration

The company is a separate trading entity within the recently established Anglo-French GEC ALSTHOM Group and is a wholly owned subsidiary of GEC ALSTHOM. It is a world wide market leader in its field. Operating from a single site in Rugby with 1248 employees, the company has recently achieved a sales turnover of £45M and is profitable.

The organization chart for the company is shown in Figure VI.1. This diagram represents the organization structure of the company. The company is in nature organised in a hierarchical structure, containing company level, cell level, and machine level which is similar to that of other companies [162]. The term cell is used in the company to represent shop in the other companies. The main activity at the company level is to manage the production properly and efficiently. The cells are the basic manufacturing units which are autonomous and self sufficient. Machines carry out the real manufacturing operations on the shop floor.

VI.3 Operation of the Company

As a separate trading unit, the company is managed under its own operating system. The emphasis is laid on the company level and cell level.

VI.3.1 Company Level

The company level is the highest level in the management of the company. GEC ALSTHOM Large Machines Ltd. contains several functional areas, including sales and
marketing department, engineering department, estimating department, production control
department, and purchasing department. The general structure of the company level is shown
in Figure VI.2.

The production of the company is Make-To-Order production type [155]. Upon
receiving a customer order by the sales and marketing department, the estimating department
will provide costing guidelines on an overall contract and provide estimated working hours
on 'cost centres'. When an order is placed, the contract is then passed to the product con­trollers in the engineering department who are responsible for small, medium, large and
spares motor groups. They either use standard templates or produce new ones based on
previous orders.

The design engineering department works on an order enquiry or a final contract order.
The new design or modified design is made according to customer specifications. When
these drawings are finished, they are released to the planning department where process
plans are generated.

The production control department manages the production on the shop floor both
day to day activity and the long term operation. This department is responsible for the
control to ship the right materials to the right place at the right time. Based on the drawings
from the drawing office in the engineering department and the process plans from the
production plan department, the materials are ordered by the purchasing department based
on the information provided by stock control. Meanwhile the work tags are produced and
some internal items are manufactured. The finished products will be tested to meet the
customer’s requirements and design specifications.

The purchasing and stock function operated at the shop level, act as a cell to purchase
materials through the subcontract under the control of material requirement planning within
the production control department and to store the materials or completed products.

VI.3.2 Cell Level

In GEC ALSTHOM Large Machines Ltd, the manufacturing system is organised in
cellular structure. The cell is defined as the basic processing unit in the company. The term
used here is different from the cell concept defined in Group Technology. Cells in GEC
ALSTHOM actually act like shops in general concept. However the company is planning
to move the company towards the development of Group Technology.

Each cell is here viewed as a separate and a self contained functional manufacturing
unit, having its own supervisor, processing people, production engineer, process planning,
work study and foremen.
The company is composed of 13 cells, namely fabrication, large machining, medium machining, press, core build, AC coils, DC coils, field coil and commutator, winding, large assembly, medium assembly, test and pack & transport. The manufacturing connection of these 13 cells is shown in Figure VI.3.

The components for final products usually pass through two cells. It is produced as the completed item from the first cell and is the required item of other components in the second cell.

**VI.4 Products and Production Procedures**

The company mainly produces cage and wound rotor induction motors over 100 kw at 1500 RPM; all salient pole AC synchronous motors and generators; AC synchronous induction motors and all other types of AC generators; all DC motors and generators over 75 kw at 100 RPM including flame-proof machines and variable frequency large AC machines. The company is responsible for the design, manufacture and marketing of these products.

In these 13 cells, three main lines are organised to produce three types of machines, namely large machine, medium machine and small machine, mainly depending on the product size. Although these products are produced based on the customer’s specification, the major procedure for the same type of machines remains the same. Figure VI.4 to Figure VI.6 present the production network to show the processing procedure for small, medium and large machines. The detailed description of key cells can be found in the following section.

**VI.5 Key Cells**

The major function and the layout of the key cells of the company are described in this section. These cells involve: fabrication cell, press cell, machine cell, field coil and commutator, and final assembly cell.

**Fabrication Cell.** The fabrication cell may be divided into two areas: cutting and assembly. In the cutting area, raw materials, usually plates, are cut to produce detailed parts; the assembly area joins them together.

The cutting area consists of shearing machines and three flamecutters for which nesting patterns, for the parts to be cut, must be produced and converted into instructions for machines. After cutting the parts usually require flattening due to the bending of insufficient
support for the parts and the material not being flat when supplied or warping during flamecutting. Hence a large proportion of jobs could bypass the flattening operation thus reducing delays due to queueing before the process and eliminating the process time itself.

The assembly area welds the details together, heat treats the assembly to relieve any stress concentrations, shot blasts it, trims where necessary and tests for any defects in the welding if called for on the work tag.

Press Cell. The press cell supplies laminations for stators, rotors and poles. All items produced originate from sheet metal. The reels are sheared into squares before blanked into a disk or a segment. If the machine, for which the parts are being pressed is too large for the presses, laminations must be made from several segments rather than a single disk. Disc production is preferable from a manufacturing viewpoint. Each disc is pressed to produce two concentric discs, one forms the rotor, the other the stator. Each is then notched to yield the gaps which are to be filled later with windings which will carry an electric current.

Machine Cell. The general processing operations in this cell include milling, drilling, turning, trimming and boring. In medium and large machining cells, the majority of machining operations are performed. The categorisation of large and medium relates to the size of the jobs which are handled by the cells rather than the classification of the final product to which the parts belong.

Field Coil and Commutator Cell. This cell functions as two independent areas only considered as one because both areas are small.

Field Coils. Three types of poles are produced in this cell, main poles, compoles and wound poles. The main poles are fabricated from strips of copper which have jigsaw shapes punched from each end. The strips can then be fitted together forming a square spiral ready to be brazed. After cleaning the coil is insulated and pressed to form a rigid field coil. Compoles are placed in between the main poles to affect the magnetic field generated by the motor. They are produced from a single strip of copper wound into a rectangular spiral. Again the poles are insulated and pressed. Enamelled copper wire is wrapped around forming tools to create wound poles. They are then dipped into varnish and baked in ovens.

Commutators. Commutators are made from several risers. Two copper bars brazed together at a right angle are one riser. The risers are sandwiched between pieces of insulation mica in a jig. The circular jig is then tightened to press the sandwich together and to form the diameter required for the commutator. The insulation glass matting is cured in an oven thus setting the risers in position. Control discs hold the risers more firmly in place. The appropriate bore diameter can then be machined.
Assembly. The assembly cell receives major items from other cells on or after their due dates. The detailed parts are obtained from the stores by submitting a picking list. This list identifies the items required and their location on the racks. The stores put the kits together ready for assembly. Subassemblies plus detailed parts then form the main assembly. After final assembly the machine is released to the test cell. If passed, it returns to the shipping area where the packing slip is produced and sent with the machine to the Dispatch and Transport Cell.

VI.6 Company Re-Organization

The company now plans to re-organise the manufacturing system in order to be suitable, in both the short and long terms to carry the company forward with effectiveness and efficiency in manufacturing and competitiveness in the market place. The most benefit to be obtained from this plan will be the better work flow attained from the improved factory layout. At the same time, work in progress should be reduced because of the high capital tied up in the business and the limited available space.

For this newly organised factory, it is needed to implement the Just-In-Time manufacturing philosophy and to change its production management system. The new production control systems will be integrated into the business in the manner shown in Figure VI.7. When the Commercial Department receives a sales enquiry an estimation for labour and materials content will be made. Attempts will be made to accommodate the possible order in the Master Production Schedule. This Master Production Plan will be verified by a Rough Cut Capacity Planning tool which will take and use data from the estimate. After an order is placed, material, internal works and sub-contracting orders will be produced by the Manufacturing Resource Planning System. Detailed scheduling of work will be carried out at the cell level to enable delivery to the customer by the required date.

The other scope of this new plan is to provide quicker deliveries, to improve the operating efficiency, to improve the quality and to reduce the costs and the overheads. The detail is shown in Figure VI.8. The physical re-organisation of the company is aimed at improving the work flow by putting successive operations next to each other, arranging the plant and equipment together in the right sub-cells and arranging the machines into the correct places. Figure VI.9 shows the cells arrangement in the company.

Besides these production control systems and physical re-organization, the computer integration is another target of the company. The functional departments and all these 13 cells will be all linked to the control centre to implement the total integration of the computer system which is shown in Figure VI.10.
Figure VI.1
Organizational Structure
Figure VI.2
Operational Structure of the Company Level

Figure VI.3
Cellular Organization Structure
I PRESS \( S_n \)

FABRICATION

I M.M. CORE

ASSEMBLY

I M.M. WINDING

A.C. COIL

M.M. FINAL

ASSEMBLY

M.M. TEST

PAINTERS

SHIPPING

Figure VI.4
Production Network of Small Size Machine

FABRICATION

M.M. MC

SHOP

M.M. CORE

ASSEMBLY

M.M. WINDING

A.C. COIL

D.C. COIL

FIELD COIL

M.M. FINAL

ASSEMBLY

M.M. TEST

PAINTERS

SHIPPING

Figure VI.5
Production Network of Medium Size Machine
Figure VI.5
Production Network of Large Size Machine
Quick Deliveries By
- Improve work flow pattern
- Improve material handling
- Cellular organization
- Adoption of JIT philosophy

Improved Quality By
- Improved organization and structure
- Improved working conditions
- New plant and equipment
- Reduced likelihood of product damage

Reduced Costs And Overheads By
- Make better use of space
- Buy versus make policy
- Rationalised equipment
- Reduced W.I.P. and inventories
- Reduced staff for more effective utilisation

Improved Operating Efficiency By
- New plant and equipment
- Logical work flow pattern
- Cellular organization structure
- Cellular machine-systems and grouping
- Less maintenance demands

Figure VI.7 Desired Planning Structure

Figure VI.8 Towards Manufacturing Efficiency
IMPROVING THE WORK FLOW
BY PUTTING SUCCESSIVE OPERATIONS NEXT TO EACH OTHER

ARRANGING THE PLANT AND EQUIPMENT TOGETHER IN THE RIGHT SUB-CELLS
Appendix VII
CASE STUDY INFORMATION

VII.1 Introduction

The operational strategies and data information of GEC ALSTHOM Large Machines Ltd. used in the industrial case study are described in this appendix. The purpose of this appendix is to highlight the way to convert the realistic data into the simulation model required data. The introduction and the operation of the company can be found in Appendix VI. The initial comments on the case study is found in Chapter 13.

VII.2 Initial Comments

As discussed in Chapter 10, the model contains two levels for quick estimation and detailed analysis. To compare these two different levels, the information should be sufficient and compatible to each other.

The major difference between these two levels is the consideration of the performance of the cell. The first level considers the machining activity of all machines as a single operation of the working area. Therefore, the detail performance of each machine is not included. The second level involves the operation performance of each machine. The process planning information of components is required.

Major input data includes production and manufacturing related information of product and its components. Most input information is provided by the company. But the company usually operates in its own way by using its own strategies and quantitative information. Some of the information is therefore not directly related. Few of them are not available from the company.

VII.3 Products Modelled

The products to be produced in GEC ALSTHOM Large Machines Ltd. can be categorised into three types according to the size: large, medium and small sized machines. The production procedures for these three types of machines are found in Figure VI.4, VI.5 and VI.6.
Seven small sized machines have been selected as the candidates for the case study in the consideration of available information and general representation capability. They are Compak Copper Barred Rotor (COMCBR), Compak With D/C Rotor (COMDCRTR), Compak With Wound Rotor (COMWRTR), Flowpak Copper Barred Rotor (FPKCBR), Impak Flowpak Flapak D/C Rotor (FPKDC), Ingot Impak Flowpak With Wound Rotor (FPKWRTR), and High Voltage Compak With Die Cast Rotor (HVCOMDC). These seven types of small machines are manufactured following the production flow shown in Figure VI.4. The detailed product structure and production sequence are found from Figure VII.1 to Figure VII.7. Each of them shows the structural information, customer information and component information.

In these diagrams, the key components, sub-assemblies and final assembly are represented. The time needed to produce these components, time required by other component, cell name and the product characteristics are also described to provide general information about these products. Other components which are not important are not shown in these diagrams, instead they are described by using one block to present the detailed information which are shown in black broken lines in the figures with the legend of outside, machine cell and fabrication cell.

VII.4 Input Data

To carry out the case study, two levels of simulation system are compared. For these two levels, the input requirements for the simulation model include customer order information, product forecast information, inventory information, product information, production capacity information, production variability information, and component information. Those input information is provided by either accurate data or statistic data. The required input information is presented in Figure VII.8. They are discussed in the following sections respectively.

It should be noted that the data collection was to follow a sequence: sales, engineering, production planning, manufacture. However, all the required information can be classified as shown in Figure VII.8, the description of data collection and collation is therefore based on this categorisation with the emphasis on the conversion from real data into the computer code format used in the model.

Customer Order Data and Product Forecast Data. It is sales and marketing department’s responsibility to deal with customers, make forecasts and research marketing strategies.
The company's actual customers normally buy products through Original Equipment Managers, who are interested in the products price and delivery schedule, and Consultant Engineers, who are interested in the product/user interface and their specifications. The user is interested mainly with the products ease of maintenance, robustness and efficiency.

Once a tender arrives, the sales and marketing department figures out all the related data, such as customer name, project name, enquiry number, delivery, quantity, validity, payment terms, selling price, related costs and profit. A simplified example is found in Figure VII.9. In this tender make-up estimate sheet, besides product information and customer information, the detailed cost information is also provided to decide product price and calculate profit.

Upon receiving this tender and according to specifications from the customer, the person who is responsible for this type of products in the sales and marketing department generates an engineering drawing with the close discussion with engineering department and negotiation with the customer. Meanwhile the sales and marketing department sends the information to production control and planning department to check the production capacity for this tender. After the company gets this tender, the final work in engineering drawing and production department begins.

The sales and marketing department also makes forecast for the products. This forecast is in terms of profit not the quantity and is built on possible contracts or tenders. This data is listed in the forward order book analysis current list which can be found in Figure VII.10.

However, the forecast data in the simulation model is described in quantity required monthly (refer to Chapter 10). The default data is therefore used in the case study. It should be noted that the company belongs to Make-To-Order production, the forecast data cannot influence the production performance. Thus the accuracy of the forecast data is not very important with regard to the system performance.

Inventory Data. GEC ALSTHOM Large Machines Ltd. may be considered as a Make-To-Order company. Nothing is produced before receiving the customer order [267]. The inventory management is operated based on this philosophy.

By receiving the cost information from the estimating department and the product design information from engineering design department, the inventory control produces the purchasing information to buy raw materials and components. Even if the inventory level falls to a certain level e.g., reorder point, the new order will not be produced until the new customer order is received [267]. Therefore the inventory data such as product Economic Order Quantity, product reorder point and average lead time demand are not available from the company. They are initialised as zero in the case study. The product lot size is equal to 1.
The percentage of delivery requests is shown by a pie chart in Figure VII.11. Since 59% of deliveries requested are due in 4 weeks or less from receipt of the requisition, the inventory stock in the company is very high. In total the tied up capital is about 2 million, in which raw material takes about 1.5 million and the components 0.5 million. This results from the conflicting goals of inventory management [267], poor forecasting and long production lead times.

Therefore the company plans to implement the Just-In-Time philosophy to improve the schedule in materials, order quantity and reduction in lead time. The inventory control department plans to reduce its 6 months stock in hand to 1 month stock in hand. The progress has been achieved recently. Figure VII.12 presents the reduction of press cell steel stock.

**Product Data.** Product data involves stock quantity, safety quantity, lead time and bill of materials. For the first two pieces of information, refer to section VII.3. Since the products the company produces are general motors and generators, they are ordered by the customer with specialisation. One may be different from the other. Standard products are rarely produced and stored in stock. The products are usually shipped out directly after testing. Therefore the stock quantity and safety quantity is 0. Lead time and structural information is found from Figures VII.1 to VII.7.

**Production Capacity.** In order to make accurate production plans, the detailed production capacity of each cell is needed. The company calculated the available hours in each month based on available manpower. Figure VII.13 and Figure VII.14 represent the capacity details of each cell in the budget period which is from September 1990 to July 1991. For the production management, the company uses the Working Cell Reference Code to present the cell name.

The production capacity used in the model (refer to Chapter 10) represents the overall system capacity in terms of quantity. Therefore the production capacity provided by the company has to be converted into the quantity figures. Since there is no simple formula for the conversion, the data used is not very accurate. Again to the Make-To-Order production, this data has less influence than in Make-To-Stock production [266].

Each cell is generally divided into three areas to store and manufacture components, namely input store, output store and working area. The capacity of each area cannot be provided with the exact number. Instead infinite capacity is used in the model. The infinite capacity here means the capacity is large enough to handle all incoming components. By analysis of the simulation results, the varying part number in each area can be got to provide the accurate capacity information of each area. For the detailed discussion the reader is recommended to refer to Chapter 10.
Production Variability. Production variability information includes production early or late variability, information delay variability, subcontract delay variability and average time in the pre-production areas.

In general the shipment of products is later than the required due date. The production lateness is variable. This lateness is caused for several reasons. Usually they are from the pre-production, such as drawing office, planning office, sales department, engineering and others. The distribution of lateness among these areas is shown in Figure VII.15.

The information delay to the cell is shown in Figure VII.16. This delay represents the detailed process planning delay in the planning office. It is mainly caused by the delay from the drawing office. In this diagram, the total number of prints from the office is 218 in that week shown on the diagram. In these printings only 31.7% is produced on time. 7.3% is delayed due to the tag date, 36.7% is late because of the delay from production planning.

The production delay in the cell results from the tag arrival delay and the delay from the drawing changes. Once the drawing office has issued a drawing, the planners plan it, and work is begun. It becomes more difficult to change the design. Drawing mistakes are often discovered by the planners and so are corrected before manufacturing begins. But other errors must be rectified on the shop floor. If this is the case, long delays may result or, at worst, the items must be scrapped.

The production delay changes among the cells. Usually the production of Coil cell is on time. The production in Fabrication cell is always later. Figure VII.17 presents the fabrication cell performance which indicates the average actually completed and the required output according to the time.

Since these variabilities vary themselves, no fixed data is available to represent the real situation. The average variability is therefore used and is represented as percentage of product lead time. Subcontract delay variability is set to be zero because all the purchased materials and components are on time.

Component Data. The component data involves lead time information and the process plan. The total lead time can be found from Figure VII.1 to Figure VII.7. The detailed process plan of each component shows the routing procedure and time required, resource needed and the moving and queueing time. They are only needed when the consideration of detailed modelling is applied. According to information from Figures VII.1 to VII.7, the component information can be got in the format used in materials requirements planning. The related data can be found from Figures VIII.1 to VIII.7.
VII.5 Input Rules

One aim of the simulation is to predict the performance of the system based on alternative scenarios which are obtained by the entry of different data and rules. Therefore, the operating rules are required to model the actions in the manufacturing system. The roles of these strategies have been discussed in Chapter 10. With regard to the user interaction with the information entry, the reader is recommended to refer to Chapter 11.

The scope of these rules is to enable the user to define his own system to model the basic activities in his own environment. The operating strategies include customer order releasing, production control, material control, abstraction level, scheduling rules, inspection and inventory management. These are the rules that the user can change during the simulation. The abstraction level rule is the simulation model required rule. These rules are listed in Figure VII.18 and discussed in detail in the following section.

Customer Order Releasing. The company usually deals with each single customer order. When an order arrives, the sales department immediately sends the information to the estimating department which generates the production estimation data and the quote for the product and materials. Since there is no stock for completed products, the customer order will be released to production and the production plan will be made depending on the forecast data.

Production Control. Due to the production type of the company, no safety stock rules and product net demand calculation rules are applied. Information delay exits in the company, the variability of information delay is found in section VII.3.

Material Control. The company implements several strategies to control the material flow on the shop floor. By comparing the different rules, the right strategies will be used to manage the material in each cell. The general rules contain material movement rules and material supply rules.

--Material movement rules
1. No material is shipped to the next cell early (ie, it goes within the required week or late). The work may be completed early but it will be stored within the cell until the correct shipping week.
2. Any materials can be shipped to the next stage when finished irrespective of its scheduled finished date (based on the lead time and off set).

--Material supply rules
1. All raw materials are always available.
2. Any raw materials have a user defined variability expressed as a %age of their lead time.
The materials or components from sub-contract are always supplied on time or early. This rule is applicable in the simulation model.

**Abstraction Level and Inspection.** These rules are not the operational strategies used by the company to control production, but the rules for selecting the simulation model level. The approximate level and the detailed analysis level are both tested in this case study. For inspection rules, the test cell in the company is used to undertake this task. Generally speaking, it is quite unusual to find the defects in the products and send them back to the cell again. There are test procedures in each cell. If some of the procedures are not satisfied, the parts are sent back to be re-worked.

**Scheduling Rules.** The company uses certain strategies to control the scheduling on the shop floor. However the task of scheduling the work for the cell is difficult. Work may not arrive on the date it is due, this has a knock-on effect on the cell. On the other hand, the order of manufacture of the jobs depends upon the date on which they are due to start their first operations. Jobs already released to the shop floor usually have priority. When a resource is idle, a new job will be issued to it from the store of tags which arrive with new work. Ideally work would flow through the cell adhering to the work-to list. Unfortunately due to delays and changing priorities this situation never occurs.

The rules implemented at the company are not clearly defined. For scheduling at this level the cell supervisor relies upon the work-to list which provides the day-to-day information for the production in the cell. The order of priorities of work on the list depends on their priority factors. This parameter is a function of lead time and working day available. Therefore for all of the available rules used in the model are tested in this case study.

There are no particular scheduling rules applied to each individual machine. Again all of the rules implemented in the model are used for the comparison.

**Inventory Management.** As discussed in section VII.4, the completed products are usually shipped out directly after testing since the company does not store the finished product. Therefore the company deals with only each order not the individual product in each order.

**VII.6 Simulation Running Input**

For the simulation model running, the required data and rules are discussed in the last few sections. However, only some data can be directly used in the model, the other available information should be edited for the simulation execution. What is more some information is not available at all. These of course can lead to the inaccuracy of the running results. For the detailed discussion on output results, refer to Chapter 14.
Figure VII.1  

Product Data (1)
Figure VII.3

Product Data (3)
Figure VII.6

Product Data (6)
Figure VII.8

Input Data for the Case Study

Input Data

- Customer Order Data
  - Quantity
  - Customer identification
  - Specification
  - Due Date
- Product Forecast Data
  - Forecast Data
  - Safety Stock
- Inventory Data
  - Product EOQ Data
  - Product Recorder Point
  - Product Lot Size
  - Average Lead Time Demand
- Product Data
  - Lead Time
  - Stock Quantity
  - BOM
  - Safety Quantity
- Production Capacity
  - System Production Capacity
  - Input Store Capacity
  - Output Store Capacity
  - Work Areas Capacity
- Production Variability
  - Production Early or Later Variability
  - Information Delay Variability
  - Sub-contract Delay Variability
  - Average Time in Production Planning Area
- Component Data
  - Lead Time
  - Process Plan
<table>
<thead>
<tr>
<th>Item No/Machine Class Code</th>
<th>R. C. Number/Date</th>
<th>Quantity/Delivery Date</th>
<th>Machine/Exciter Frame Size</th>
<th>Output</th>
<th>Poles/Speed</th>
<th>Supply</th>
<th>Enclosure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>% Value</td>
<td>% Total Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Material</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labour</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factory Overheads</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shop Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Shop Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Special Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Witness Testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factory Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Selling Price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selling Price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gross Margin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>---------------</td>
<td>------------</td>
<td>------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>HYDRO GENERATORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LARGE</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIUM</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALL</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC GENERATORS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LARGE</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIUM</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALL</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H. O. D.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LARGE</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIUM</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALL</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC MACHINES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LARGE</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIUM</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMALL</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MACHINE OTHER MAKE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HYDRO</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERATOR</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC GENERATOR</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MACHINE CONSTRUCT'N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HYDRO</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERATOR</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC GENERATOR</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure VII.10 Forward Order Book Analysis Current
Analysis of Delivery Requests

Note: 59% of deliveries requested in 4 weeks or less from receipt of requisition.
Figure VII.12

Press Shop Steel Stock Reduction

PRESS SHOP STEEL STOCK REDUCTION

STEEL REDUCTION

TARGET

ACTUAL

WEEK NUMBER

940 930 920 910 900 890 880 870 860 850 840 830 820 810 800 790 780 770 760 750 740 730 720 710 700 690 680 670 660

29 31 33 35 37 39 41 43 45
<table>
<thead>
<tr>
<th></th>
<th>9009</th>
<th>9010</th>
<th>9011</th>
<th>9012</th>
<th>9013</th>
<th>9014</th>
<th>9015</th>
<th>9016</th>
<th>9017</th>
<th>9018</th>
</tr>
</thead>
<tbody>
<tr>
<td>Press</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>811</td>
<td>953</td>
<td>1015</td>
<td>1017</td>
<td>885</td>
<td>792</td>
<td>942</td>
<td>879</td>
<td>832</td>
<td>920</td>
</tr>
<tr>
<td>Mot Week</td>
<td>893</td>
<td>1049</td>
<td>1116</td>
<td>1119</td>
<td>973</td>
<td>872</td>
<td>1037</td>
<td>967</td>
<td>915</td>
<td>1012</td>
</tr>
<tr>
<td>Corebuild</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>278</td>
<td>326</td>
<td>347</td>
<td>348</td>
<td>303</td>
<td>271</td>
<td>322</td>
<td>301</td>
<td>285</td>
<td>315</td>
</tr>
<tr>
<td>Mot Week</td>
<td>305</td>
<td>359</td>
<td>382</td>
<td>383</td>
<td>333</td>
<td>298</td>
<td>355</td>
<td>331</td>
<td>313</td>
<td>346</td>
</tr>
<tr>
<td>DC Coil</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>648</td>
<td>761</td>
<td>810</td>
<td>812</td>
<td>706</td>
<td>632</td>
<td>752</td>
<td>702</td>
<td>664</td>
<td>735</td>
</tr>
<tr>
<td>Mot Week</td>
<td>712</td>
<td>837</td>
<td>891</td>
<td>893</td>
<td>777</td>
<td>696</td>
<td>828</td>
<td>772</td>
<td>731</td>
<td>808</td>
</tr>
<tr>
<td>AC Coil</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>1275</td>
<td>1499</td>
<td>1595</td>
<td>1599</td>
<td>1391</td>
<td>1245</td>
<td>1482</td>
<td>1382</td>
<td>1308</td>
<td>1447</td>
</tr>
<tr>
<td>Mot Week</td>
<td>1403</td>
<td>1648</td>
<td>1754</td>
<td>1759</td>
<td>1530</td>
<td>1370</td>
<td>1630</td>
<td>1520</td>
<td>1439</td>
<td>1591</td>
</tr>
<tr>
<td>Field Coil</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>676</td>
<td>794</td>
<td>846</td>
<td>848</td>
<td>737</td>
<td>660</td>
<td>765</td>
<td>733</td>
<td>694</td>
<td>767</td>
</tr>
<tr>
<td>Mot Week</td>
<td>744</td>
<td>874</td>
<td>930</td>
<td>932</td>
<td>811</td>
<td>726</td>
<td>864</td>
<td>806</td>
<td>763</td>
<td>844</td>
</tr>
<tr>
<td>Winding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>1164</td>
<td>1368</td>
<td>1456</td>
<td>1459</td>
<td>1270</td>
<td>1137</td>
<td>1352</td>
<td>1261</td>
<td>1194</td>
<td>1320</td>
</tr>
<tr>
<td>Mot Week</td>
<td>1281</td>
<td>1505</td>
<td>1601</td>
<td>1605</td>
<td>1397</td>
<td>1250</td>
<td>1487</td>
<td>1388</td>
<td>1313</td>
<td>1452</td>
</tr>
<tr>
<td>Fabrication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>577</td>
<td>677</td>
<td>721</td>
<td>723</td>
<td>629</td>
<td>563</td>
<td>670</td>
<td>625</td>
<td>591</td>
<td>654</td>
</tr>
<tr>
<td>Mot Week</td>
<td>634</td>
<td>745</td>
<td>793</td>
<td>795</td>
<td>692</td>
<td>619</td>
<td>737</td>
<td>687</td>
<td>650</td>
<td>719</td>
</tr>
<tr>
<td>M/C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norm Week</td>
<td>1245</td>
<td>1463</td>
<td>1557</td>
<td>1561</td>
<td>1358</td>
<td>1216</td>
<td>1446</td>
<td>1349</td>
<td>1277</td>
<td>1412</td>
</tr>
<tr>
<td>Mot Week</td>
<td>1369</td>
<td>1609</td>
<td>1712</td>
<td>1717</td>
<td>1494</td>
<td>1337</td>
<td>1591</td>
<td>1484</td>
<td>1405</td>
<td>1553</td>
</tr>
</tbody>
</table>

**Figure VII.13**

Rough Cut Capacity Planning Work Centre Details (1)
<table>
<thead>
<tr>
<th></th>
<th>9009</th>
<th>9010</th>
<th>9011</th>
<th>9012</th>
<th>9101</th>
<th>9102</th>
<th>9103</th>
<th>9104</th>
<th>9105</th>
<th>9106</th>
<th>9107</th>
<th>9108</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M/C NC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Norm Week</strong></td>
<td>258</td>
<td>306</td>
<td>327</td>
<td>327</td>
<td>283</td>
<td>252</td>
<td>302</td>
<td>281</td>
<td>265</td>
<td>295</td>
<td>295</td>
<td>323</td>
</tr>
<tr>
<td><strong>Mot Week</strong></td>
<td>284</td>
<td>337</td>
<td>359</td>
<td>360</td>
<td>311</td>
<td>277</td>
<td>333</td>
<td>309</td>
<td>292</td>
<td>324</td>
<td>324</td>
<td>355</td>
</tr>
<tr>
<td><strong>M/C Large</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Norm Week</strong></td>
<td>369</td>
<td>434</td>
<td>461</td>
<td>463</td>
<td>402</td>
<td>360</td>
<td>429</td>
<td>400</td>
<td>378</td>
<td>418</td>
<td>418</td>
<td>456</td>
</tr>
<tr>
<td><strong>Mot Week</strong></td>
<td>406</td>
<td>477</td>
<td>508</td>
<td>509</td>
<td>443</td>
<td>396</td>
<td>471</td>
<td>440</td>
<td>416</td>
<td>460</td>
<td>460</td>
<td>502</td>
</tr>
<tr>
<td><strong>Fin Assy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Norm Week</strong></td>
<td>519</td>
<td>610</td>
<td>649</td>
<td>651</td>
<td>566</td>
<td>507</td>
<td>603</td>
<td>563</td>
<td>533</td>
<td>589</td>
<td>589</td>
<td>642</td>
</tr>
<tr>
<td><strong>Mot Week</strong></td>
<td>571</td>
<td>671</td>
<td>714</td>
<td>716</td>
<td>623</td>
<td>558</td>
<td>664</td>
<td>619</td>
<td>586</td>
<td>648</td>
<td>648</td>
<td>706</td>
</tr>
<tr>
<td><strong>Fin Assy Lar</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Norm Week</strong></td>
<td>694</td>
<td>815</td>
<td>867</td>
<td>869</td>
<td>756</td>
<td>677</td>
<td>806</td>
<td>752</td>
<td>711</td>
<td>787</td>
<td>787</td>
<td>857</td>
</tr>
<tr>
<td><strong>Mot Week</strong></td>
<td>763</td>
<td>896</td>
<td>954</td>
<td>956</td>
<td>832</td>
<td>745</td>
<td>886</td>
<td>827</td>
<td>782</td>
<td>865</td>
<td>865</td>
<td>843</td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Norm Week</strong></td>
<td>683</td>
<td>803</td>
<td>854</td>
<td>857</td>
<td>745</td>
<td>667</td>
<td>794</td>
<td>740</td>
<td>701</td>
<td>775</td>
<td>775</td>
<td>845</td>
</tr>
<tr>
<td><strong>Mot Week</strong></td>
<td>752</td>
<td>883</td>
<td>940</td>
<td>942</td>
<td>820</td>
<td>734</td>
<td>873</td>
<td>814</td>
<td>771</td>
<td>852</td>
<td>852</td>
<td>929</td>
</tr>
</tbody>
</table>
Total number of P/prints 218

Figure VII.15  Distribution of Lateness among Areas

Figure VII.16  Information Delay
FABRICATION LOAD TRENDS

Week Ending

2/23 3/22 4/19 5/17 6/14 7/12 8/9 9/7 10/5 11/2

LOAD  OVERDUE  PRODUCTIVITY

GEC ALSTHOM
Figure VII.18

Input Strategies for the Case Study

- Customer Order Releasing
  - Forecast Data
  - Safety Stock
- Production Control
  - Product Net Demand Calculation Rule
  - Information Delay Rule
  - Production Type Rule
  - Product Safety Stock Rule
- Material Control
  - Material Supply Rule
  - Material Movement Rule
  - Subcontract Variability Rule
- Abstraction Level
  - Detail Machine Level Rule
  - Approximate Machine Level Rule
- Scheduling
  - Shop Level Scheduling Rule
  - Machine Level Scheduling Rule
- Inspection
  - Inspection Involved Rule
  - No Inspection Involved Rule
- Inventory Management
  - Product Shipping Rule
  - Product Releasing Rule
Appendix VIII
SIMULATION INPUTS AND RESULTS

VIII.1 Introduction

This appendix mainly contains the detailed information on input information and presentation of output results discussed in Chapter 14.

VIII.2 General Product Data

General product data shows the real component information in the model. The data includes processing item and routing plan. They are listed from Figure VIII.1 to Figure VIII.7. Figure VIII.8 is the information of processing plan at detailed machine level.

VIII.3 Approximate Level

VIII.3.1 Input Rules and Data

At approximate level, the input rules and data which have been used under random process and fixed condition are listed in Figures VIII.9 and VIII.10.

VIII.3.2 Outputs

In Chapter 14, the general performance results have been presented in Figures 14.3, 14.4, and 14.5. The component performance results of these seven products are summarised in Figures VIII.11, VIII.12, VIII.13, VIII.14, VIII.15, VIII.16 and VIII.17.

VIII.4 Detailed Analysis Level

VIII.4.1 Input Rules and Data

At detailed level, the same input rules and data as those at approximate level have been used in the experiment with the only exception: abstraction level flag. The reader is recommended to refer to Figures VIII.9 and VIII.10.
VIII.4.2 Outputs

In Chapter 14, the general performance results at this level have been presented in Figures 14.12, 14.13 and 14.14. The component performance results of these seven products are summarised in Figures VIII.18, VIII.19, VIII.20, VIII.21, VIII.22, VIII.23 and VIII.24.

VIII.5 Work In Progress Level

For each experiment, the work in progress at each working area can be obtained by using the logger monitor and analyser. These dynamic figures represent the real component flow and therefore can help decide the working area capacity under certain input levels. The plot data under random process at approximate level is summarised in Figures VIII.25, VIII.26, VIII.27, VIII.28, VIII.29, VIII.30, VIII.31, VIII.32, VIII.33, VIII.34, VIII.35, VIII.36, VIII.37, VIII.38, VIII.39, VIII.40, VIII.41, VIII.42, VIII.43, showing the number of parts at input store, working area and output store at each shop.

The histogram of number of parts in each area can also be obtained, displaying the frequency of part number changes according to time. Some examples are presented in Figures VIII.44, VIII.45 and VIII.46.

VIII.6 Queue at Machines

At detailed level, there exists a queue at some machines. The queue length and the time in queue at each machine can be monitored. Figures VIII.47, VIII.48, VIII.49, VIII.50, VIII.51, VIII.52, VIII.53, VIII.54, VIII.55 and VIII.56 present the queue length changes and time in queue at machines.
Figure VIII.1

Component Information of Product A

```prolog
(((A214 1 (NIL) 30 45 0)
  (A114 1 (NIL) 30 40 0)
  (A133 1 (NIL) 20 25 0)
  (A213 1 (A214) 25 20 0)
  (A113 1 (A114) 5 35 0)
  (A143 1 (NIL) 10 30 5)
  (A112 1 (A113) 15 20 0)
  (A411 1 (NIL) 20 15 5)
  (A122 1 (A133 A143) 5 20 0)
  (A111 1 (A112 A122) 10 10 0)
  (A212 1 (A213) 5 10 0)
  (OrderA 1 (A111 A211 A411) 10 0 0))

((A214 ((Outside (NIL NIL)) (Outside (NIL NIL)))
  (A114 ((Outside (NIL NIL)) (S3-input (S3-output S3-exit)))
  (A133 ((S2-input (S2-output S2-exit)) (S4-input (S4-output S4-exit)))
  (A213 ((Outside (NIL NIL)) (S4-input (S4-output S4-exit)))
  (A113 ((S3-input (S3-output S3-exit)) (S3-input (S3-output S3-exit)))
  (A143 ((S1-input (S1-output S1-exit)) (S4-input (S4-output S4-exit)))
  (A112 ((S3-input (S3-output S3-exit)) (S5-input (S5-output S5-exit)))
  (A411 ((Outside (NIL NIL)) (A-input (A-output A-exit)))
  (A122 ((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit)))
  (A111 ((S5-input (S5-output S5-exit)) (A-input (A-output A-exit)))
  (A212 ((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit)))
  (A211 ((S5-input (S5-output S5-exit)) (A-input (A-output A-exit)))
  (OrderA ((A-input (A-output A-exit)) (NIL (NIL NIL))))
```

Figure VIII.2

Component Information of Product B

```
(((B124 1 (NIL) 30 45 0)
  (B153 1 (NIL) 20 30 5)
  (B123 1 (B124) 15 30 5)
  (B132 1 (NIL) 15 25 5)
  (B233 1 (NIL) 15 25 5)
  (B122 1 (B123 B153) 5 20 0)
  (B111 1 (B122 B132) 10 10 0)
  (B212 1 (B233) 5 15 0)
  (B211 1 (B212) 5 10 0)
  (OrderB 1 (B111 B211) 10 0 0))

(((B124 (((S1-input (S1-output S1-exit)) (S6-input (S6-output S6-exit))))
  (B153 (((S1-input (S1-output S1-exit)) (S4-input (S4-output S4-exit))))
  (B123 (((S5-input (S6-output S6-exit)) (S4-input (S4-output S4-exit))))
  (B132 (((Outside (NIL (NIL NIL))) (S5-input (S5-output S5-exit))))
  (B233 (((S1-input (S1-output S1-exit)) (S4-input (S4-output S4-exit))))
  (B122 (((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit))))
  (B111 (((S5-input (S5-output S5-exit)) (A-input (A-output A-exit))))
  (B212 (((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit))))
  (B211 (((S5-input (S5-output S5-exit)) (A-input (A-output A-exit))))
  (OrderB (((A-input (A-output A-exit)) (NIL (NIL NIL)))))
```
Figure VIII.3
Component Information of Product C

(((C124 1 (NIL) 30 45 0)
(C114 1 (NIL) 30 40 0)
(C153 1 (NIL) 20 30 5)
(C123 1 (C124) 15 30 5)
(C133 1 (NIL) 20 25 0)
(C113 1 (C114) 5 35 0)
(C132 1 (NIL) 15 25 5)
(C112 1 (C113) 15 20 0)
(C122 1 (C123 C133 C153) 5 20 0)
(C111 1 (C112 C122 C132) 10 10 0)
(OrderC 1 (C111) 10 10 0))

(((C124 ((S1-input (S1-output S1-exit)) (S6-input (S6-output S6-exit))))
(C114 ((Outside (NIL NIL)) (S3-input (S3-output S3-exit))))
(C153 ((S1-input (S1-output S1-exit)) (S4-input (S4-output S4-exit))))
(C123 ((S6-input (S6-output S6-exit)) (S4-input (S4-output S4-exit))))
(C133 ((S2-input (S2-output S2-exit)) (S4-input (S4-output S4-exit))))
(C113 ((S3-input (S3-output S3-exit)) (S3-input (S3-output S3-exit))))
(C132 ((Outside (NIL NIL)) (S5-input (S5-output S5-exit))))
(C112 ((S3-input (S3-output S3-exit)) (S5-input (S5-output S5-exit))))
(C122 ((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit))))
(C111 ((S5-input (S5-output S5-exit)) (A-input (A-output A-exit))))
(OrderC ((A-input (A-output A-exit)) (NIL (NIL NIL))))
Figure VIII.4

Component Information of Product D

| (D214 1 (NIL) 30 35 0)          | (D224 1 (NIL) 35 30 0)          |
| (D234 1 (NIL) 30 35 0)          | (D253 1 (NIL) 35 25 5)          |
| (D223 1 (NIL) 25 20 0)          | (D213 1 (D214) 10 25 5)         |
| (D283 1 (D234) 10 25 5)         | (D243 1 (D224) 10 20 0)         |
| (D212 1 (D213 D223 D243 D253 D283) 5 15 0) | (D211 1 (D212) 5 10 0)         |
| (OrderD 1 (D211) 10 0 0)        |                                      |

| (D214 ((Outside (NIL NIL)) (S6-input (S6-output S6-exit)))) |
| (D224 ((Outside (NIL NIL)) (S6-input (S6-output S6-exit)))) |
| (D234 ((Outside (NIL NIL)) (S6-input (S6-output S6-exit)))) |
| (D253 ((Outside (NIL NIL)) (S4-input (S4-output S4-exit)))) |
| (D223 ((S2-input (S2-output S2-exit)) (S4-input (S4-output S4-exit)))) |
| (D213 ((S6-input (S6-output S6-exit)) (S4-input (S4-output S4-exit)))) |
| (D283 ((S6-input (S6-output S6-exit)) (S4-input (S4-output S4-exit)))) |
| (D243 ((S6-input (S6-output S6-exit)) (S4-input (S4-output S4-exit)))) |
| (D212 ((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit)))) |
| (D211 ((S5-input (S5-output S5-exit)) (A-input (A-output A-exit)))) |
| (OrderD ((A-input (A-output A-exit)) (NIL (NIL NIL)))) |
((E215 1 (NIL) 30 50 0)
(E312 1 (NIL) 30 30 0)
(E512 1 (NIL) 35 25 0)
(E214 1 (E215) 20 30 0)
(E222 1 (NIL) 15 25 0)
(E611 1 (NIL) 20 15 5)
(E311 1 (E312) 15 15 5)
(E223 1 (E214) 10 20 0)
(E511 1 (E512) 10 15 5)
(E212 1 (E223) 5 15 0)
(E211 1 (E212 E222) 5 10 0)
(OrderE 1 (E211 E311 E511 E611) 10 0 0))

((E215 ((Outside (NIL NIL)) (S2-input (S2-output S2-exit))))
(E312 ((S1-input (S1-output S1-exit)) (S6-input (S6-output S6-exit))))
(E512 ((Outside (NIL NIL)) (Outside (NIL NIL))))
(E214 ((S2-input (S2-output S2-exit)) (Outside (NIL NIL))))
(E222 ((Outside (NIL NIL)) (S5-input (S5-output S5-exit))))
(E611 ((Outside (NIL NIL)) (A-input (A-output A-exit))))
(E311 ((S6-input (S6-output S6-exit)) (A-input (A-output A-exit))))
(E223 ((Outside (NIL NIL)) (S4-input (S4-output S4-exit))))
(E511 ((Outside (NIL NIL)) (A-input (A-output A-exit))))
(E212 ((S4-input (S4-output S4-exit)) (S5-input (S5-output S5-exit))))
(E211 ((S5-input (S5-output S5-exit)) (A-input (A-output A-exit))))
(OrderE ((A-input (A-output A-exit)) (NIL (NIL NIL))))

Figure VII.5
Component Information of Product E
Figure VIII.5
Component Information of Product F

```lisp
((F511 1 (NIL) 65 10 0)
 (F312 1 (NIL) 30 30 0)
 (F612 1 (NIL) 35 25 0)
 (F622 1 (NIL) 35 25 0)
 (F411 1 (NIL) 35 15 5)
 (F811 1 (NIL) 20 15 5)
 (F311 1 (F312) 15 15 5)
 (F611 1 (F612 F622) 10 15 5)
 (OrderF 1 (F311 F411 F511 F611 F811) 10 0 0))

(F511 ((Outside (NIL NIL)) (A-input (A-output A-exit)))
 (F312 ((S1-input (S1-output S1-exit)) (S6-input (S6-output S6-exit))))
 (F612 ((Outside (NIL NIL)) (Outside (NIL NIL))))
 (F622 ((Outside (NIL NIL)) (Outside (NIL NIL))))
 (F411 ((Outside (NIL NIL)) (A-input (A-output A-exit))))
 (F811 ((Outside (NIL NIL)) (A-input (A-output A-exit))))
 (F311 ((S8-input (S8-output S8-exit)) (A-input (A-output A-exit))))
 (F611 ((Outside (NIL NIL)) (A-input (A-output A-exit))))
 (OrderF ((A-input (A-output A-exit)) (NIL (NIL NIL))))
```
Figure VIII.7

Component Information of Product G
(A143 (S1-m1 S1-m3 S1-m4 S1-e 4 0.2 3 0.7 2 0.1))
(B124 (S1-m2 S1-m3 S1-m8 S1-e 12 1 5 0.5 11 0.5))
(B153 (S1-m3 S1-m8 S1-m5 S1-e 4 0.8 8 0.7 6 0.5))
(B233 (S1-m4 S1-m5 S1-m6 S1-e 3 0.6 6 0.6 4 0.8))
(C124 (S1-m2 S1-m3 S1-m8 S1-e 7 1.5 10 2 8 1.5))
(C153 (S1-m3 S1-m8 S1-m5 S1-e 6 1 5 1 6 1))
(E312 (S1-m5 S1-m4 S1-m6 S1-e 9 2 7 1 6 3))
(F312 (S1-m2 S1-m7 S1-m4 S1-e 10 2 7 2 8 1))
(G312 (S1-m1 S1-m4 S1-m6 S1-e 7 3 8 2 8 2))

Figure VIII.8
Component Information on Detailed Machine Level
**Input Information in Computer Code Under Random Condition**

---

**Order List:** (Orders: OrdersA OrdersB OrdersC OrdersD OrdersE OrdersF OrdersG)

**Customer Order Data List:**
- OrdersA: 55, 55, 55, 55, 55, 55, 55, 55, 55, 55
- OrdersB: 45, 45, 45, 45, 45, 45, 45, 45, 45, 45
- OrdersE: 15, 15, 15, 15, 15, 15, 15, 15, 15, 15
- OrdersF: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
- OrdersG: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

**Order Release Rule:** 7

**Order Release Interval:** 10

**Shop Level Scheduling Rule:** 5

**Machine Level Scheduling Rule:** 1

**Abstaction Level Selection:** 7

**Product Release Selection Rule:** 7

**Production Type Rule:** 7

**Product Inspection Rule:** 2

**Subcontracting Variability Flag:** 7

**Subcontract Variability Value:** 0

**Product Safety Stock:** 0

**Material Movement at Machine:** 7

**Time in Production Area:**
- Product Inventory Data: Early Range: 1, Late Range: 3
- Engineering: 1, 5
- Forwarsing: 1, 3
- Master Production Planning: 1, 4
- Master Production Scheduling: 1, 3

**Product Inventory Data:**
- OrdersA: 0
- OrdersB: 0
- OrdersC: 0
- OrdersD: 0
- OrdersE: 0
- OrdersF: 0
- OrdersG: 0

**Safety Stock Data:**
- OrdersA: 0
- OrdersB: 0
- OrdersC: 0
- OrdersD: 0
- OrdersE: 0
- OrdersF: 0
- OrdersG: 0

**Product Lead Time Data:**
- OrdersA: 0
- OrdersB: 6
- OrdersC: 0
- OrdersD: 0
- OrdersE: 0
- OrdersF: 0
- OrdersG: 0

**Product EQ:**
- OrdersA: 0
- OrdersB: 0
- OrdersC: 0
- OrdersD: 0
- OrdersE: 0
- OrdersF: 0
- OrdersG: 0

**Production Capacity:**
- OrdersA: 100
- OrdersB: 100
- OrdersC: 95
- OrdersD: 90
- OrdersE: 90
- OrdersF: 90
- OrdersG: 90

---

**Figure VIII.9**
**Input Data of Simulation Model**

**Order List:**
- Orders: 0
- Order: 0
- Order: 0
- Order: 0
- Order: 0
- Order: 0
- Order: 0
- Order: 0

**Customer Order Data List:**
- Quantity: 0
- Quantity: 0
- Quantity: 0
- Quantity: 0
- Quantity: 0
- Quantity: 0
- Quantity: 0
- Quantity: 0

**Order Release Rule:**
- T

**Order Release Interval:**
- T

**Shop Level Scheduling Rule:**
- 5

**Machine Level Scheduling Rule:**
- I

**Abstraction Level Selection:**
- T

**Product Release Selection Rule:**
- T

**Production Type Rule:**
- T

**Product Inspection Rule:**
- T

**Concurrent Variability:**
- T

**Concurrent Variability Value:**
- 0

**Product Safety Stock:**
- 0

**Material Movement of Machine:**
- T

**Time in Production Area:**
- Early Range
- Late Range

**Product Inventory Data:**
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0

**Safety Stock Data:**
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0

**Product Lead Time Data:**
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0

**Product MQ:**
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0

**Production Capacity:**
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0
- Orders: 0

**Shop Assembly:**
- Shop 1: Stop
- Shop 2: Stop
- Shop 3: Stop
- Shop 4: Stop
- Shop 5: Stop

**Input Information in Computer Code Under Fixed Condition**

---

Figure VIII.10
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A114</td>
<td>10</td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>41</td>
<td>30</td>
<td>30</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>12.5</td>
<td>42.5</td>
<td>30</td>
<td>30</td>
<td>56</td>
</tr>
<tr>
<td>A141</td>
<td>40.0</td>
<td>60.22017</td>
<td>20.21017</td>
<td>20</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>41.0</td>
<td>61.36017</td>
<td>20.36017</td>
<td>20</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>42.5</td>
<td>62.93017</td>
<td>20.93017</td>
<td>20</td>
<td>76</td>
</tr>
<tr>
<td>A113</td>
<td>40</td>
<td>65</td>
<td>25</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>66</td>
<td>25</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>42.5</td>
<td>67.5</td>
<td>25</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>A143</td>
<td>45</td>
<td>50.29413</td>
<td>5.0</td>
<td>5</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>46.0</td>
<td>51.29413</td>
<td>5.0</td>
<td>5</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>47.5</td>
<td>52.54013</td>
<td>5.0</td>
<td>5</td>
<td>66</td>
</tr>
<tr>
<td>A112</td>
<td>50</td>
<td>70</td>
<td>20</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>71</td>
<td>20</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>52.5</td>
<td>72.5</td>
<td>20</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td>A122</td>
<td>78.0</td>
<td>81.0</td>
<td>5.0</td>
<td>5</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>78.0</td>
<td>81.0</td>
<td>5.0</td>
<td>5</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>78.0</td>
<td>81.0</td>
<td>5.0</td>
<td>5</td>
<td>81</td>
</tr>
<tr>
<td>A111</td>
<td>81.00144</td>
<td>81.09413</td>
<td>10.09413</td>
<td>10</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>81.00144</td>
<td>81.09413</td>
<td>10.09413</td>
<td>10</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>81.00153</td>
<td>81.09413</td>
<td>10.09413</td>
<td>10</td>
<td>91</td>
</tr>
<tr>
<td>A212</td>
<td>55.0</td>
<td>70.0</td>
<td>5.0</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>56.0</td>
<td>71.0</td>
<td>5.0</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>62.5</td>
<td>72.5</td>
<td>5.0</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td>A211</td>
<td>86.0</td>
<td>91.04601</td>
<td>5.04601</td>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>86.0</td>
<td>91.04601</td>
<td>5.04601</td>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>86.0</td>
<td>91.04601</td>
<td>5.04601</td>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td>OrderA</td>
<td>91.09401</td>
<td>101.14976</td>
<td>10.06504</td>
<td>10</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>91.09401</td>
<td>101.14976</td>
<td>10.06504</td>
<td>10</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>91.09401</td>
<td>101.14976</td>
<td>10.06504</td>
<td>10</td>
<td>101</td>
</tr>
</tbody>
</table>

Figure VIII.11: Performance of Component of Product A
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>B214</td>
<td>Run 1 22.0</td>
<td>52.40745</td>
<td>30.40745</td>
<td>30</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Run 2 21.0</td>
<td>51.40745</td>
<td>30.40745</td>
<td>30</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Run 3 22.5</td>
<td>52.546533</td>
<td>30.546533</td>
<td>30</td>
<td>62</td>
</tr>
<tr>
<td>B153</td>
<td>Run 1 47.0</td>
<td>67.08463</td>
<td>20.08463</td>
<td>20</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Run 2 46.0</td>
<td>66.28537</td>
<td>20.28537</td>
<td>20</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Run 3 47.5</td>
<td>68.045066</td>
<td>20.045066</td>
<td>20</td>
<td>62</td>
</tr>
<tr>
<td>B123</td>
<td>Run 1 52.0</td>
<td>71.022222</td>
<td>15.022222</td>
<td>15</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 2 52.0</td>
<td>71.022222</td>
<td>15.022222</td>
<td>15</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 3 51.0</td>
<td>70.50066</td>
<td>15.00066</td>
<td>15</td>
<td>82</td>
</tr>
<tr>
<td>B132</td>
<td>Run 1 57.0</td>
<td>71.18558</td>
<td>15.18558</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 2 55.5</td>
<td>71.19059</td>
<td>15.19059</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 3 57.5</td>
<td>71.28556</td>
<td>15.28556</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td>B223</td>
<td>Run 1 57.0</td>
<td>71.18556</td>
<td>15.18556</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 2 56.0</td>
<td>71.19059</td>
<td>15.19059</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 3 57.5</td>
<td>71.28556</td>
<td>15.28556</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td>B122</td>
<td>Run 1 82.0</td>
<td>97.0</td>
<td>5.0</td>
<td>5</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 2 82.0</td>
<td>97.0</td>
<td>5.0</td>
<td>5</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 3 81.0</td>
<td>96.9</td>
<td>5.0</td>
<td>5</td>
<td>86</td>
</tr>
<tr>
<td>B111</td>
<td>Run 1 87.0</td>
<td>97.06647</td>
<td>10.06647</td>
<td>10</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 2 87.0</td>
<td>97.06647</td>
<td>10.06647</td>
<td>10</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 3 86.0</td>
<td>96.15632</td>
<td>10.15631</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>B212</td>
<td>Run 1 87.0</td>
<td>92.3</td>
<td>5.0</td>
<td>5</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 2 87.0</td>
<td>92.3</td>
<td>5.0</td>
<td>5</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 3 86.0</td>
<td>91.3</td>
<td>5.0</td>
<td>5</td>
<td>91</td>
</tr>
<tr>
<td>B211</td>
<td>Run 1 92.0</td>
<td>97.00</td>
<td>5.09999</td>
<td>5</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 2 92.0</td>
<td>97.0</td>
<td>5.09999</td>
<td>5</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 3 81.0</td>
<td>96.39999</td>
<td>5.09999</td>
<td>5</td>
<td>97</td>
</tr>
<tr>
<td>OrderB</td>
<td>Run 1 97.0547</td>
<td>107.15442</td>
<td>10.09752</td>
<td>10</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 2 97.0547</td>
<td>107.15442</td>
<td>10.09752</td>
<td>10</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 3 96.15632</td>
<td>106.24711</td>
<td>10.09752</td>
<td>10</td>
<td>106</td>
</tr>
</tbody>
</table>

Figure VIII.12

Performance of Component of Product B
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C124</td>
<td>Run 1: 26.0</td>
<td>56.813564</td>
<td>30.813564</td>
<td>30</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Run 2: 57.813564</td>
<td>82.7002</td>
<td>30.7002</td>
<td>30</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Run 3: 67.813564</td>
<td>97.813564</td>
<td>30.813564</td>
<td>30</td>
<td>75</td>
</tr>
<tr>
<td>C114</td>
<td>Run 1: 31.5</td>
<td>67.813564</td>
<td>30.813564</td>
<td>30</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>Run 2: 62.813564</td>
<td>92.813564</td>
<td>30.813564</td>
<td>30</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 3: 97.813564</td>
<td>137.813564</td>
<td>30.813564</td>
<td>30</td>
<td>81</td>
</tr>
<tr>
<td>C153</td>
<td>Run 1: 51.8</td>
<td>71.524284</td>
<td>20.524284</td>
<td>20</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Run 2: 52.8</td>
<td>72.524284</td>
<td>20.524284</td>
<td>20</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Run 3: 77.524284</td>
<td>107.524284</td>
<td>30.524284</td>
<td>30</td>
<td>96</td>
</tr>
<tr>
<td>C123</td>
<td>Run 1: 74.8</td>
<td>89.149536</td>
<td>15.149536</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 2: 74.8</td>
<td>89.149536</td>
<td>15.149536</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 3: 76.8</td>
<td>91.149536</td>
<td>15.149536</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>C133</td>
<td>Run 1: 56.8</td>
<td>76.29313</td>
<td>20.29313</td>
<td>20</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Run 2: 57.8</td>
<td>77.29313</td>
<td>20.29313</td>
<td>20</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Run 3: 82.29313</td>
<td>112.29313</td>
<td>30.29313</td>
<td>30</td>
<td>96</td>
</tr>
<tr>
<td>C113</td>
<td>Run 1: 61.5</td>
<td>66.30779</td>
<td>5.30779</td>
<td>5</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Run 2: 62.5</td>
<td>67.30779</td>
<td>5.30779</td>
<td>5</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Run 3: 67.5</td>
<td>72.30779</td>
<td>5.30779</td>
<td>5</td>
<td>88</td>
</tr>
<tr>
<td>C132</td>
<td>Run 1: 61.8</td>
<td>76.81507</td>
<td>15.81507</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 2: 62.8</td>
<td>77.81507</td>
<td>15.81507</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 3: 63.8</td>
<td>82.81507</td>
<td>15.81507</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>C112</td>
<td>Run 1: 84.8</td>
<td>90.00024</td>
<td>15.00024</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 2: 84.8</td>
<td>90.00024</td>
<td>15.00024</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 3: 86.00024</td>
<td>101.00024</td>
<td>15.00024</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>C122</td>
<td>Run 1: 84.0</td>
<td>90.00024</td>
<td>15.00024</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 2: 84.0</td>
<td>90.00024</td>
<td>15.00024</td>
<td>15</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Run 3: 96.0</td>
<td>101.00024</td>
<td>15.00024</td>
<td>15</td>
<td>101</td>
</tr>
<tr>
<td>C111</td>
<td>Run 1: 99.0</td>
<td>109.23138</td>
<td>10.23138</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Run 2: 94.149536</td>
<td>104.149536</td>
<td>10.149536</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Run 3: 101.149536</td>
<td>111.149536</td>
<td>10.149536</td>
<td>10</td>
<td>101</td>
</tr>
<tr>
<td>OrderC</td>
<td>Run 1: 109.2338</td>
<td>119.32599</td>
<td>10.02146</td>
<td>10</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>Run 2: 105.2338</td>
<td>119.2103</td>
<td>10.02146</td>
<td>10</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>Run 3: 111.2338</td>
<td>121.2103</td>
<td>10.02146</td>
<td>10</td>
<td>121</td>
</tr>
</tbody>
</table>

Figure VIII.13: Performance of Component of Product C
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>D214 Run 1</td>
<td>42</td>
<td>72</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D214 Run 2</td>
<td>41</td>
<td>71</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D214 Run 3</td>
<td>41.5</td>
<td>71.5</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D224 Run 1</td>
<td>42</td>
<td>77</td>
<td>35</td>
<td>35</td>
<td>84</td>
</tr>
<tr>
<td>D224 Run 2</td>
<td>41</td>
<td>76</td>
<td>35</td>
<td>35</td>
<td>84</td>
</tr>
<tr>
<td>D224 Run 3</td>
<td>41.5</td>
<td>76.5</td>
<td>35</td>
<td>35</td>
<td>84</td>
</tr>
<tr>
<td>D234 Run 1</td>
<td>42</td>
<td>72</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D234 Run 2</td>
<td>41</td>
<td>71</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D234 Run 3</td>
<td>41.5</td>
<td>71.5</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td>D253 Run 1</td>
<td>47</td>
<td>82</td>
<td>35</td>
<td>35</td>
<td>98</td>
</tr>
<tr>
<td>D253 Run 2</td>
<td>46</td>
<td>81</td>
<td>35</td>
<td>35</td>
<td>98</td>
</tr>
<tr>
<td>D253 Run 3</td>
<td>46.5</td>
<td>81.5</td>
<td>35</td>
<td>35</td>
<td>98</td>
</tr>
<tr>
<td>D223 Run 1</td>
<td>62.0</td>
<td>87.4762</td>
<td>25.47619</td>
<td>25</td>
<td>96</td>
</tr>
<tr>
<td>D223 Run 2</td>
<td>61.0</td>
<td>86.67245</td>
<td>25.67244</td>
<td>25</td>
<td>96</td>
</tr>
<tr>
<td>D223 Run 3</td>
<td>61.5</td>
<td>86.76519</td>
<td>25.76566</td>
<td>25</td>
<td>96</td>
</tr>
<tr>
<td>D213 Run 1</td>
<td>72.0</td>
<td>92.02165</td>
<td>10.021652</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D213 Run 2</td>
<td>71.0</td>
<td>81.02165</td>
<td>10.01652</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D213 Run 3</td>
<td>71.5</td>
<td>81.54558</td>
<td>10.03475</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>D283 Run 1</td>
<td>72.0</td>
<td>82.01049</td>
<td>10.01049</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D283 Run 2</td>
<td>71.0</td>
<td>81.01049</td>
<td>10.01648</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D283 Run 3</td>
<td>71.5</td>
<td>81.55035</td>
<td>10.05057</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>D243 Run 1</td>
<td>77.0</td>
<td>87.05675</td>
<td>10.05675</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D243 Run 2</td>
<td>76.0</td>
<td>86.09279</td>
<td>10.09279</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D243 Run 3</td>
<td>76.5</td>
<td>85.50931</td>
<td>10.05093</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>D212 Run 1</td>
<td>58.0</td>
<td>102.0</td>
<td>5.0</td>
<td>5</td>
<td>103</td>
</tr>
<tr>
<td>D212 Run 2</td>
<td>58.0</td>
<td>102.0</td>
<td>5.0</td>
<td>5</td>
<td>103</td>
</tr>
<tr>
<td>D212 Run 3</td>
<td>58.0</td>
<td>102.0</td>
<td>5.0</td>
<td>5</td>
<td>103</td>
</tr>
<tr>
<td>D211 Run 1</td>
<td>102.0</td>
<td>106.04158</td>
<td>5.04158</td>
<td>5</td>
<td>108</td>
</tr>
<tr>
<td>D211 Run 2</td>
<td>102.0</td>
<td>106.04158</td>
<td>5.04158</td>
<td>5</td>
<td>108</td>
</tr>
<tr>
<td>D211 Run 3</td>
<td>102.0</td>
<td>106.06656</td>
<td>5.06657</td>
<td>5</td>
<td>108</td>
</tr>
<tr>
<td>OrderD Run 1</td>
<td>105.04158</td>
<td>118.0654</td>
<td>10.023451</td>
<td>10</td>
<td>118</td>
</tr>
<tr>
<td>OrderD Run 2</td>
<td>105.04158</td>
<td>118.07548</td>
<td>10.023457</td>
<td>10</td>
<td>118</td>
</tr>
<tr>
<td>OrderD Run 3</td>
<td>105.09055</td>
<td>118.07505</td>
<td>10.012551</td>
<td>10</td>
<td>118</td>
</tr>
</tbody>
</table>

Figure VIII.14

Performance of Component of Product D
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>E215</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E212</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E312</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E512</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E214</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E222</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E611</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E311</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E223</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E511</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E212</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>E211</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
<tr>
<td>OrderE</td>
<td>50</td>
<td>80</td>
<td>30</td>
<td>30</td>
<td>103</td>
</tr>
</tbody>
</table>

**Figure VIII.15**

Performance of Component of Product E
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>F511</td>
<td>Run 1</td>
<td>66</td>
<td>131</td>
<td>65</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>66</td>
<td>131</td>
<td>65</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>62.5</td>
<td>127.5</td>
<td>65</td>
<td>141</td>
</tr>
<tr>
<td>F312</td>
<td>Run 1</td>
<td>61.0</td>
<td>111.27613</td>
<td>30.37613</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>61.0</td>
<td>111.27613</td>
<td>30.37613</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>77.5</td>
<td>156.000914</td>
<td>30.500914</td>
<td>121</td>
</tr>
<tr>
<td>F612</td>
<td>Run 1</td>
<td>81</td>
<td>116</td>
<td>35</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>81</td>
<td>116</td>
<td>35</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>77.5</td>
<td>112.5</td>
<td>35</td>
<td>126</td>
</tr>
<tr>
<td>F622</td>
<td>Run 1</td>
<td>81</td>
<td>116</td>
<td>35</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>81</td>
<td>116</td>
<td>35</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>77.5</td>
<td>112.5</td>
<td>35</td>
<td>126</td>
</tr>
<tr>
<td>F411</td>
<td>Run 1</td>
<td>91</td>
<td>126</td>
<td>35</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>91</td>
<td>126</td>
<td>35</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>87.5</td>
<td>122.5</td>
<td>35</td>
<td>141</td>
</tr>
<tr>
<td>F811</td>
<td>Run 1</td>
<td>106</td>
<td>126</td>
<td>20</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>106</td>
<td>126</td>
<td>20</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>102.5</td>
<td>122.5</td>
<td>20</td>
<td>141</td>
</tr>
<tr>
<td>F311</td>
<td>Run 1</td>
<td>150.9</td>
<td>145.01779</td>
<td>15.01779</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>150.9</td>
<td>145.01779</td>
<td>15.01779</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>151.0</td>
<td>156.04585</td>
<td>15.04585</td>
<td>141</td>
</tr>
<tr>
<td>F611</td>
<td>Run 1</td>
<td>116</td>
<td>126</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>116</td>
<td>126</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>112.5</td>
<td>122.5</td>
<td>10</td>
<td>141</td>
</tr>
<tr>
<td>OrderF</td>
<td>Run 1</td>
<td>145.01779</td>
<td>155.04307</td>
<td>10.04307</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>145.01779</td>
<td>155.11006</td>
<td>10.11006</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>136.04585</td>
<td>146.10345</td>
<td>10.10345</td>
<td>153</td>
</tr>
</tbody>
</table>

Figure VIII.16

Performance of Component of Product F
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>G214</td>
<td>Run 1: 80</td>
<td>105</td>
<td>25</td>
<td>25</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>Run 2: 81</td>
<td>106</td>
<td>25</td>
<td>25</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>Run 3: 72</td>
<td>97</td>
<td>25</td>
<td>25</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Run 1: 100.0</td>
<td>125.3745</td>
<td>20.37452</td>
<td>20</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>Run 2: 101.0</td>
<td>123.3765</td>
<td>20.37652</td>
<td>20</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>Run 3: 97.0</td>
<td>112.0650</td>
<td>20.06504</td>
<td>20</td>
<td>110</td>
</tr>
<tr>
<td>G312</td>
<td>Run 1: 100.0</td>
<td>130.3438</td>
<td>30.34387</td>
<td>30</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 2: 101.0</td>
<td>124.3438</td>
<td>30.34387</td>
<td>30</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 3: 92.0</td>
<td>122.6419</td>
<td>30.64194</td>
<td>30</td>
<td>141</td>
</tr>
<tr>
<td>G213</td>
<td>Run 1: 105</td>
<td>120</td>
<td>25</td>
<td>25</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 2: 106</td>
<td>131</td>
<td>25</td>
<td>25</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 3: 97</td>
<td>122</td>
<td>25</td>
<td>25</td>
<td>141</td>
</tr>
<tr>
<td>G223</td>
<td>Run 1: 120</td>
<td>154.0</td>
<td>10</td>
<td>10</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 2: 121</td>
<td>154.0</td>
<td>10</td>
<td>10</td>
<td>154</td>
</tr>
<tr>
<td></td>
<td>Run 3: 112</td>
<td>141</td>
<td>10</td>
<td>10</td>
<td>141</td>
</tr>
<tr>
<td>G411</td>
<td>Run 1: 120</td>
<td>156</td>
<td>15</td>
<td>15</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 2: 121</td>
<td>156</td>
<td>15</td>
<td>15</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 3: 112</td>
<td>141</td>
<td>15</td>
<td>15</td>
<td>151</td>
</tr>
<tr>
<td>G212</td>
<td>Run 1: 154.0</td>
<td>159.0</td>
<td>5.0</td>
<td>5</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>Run 2: 154.0</td>
<td>159.0</td>
<td>5.0</td>
<td>5</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>Run 3: 141.0</td>
<td>146.0</td>
<td>5.0</td>
<td>5</td>
<td>149</td>
</tr>
<tr>
<td>G311</td>
<td>Run 1: 154.0</td>
<td>164.0084</td>
<td>10.00847</td>
<td>10</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 2: 154.0</td>
<td>164.0084</td>
<td>10.00847</td>
<td>10</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 3: 141.0</td>
<td>151.0041</td>
<td>10.00411</td>
<td>10</td>
<td>151</td>
</tr>
<tr>
<td>G211</td>
<td>Run 1: 159.0</td>
<td>164.0092</td>
<td>5.0009273</td>
<td>5</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 2: 159.0</td>
<td>164.0092</td>
<td>5.0009273</td>
<td>5</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Run 3: 145.0</td>
<td>151.0844</td>
<td>5.0841745</td>
<td>5</td>
<td>148</td>
</tr>
<tr>
<td>OrderG</td>
<td>Run 1: 164.00156</td>
<td>174.10732</td>
<td>10.05736</td>
<td>10</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>Run 2: 151.08948</td>
<td>161.1490</td>
<td>10.05926</td>
<td>10</td>
<td>161</td>
</tr>
</tbody>
</table>

Figure VIII.17  
Performance of Component of Product G
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>A214</td>
<td>Run 1</td>
<td>10</td>
<td>40</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>10</td>
<td>40</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>13.5</td>
<td>45.5</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>A114</td>
<td>Run 1</td>
<td>15</td>
<td>45</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>15</td>
<td>45</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>17.5</td>
<td>47.5</td>
<td>30</td>
<td>50</td>
</tr>
<tr>
<td>A133</td>
<td>Run 1</td>
<td>40.0</td>
<td>60.076000</td>
<td>20.079803</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>40.0</td>
<td>60.076000</td>
<td>20.079803</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>42.5</td>
<td>62.586978</td>
<td>20.08978</td>
<td>70</td>
</tr>
<tr>
<td>A213</td>
<td>Run 1</td>
<td>40.0</td>
<td>65</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>40.0</td>
<td>65</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>42.5</td>
<td>67.5</td>
<td>25</td>
<td>81</td>
</tr>
<tr>
<td>A113</td>
<td>Run 1</td>
<td>45.0</td>
<td>62.3</td>
<td>17.3</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>45.0</td>
<td>62.3</td>
<td>17.3</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>47.5</td>
<td>62.8</td>
<td>15.999999</td>
<td>70</td>
</tr>
<tr>
<td>A143</td>
<td>Run 1</td>
<td>50.043415</td>
<td>50.043415</td>
<td>5.0</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>45.0</td>
<td>62.3</td>
<td>17.3</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>47.5</td>
<td>62.8</td>
<td>15.999999</td>
<td>70</td>
</tr>
<tr>
<td>A112</td>
<td>Run 1</td>
<td>50.04902</td>
<td>81.04992</td>
<td>15.0</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>50.04902</td>
<td>81.04992</td>
<td>15.0</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>50.05792</td>
<td>81.05792</td>
<td>15.0</td>
<td>81</td>
</tr>
<tr>
<td>A111</td>
<td>Run 1</td>
<td>50</td>
<td>70</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>50</td>
<td>70</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>52.5</td>
<td>72.5</td>
<td>20</td>
<td>91</td>
</tr>
<tr>
<td>A122</td>
<td>Run 1</td>
<td>76.0</td>
<td>81.0</td>
<td>5.0</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>76.0</td>
<td>81.0</td>
<td>5.0</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>76.0</td>
<td>81.0</td>
<td>5.0</td>
<td>81</td>
</tr>
<tr>
<td>A11</td>
<td>Run 1</td>
<td>81.04992</td>
<td>91.08934</td>
<td>10.03421</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>81.04992</td>
<td>91.08934</td>
<td>10.03421</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>81.07292</td>
<td>91.07295</td>
<td>10.02532</td>
<td>91</td>
</tr>
<tr>
<td>A212</td>
<td>Run 1</td>
<td>55.0</td>
<td>70.0</td>
<td>5.0</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>55.0</td>
<td>70.0</td>
<td>5.0</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>57.5</td>
<td>72.5</td>
<td>5.0</td>
<td>86</td>
</tr>
<tr>
<td>A211</td>
<td>Run 1</td>
<td>50.0</td>
<td>61.07604</td>
<td>5.079422</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>56.0</td>
<td>61.07604</td>
<td>5.079422</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>56.0</td>
<td>61.07558</td>
<td>5.015579</td>
<td>91</td>
</tr>
<tr>
<td>OrderA</td>
<td>Run 1</td>
<td>91.038893</td>
<td>101.59493</td>
<td>10.035953</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>91.03354</td>
<td>101.03354</td>
<td>10.035953</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>91.076025</td>
<td>101.09173</td>
<td>10.021472</td>
<td>101</td>
</tr>
</tbody>
</table>

Figure VIII.18

Performance of Component of Product A (Detail Level)
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>B214</td>
<td>Run 1</td>
<td>22.0</td>
<td>52.0</td>
<td>30.0</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>22.0</td>
<td>52.0</td>
<td>30.0</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>22.0</td>
<td>52.0</td>
<td>30.0</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 1</td>
<td>47.0</td>
<td>84.5</td>
<td>37.5</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>47.0</td>
<td>84.5</td>
<td>37.5</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>47.0</td>
<td>78.2</td>
<td>30.66667</td>
<td>81</td>
</tr>
<tr>
<td>B153</td>
<td>Run 1</td>
<td>62.0</td>
<td>77.6222</td>
<td>15.02219</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>62.0</td>
<td>77.6222</td>
<td>15.02219</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>61.0</td>
<td>76.6855</td>
<td>15.006529</td>
<td>81</td>
</tr>
<tr>
<td>B123</td>
<td>Run 1</td>
<td>57.7</td>
<td>72.7</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>57.7</td>
<td>72.7</td>
<td>15</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>57.1</td>
<td>72.5</td>
<td>15</td>
<td>86</td>
</tr>
<tr>
<td>B233</td>
<td>Run 1</td>
<td>57.0</td>
<td>71.4</td>
<td>14.400002</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>57.0</td>
<td>71.4</td>
<td>14.400002</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>57.5</td>
<td>71.9</td>
<td>14.400002</td>
<td>86</td>
</tr>
<tr>
<td>B122</td>
<td>Run 1</td>
<td>84.5</td>
<td>89.5</td>
<td>5.0</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>84.5</td>
<td>89.2</td>
<td>5.0</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>81.0</td>
<td>86.9</td>
<td>5.0</td>
<td>86</td>
</tr>
<tr>
<td>B111</td>
<td>Run 1</td>
<td>89.5</td>
<td>99.54789</td>
<td>10.04789</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>89.5</td>
<td>99.54789</td>
<td>10.04789</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>86.9</td>
<td>98.5896</td>
<td>10.02896</td>
<td>96</td>
</tr>
<tr>
<td>B212</td>
<td>Run 1</td>
<td>87.0</td>
<td>92.9</td>
<td>5.0</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>87.0</td>
<td>92.9</td>
<td>5.0</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>86.0</td>
<td>91.0</td>
<td>5.0</td>
<td>91</td>
</tr>
<tr>
<td>B211</td>
<td>Run 1</td>
<td>92.0</td>
<td>97.09451</td>
<td>5.09451</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>92.0</td>
<td>97.09451</td>
<td>5.09451</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>91.0</td>
<td>96.05624</td>
<td>5.05624</td>
<td>96</td>
</tr>
<tr>
<td>Order B</td>
<td>Run 1</td>
<td>99.54789</td>
<td>109.640076</td>
<td>10.09786</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>99.54789</td>
<td>109.640076</td>
<td>10.09786</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>96.09524</td>
<td>106.34563</td>
<td>10.34563</td>
<td>106</td>
</tr>
</tbody>
</table>

**Figure VIII.19**

**Performance of Component of Product B (Detail Level)**
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C124</td>
<td>Run 1: 21.0</td>
<td>36.0</td>
<td>36.0</td>
<td>30</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Run 2: 32.5</td>
<td>64.0</td>
<td>36.0</td>
<td>30</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Run 3: 30</td>
<td>32.0</td>
<td>30</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>C114</td>
<td>Run 1: 33</td>
<td>63</td>
<td>30</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 33</td>
<td>63</td>
<td>30</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 37.5</td>
<td>67.5</td>
<td>30</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>C153</td>
<td>Run 1: 50.0</td>
<td>92.0</td>
<td>30</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 50.0</td>
<td>92.0</td>
<td>30</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 51.5</td>
<td>93.0</td>
<td>30</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>C123</td>
<td>Run 1: 74</td>
<td>69.0</td>
<td>15</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 74</td>
<td>69.0</td>
<td>15</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 76</td>
<td>81.0</td>
<td>15</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td>C133</td>
<td>Run 1: 58.0</td>
<td>76.0</td>
<td>20</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 58.0</td>
<td>76.0</td>
<td>20</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 62.5</td>
<td>82.0</td>
<td>20</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>C113</td>
<td>Run 1: 63.0</td>
<td>58.0</td>
<td>5.0</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 63.0</td>
<td>58.0</td>
<td>5.0</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 63.0</td>
<td>58.0</td>
<td>5.0</td>
<td>86</td>
<td></td>
</tr>
<tr>
<td>C132</td>
<td>Run 1: 63</td>
<td>78</td>
<td>15</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 63</td>
<td>78</td>
<td>15</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 63</td>
<td>78</td>
<td>15</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>C112</td>
<td>Run 1: 84.0</td>
<td>96.0</td>
<td>15</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 84.0</td>
<td>96.0</td>
<td>15</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 86.0</td>
<td>101.0</td>
<td>15</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>C122</td>
<td>Run 1: 94.0</td>
<td>99.0</td>
<td>5.0</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 94.0</td>
<td>99.0</td>
<td>5.0</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 96.0</td>
<td>101.0</td>
<td>5.0</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>C111</td>
<td>Run 1: 99.0</td>
<td>101.0</td>
<td>10</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 99.0</td>
<td>101.0</td>
<td>10</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 101.0</td>
<td>101.0</td>
<td>10</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>OrderC</td>
<td>Run 1: 119.0</td>
<td>121.0</td>
<td>10</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 2: 119.0</td>
<td>121.0</td>
<td>10</td>
<td>121</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Run 3: 111.1</td>
<td>111.1</td>
<td>10</td>
<td>121</td>
<td></td>
</tr>
</tbody>
</table>

Figure VIII.20

Performance of Component of Product C (Detail Level)
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>D214</td>
<td>Run 1: 43</td>
<td>73</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Run 2: 43</td>
<td>73</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Run 3: 41.5</td>
<td>71.5</td>
<td>30</td>
<td>30</td>
<td>81</td>
</tr>
<tr>
<td>D224</td>
<td>Run 1: 43</td>
<td>78</td>
<td>35</td>
<td>35</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Run 2: 43</td>
<td>78</td>
<td>35</td>
<td>35</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Run 3: 41.5</td>
<td>76.5</td>
<td>35</td>
<td>35</td>
<td>85</td>
</tr>
<tr>
<td>D234</td>
<td>Run 1: 43</td>
<td>73</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Run 2: 43</td>
<td>73</td>
<td>30</td>
<td>30</td>
<td>83</td>
</tr>
<tr>
<td></td>
<td>Run 3: 41.5</td>
<td>71.5</td>
<td>30</td>
<td>30</td>
<td>81</td>
</tr>
<tr>
<td>D253</td>
<td>Run 1: 48</td>
<td>83</td>
<td>35</td>
<td>35</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 2: 48</td>
<td>83</td>
<td>35</td>
<td>35</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 3: 45.5</td>
<td>81.5</td>
<td>35</td>
<td>35</td>
<td>96</td>
</tr>
<tr>
<td>D223</td>
<td>Run 1: 53.0</td>
<td>88.25079</td>
<td>25</td>
<td>25</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Run 2: 53.0</td>
<td>88.25079</td>
<td>25</td>
<td>25</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>Run 3: 61.5</td>
<td>86.25055</td>
<td>25</td>
<td>25</td>
<td>86</td>
</tr>
<tr>
<td>D213</td>
<td>Run 1: 73.0</td>
<td>83.25433</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 2: 73.0</td>
<td>83.25433</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 3: 71.5</td>
<td>81.55826</td>
<td>10</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>D283</td>
<td>Run 1: 73.0</td>
<td>83.59579</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 2: 73.0</td>
<td>83.59579</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 3: 71.5</td>
<td>81.59615</td>
<td>10</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>D243</td>
<td>Run 1: 79.0</td>
<td>88.072304</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 2: 79.0</td>
<td>88.072304</td>
<td>10</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Run 3: 79.5</td>
<td>86.59915</td>
<td>10</td>
<td>10</td>
<td>96</td>
</tr>
<tr>
<td>D212</td>
<td>Run 1: 98.0</td>
<td>103.0</td>
<td>5.0</td>
<td>5.0</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Run 2: 98.0</td>
<td>103.0</td>
<td>5.0</td>
<td>5.0</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Run 3: 96.0</td>
<td>101.0</td>
<td>5.0</td>
<td>5.0</td>
<td>101</td>
</tr>
<tr>
<td>D211</td>
<td>Run 1: 103.0</td>
<td>108.04394</td>
<td>5.0</td>
<td>5.0</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Run 2: 103.0</td>
<td>108.04394</td>
<td>5.0</td>
<td>5.0</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Run 3: 101.0</td>
<td>106.01049</td>
<td>5.0</td>
<td>5.0</td>
<td>101</td>
</tr>
<tr>
<td>OrderD</td>
<td>Run 1: 108.04394</td>
<td>118.04394</td>
<td>10.07719</td>
<td>10</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>Run 2: 108.04394</td>
<td>118.05173</td>
<td>10.07719</td>
<td>10</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>Run 3: 106.01049</td>
<td>116.06454</td>
<td>10.05146</td>
<td>10</td>
<td>116</td>
</tr>
</tbody>
</table>

Figure VIII.21
Performance of Component of Product D (Detail Level)
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>E215</td>
<td>Run 1</td>
<td>49</td>
<td>79</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>48</td>
<td>78</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>58.0</td>
<td>83.0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>E312</td>
<td>Run 1</td>
<td>60.0</td>
<td>101.0</td>
<td>32.0</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>60.0</td>
<td>101.0</td>
<td>32.0</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>108.5</td>
<td>138.5</td>
<td>35.5</td>
<td>30</td>
</tr>
<tr>
<td>E512</td>
<td>Run 1</td>
<td>53.0</td>
<td>83.0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>60.0</td>
<td>104</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>72.0</td>
<td>103.0</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>E214</td>
<td>Run 1</td>
<td>70.0</td>
<td>99.14706</td>
<td>20.0</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>79.0</td>
<td>99.14706</td>
<td>20.0</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>102.2763</td>
<td></td>
<td>20.2763</td>
<td>20</td>
</tr>
<tr>
<td>E222</td>
<td>Run 1</td>
<td>89.0</td>
<td>104</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>89.0</td>
<td>104</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>108.0</td>
<td></td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>E611</td>
<td>Run 1</td>
<td>54.0</td>
<td>114</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>54.0</td>
<td>114</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>96.0</td>
<td></td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>E311</td>
<td>Run 1</td>
<td>120.0</td>
<td>125.0794</td>
<td>15.0</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>120.0</td>
<td>125.0794</td>
<td>15.0</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>120.0</td>
<td>136.0297</td>
<td>15.0297</td>
<td>15</td>
</tr>
<tr>
<td>E223</td>
<td>Run 1</td>
<td>99.0</td>
<td>130.0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>99.0</td>
<td>130.0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>103.0</td>
<td></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>E512</td>
<td>Run 1</td>
<td>104.0</td>
<td>114</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>104.0</td>
<td>114</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>168.0</td>
<td></td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>E212</td>
<td>Run 1</td>
<td>130.0</td>
<td>135.0</td>
<td>5.0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>130.0</td>
<td>135.0</td>
<td>5.0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>130.0</td>
<td></td>
<td>5.0</td>
<td>5</td>
</tr>
<tr>
<td>E211</td>
<td>Run 1</td>
<td>135.0</td>
<td>149.09167</td>
<td>5.091675</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>135.0</td>
<td>149.09167</td>
<td>5.091675</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>135.0</td>
<td>141.02323</td>
<td>5.02323</td>
<td>5</td>
</tr>
<tr>
<td>OrderE</td>
<td>Run 1</td>
<td>140.0</td>
<td>150.11148</td>
<td>10.019006</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 2</td>
<td>140.0</td>
<td>150.11148</td>
<td>10.019006</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Run 3</td>
<td>141.02323</td>
<td></td>
<td>10.02323</td>
<td>10</td>
</tr>
</tbody>
</table>

Figure VIII.22

Performance of Component of Product E (Detail Level)
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>F511</td>
<td>Run 1: 55</td>
<td>120</td>
<td>65</td>
<td>65</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 2: 55</td>
<td>120</td>
<td>65</td>
<td>65</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 3: 62.5</td>
<td>137.5</td>
<td>65</td>
<td>65</td>
<td>141</td>
</tr>
<tr>
<td>F312</td>
<td>Run 1: 70</td>
<td>105.0</td>
<td>35.0</td>
<td>30</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Run 2: 70</td>
<td>105.0</td>
<td>35.0</td>
<td>30</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Run 3: 77.5</td>
<td>109.5</td>
<td>32.5</td>
<td>30</td>
<td>121</td>
</tr>
<tr>
<td>F612</td>
<td>Run 1: 70</td>
<td>105</td>
<td>25</td>
<td>35</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Run 2: 70</td>
<td>105</td>
<td>25</td>
<td>35</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Run 3: 77.5</td>
<td>112.5</td>
<td>35</td>
<td>35</td>
<td>126</td>
</tr>
<tr>
<td>F822</td>
<td>Run 1: 70</td>
<td>105</td>
<td>35</td>
<td>35</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Run 2: 70</td>
<td>105</td>
<td>35</td>
<td>35</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>Run 3: 77.5</td>
<td>112.5</td>
<td>35</td>
<td>35</td>
<td>126</td>
</tr>
<tr>
<td>F411</td>
<td>Run 1: 80</td>
<td>115</td>
<td>35</td>
<td>35</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 2: 80</td>
<td>115</td>
<td>35</td>
<td>35</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 3: 81.5</td>
<td>122.5</td>
<td>35</td>
<td>35</td>
<td>141</td>
</tr>
<tr>
<td>F811</td>
<td>Run 1: 95</td>
<td>115</td>
<td>29</td>
<td>20</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 2: 95</td>
<td>115</td>
<td>29</td>
<td>20</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 3: 102.5</td>
<td>122.5</td>
<td>20</td>
<td>20</td>
<td>141</td>
</tr>
<tr>
<td>F311</td>
<td>Run 1: 116.0</td>
<td>131.0 7367</td>
<td>15.072958</td>
<td>15</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 2: 116.0</td>
<td>131.0 7367</td>
<td>15.072958</td>
<td>15</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 3: 121.0</td>
<td>130.03604</td>
<td>15.095441</td>
<td>15</td>
<td>141</td>
</tr>
<tr>
<td>F611</td>
<td>Run 1: 105</td>
<td>115</td>
<td>10</td>
<td>10</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 2: 105</td>
<td>115</td>
<td>10</td>
<td>10</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Run 3: 112.5</td>
<td>122.5</td>
<td>10</td>
<td>10</td>
<td>141</td>
</tr>
<tr>
<td>OrderF</td>
<td>Run 1: 131.07787</td>
<td>141.10265</td>
<td>10.02479</td>
<td>10</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>Run 2: 131.07787</td>
<td>141.10265</td>
<td>10.02479</td>
<td>10</td>
<td>146</td>
</tr>
<tr>
<td></td>
<td>Run 3: 126.09004</td>
<td>146.10114</td>
<td>10.052094</td>
<td>10</td>
<td>151</td>
</tr>
</tbody>
</table>

Figure VIII.23

Performance of Component of Product F (Detail Level)
<table>
<thead>
<tr>
<th>Product</th>
<th>Start Time</th>
<th>Finish Time</th>
<th>Processing Time</th>
<th>Lead Time</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>G214</td>
<td>Run 1: 50</td>
<td>83</td>
<td>25</td>
<td>25</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 2: 58</td>
<td>83</td>
<td>25</td>
<td>25</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>Run 3: 72</td>
<td>97</td>
<td>25</td>
<td>25</td>
<td>119</td>
</tr>
<tr>
<td>G224</td>
<td>Run 1: 78.0</td>
<td>98.00564</td>
<td>20.005641</td>
<td>20</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>Run 2: 78.0</td>
<td>98.00564</td>
<td>20.005641</td>
<td>20</td>
<td>122</td>
</tr>
<tr>
<td></td>
<td>Run 3: 92.0</td>
<td>112.17003</td>
<td>20.170028</td>
<td>20</td>
<td>131</td>
</tr>
<tr>
<td>G312</td>
<td>Run 1: 78.0</td>
<td>104.0</td>
<td>30</td>
<td>30</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 2: 78.0</td>
<td>104.0</td>
<td>30</td>
<td>30</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 3: 97</td>
<td>127.7</td>
<td>20.009997</td>
<td>30</td>
<td>141</td>
</tr>
<tr>
<td>G213</td>
<td>Run 1: 83</td>
<td>106</td>
<td>25</td>
<td>25</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 2: 83</td>
<td>106</td>
<td>25</td>
<td>25</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 3: 97</td>
<td>122.0</td>
<td>25</td>
<td>25</td>
<td>141</td>
</tr>
<tr>
<td>G223</td>
<td>Run 1: 98</td>
<td>122.0</td>
<td>10</td>
<td>10</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 2: 98</td>
<td>122.0</td>
<td>10</td>
<td>10</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>Run 3: 112</td>
<td>141</td>
<td>10</td>
<td>10</td>
<td>141</td>
</tr>
<tr>
<td>G411</td>
<td>Run 1: 98</td>
<td>113</td>
<td>15</td>
<td>15</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 2: 98</td>
<td>113</td>
<td>15</td>
<td>15</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 3: 112</td>
<td>127</td>
<td>15</td>
<td>15</td>
<td>151</td>
</tr>
<tr>
<td>G212</td>
<td>Run 1: 132.0</td>
<td>137.0</td>
<td>5.0</td>
<td>5</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Run 2: 132.0</td>
<td>137.0</td>
<td>5.0</td>
<td>5</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Run 3: 141.0</td>
<td>146.0</td>
<td>5.0</td>
<td>5</td>
<td>146</td>
</tr>
<tr>
<td>G311</td>
<td>Run 1: 132.0</td>
<td>142.00257</td>
<td>10.002561</td>
<td>10</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 2: 132.0</td>
<td>142.00257</td>
<td>10.002561</td>
<td>10</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 3: 141.0</td>
<td>151.07999</td>
<td>10.002561</td>
<td>10</td>
<td>151</td>
</tr>
<tr>
<td>G211</td>
<td>Run 1: 137.0</td>
<td>142.006714</td>
<td>5.006714</td>
<td>5</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 2: 137.0</td>
<td>142.006714</td>
<td>5.006714</td>
<td>5</td>
<td>142</td>
</tr>
<tr>
<td></td>
<td>Run 3: 145.0</td>
<td>151.06281</td>
<td>5.006714</td>
<td>5</td>
<td>145</td>
</tr>
<tr>
<td>OrderG</td>
<td>Run 1: 142.009714</td>
<td>152.1529</td>
<td>10.055756</td>
<td>10</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>Run 2: 142.009714</td>
<td>152.1529</td>
<td>10.055756</td>
<td>10</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>Run 3: 142.009714</td>
<td>152.1529</td>
<td>10.055756</td>
<td>10</td>
<td>152</td>
</tr>
</tbody>
</table>

Figure VIII.24: Performance of Component of Product G (Detail Level)
Figure VIII.25
Number of Parts in Shop1 Input Store

Figure VIII.26
Number of Parts in Shop2 Input Store

Figure VIII.27
Number of Parts in Shop3 Input Store
Figure VIII.28
Number of Parts in Shop4 Input Store

Figure VIII.29
Number of Parts in Shop5 Input Store

Figure VIII.30
Number of Parts in Shop6 Input Store
Figure VIII.31  Number of Parts in Assembly Input Store

Figure VIII.32  Number of Parts in Shop1 Working Area

Figure VIII.33  Number of Parts in Shop2 Working Area

372
Figure VIII.34
Number of Parts in Shop3 Working Area

Figure VIII.35
Number of Parts in Shop4 Working Area

Figure VIII.36
Number of Parts in Shop5 Working Area
Figure VIII.37  Number of Parts in Shop6 Working Area

Figure VIII.38  Number of Parts in Assembly Working Area

Figure VIII.39  Number of Parts in Shop1 Output Store
Figure VIII.40  Number of Parts in Shop2 Output Store

Figure VIII.41  Number of Parts in Shop3 Output Store

Figure VIII.42  Number of Parts in Shop4 Output Store
Figure VIII.43  
Number of Parts in Assembly Output Store

Figure VIII.44  
Histogram of Number in Input Store (shop2)

Figure VIII.45  
Histogram of Number in Work Area (shop2)
Figure VIII.46  
Histogram of Numberinoutputstore (shop2)

Figure VIII.47  
Queue Length at Machine2

Figure VIII.48  
Time in Queue at Machine2
Figure VIII.49  
Queue Length at Machine 3

Figure VIII.50  
Time in Queue at Machine 3

Figure VIII.51  
Queue Length at Machine 4
Figure VIII.52
Time in Queue at Machine 4

Figure VIII.53
Queue Length at Machine 5

Figure VIII.54
Time in Queue at Machine 5
Figure VIII.55 Queue Length at Machine8

Figure VIII.56 Time in Queue at Machine8