Simulation Modelling Software Approaches to Manufacturing Problems

Vlatka Hlupić

Thesis submitted in Fulfilment of the Requirement

for the Degree of Doctor of Philosophy

at the

London School of Economics and Political Science,

University of London

June, 1993

To Helena and Vlado, my dear parents, for their love and support for everything I do.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Professor Ray J.Paul for his excellent supervision and support throughout this research.

I am grateful to Professor Vlatko Čerić for introducing me to simulation.

I am thankful to Ms Lidia Jacobs and Ms Sajda Qureshi for their friendship and understanding during the last two years.

My greatest appreciation is devoted to Mr Hrvoje Dominic Vidjak, whose help and encouragement was crucial for the completion of this thesis on time.

ABSTRACT

Increased competition in many industries has resulted in a greater emphasis on developing and using advanced manufacturing systems to improve productivity and reduce costs. The complexity and dynamic behaviour of such systems, make simulation modelling one of the most popular methods to facilitate the design and assess operating strategies of these systems.

The growing need for the use of simulation is reflected by a growth in the number of simulation languages and data-driven simulators in the software market. This thesis investigates which characteristics typical manufacturing simulators possess, and how the user requirements can be better fulfilled.

For the purpose of software evaluation, a case study has been carried out on a real manufacturing system. Several simulation models of an automated system for electrostatic powder coating have been developed using different simulators. In addition to the evaluation of these simulators, a comprehensive evaluation framework has been developed to facilitate selection of simulation software for modelling manufacturing systems. Different hierarchies of evaluation criteria have been established for different software purposes. In particular, the criteria that have to be satisfied for users in education differ from those for users in industry.

A survey has also been conducted involving a number of users of software for manufacturing simulation. The purpose of the survey was to investigate users' opinions about simulation software, and the features that they desire to be incorporated in simulation software. A methodology for simulation software selection is also derived. It consists of guidelines related to the actions to be taken and factors to be considered during the evaluation and selection of simulation software.

On the basis of all the findings, proposals on how manufacturing simulators can be improved are made, both for use in education and in industry. These software improvements should result in a reduction in the amount of time and effort needed for simulation model development, and therefore make simulation more beneficial.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

The future offers us very little hope for those who expect that our new mechanical slaves will offer us a world in which we may rest from thinking. (Norbert Wiener)

Advanced manufacturing systems are being used increasingly in many industries in order to improve productivity and reduce costs. Because of the complexity, dynamic and stochastic behaviour of these systems, simulation modelling is becoming one of the most popular methods used to investigate their configuration alternatives and potential operation strategies (Hlupić and Paul, 1993a), (Ekere and Hannam, 1989), (Law and McComas, 1989).

The rising acceptance of simulation has resulted in a growth in the number of simulation languages and simulators in the software market. When a model is developed using a simulation language, the simulation analyst has to write a program using the modelling constructs of the language. On the other hand, a simulator allows the modelling of the problem with little or no programming, where the analyst has to provide data related to the system being modelled.

Although the existence of alternative software products is beneficial to simulation software users, this might become a problem when deciding which software package to choose. Whilst the selection of a suitable software product can result in significantly improved productivity and reduced manufacturing costs, the choice of an inadequate package can result not only in the loss of the actual purchase cost but also in the costly disruption of manufacturing processes (Ghanforoush *et al*, 1985) and planned simulation projects. In addition, despite continuous advances in simulation software products, they should be further improved in order to make simulation modelling easier, faster and more effective.

This thesis addresses the issues related to evaluation, selection and possible ways of improving manufacturing simulators. Several manufacturing simulators are evaluated on the basis of a case study carried out in a real manufacturing environment. A comprehensive evaluation framework is developed in order to assist selection of software for manufacturing simulation. A methodology for simulation software selection is derived as well as proposals for the improvement of manufacturing simulators.

Introduction

This chapter provides introductory information relating to the research presented in this thesis. Fundamental information regarding simulation and manufacturing systems is given in section 1.1. Section 1.2 addresses the main types of simulation software with an emphasis on data-driven simulators. Issues related to a case study carried out are presented in section 1.3. Research objectives are specified in section 1.4, whilst section 1.5 provides an outline of this thesis. A summary of this chapter is given in section 1.6.

1.1. SIMULATION AND MANUFACTURING

This section addresses the issues relevant to simulation modelling of manufacturing systems. Basic information about simulation modelling is presented in sub-section 1.1.1. Sub-section 1.1.2 considers advanced manufacturing systems. Reasons for using simulation for modelling advanced manufacturing systems are given in sub-section 1.1.3.

1.1.1. Computer Simulation Modelling

Simulation modelling is the process of designing a model of a real system and conducting experiments with this model for the purpose of either understanding the behaviour of the system or of evaluating various operating strategies of the system (Shannon, 1975). Although simulation can be done manually, it is usually referred to as computer simulation, because not many reasonable simulation studies can be carried out without the use of computers. In this context, Pidd (1992a) specifies the basic principles of computer simulation: "The analyst builds a model of the system of interest, writes computer programs which embody the model and uses a computer to imitate the system's behaviour when subject to a variety of operating policies. Thus, the most desirable policy may be selected".

The main types of simulation can be distinguished on the basis of changing the state of the system through time. The state of the system can be changed at discrete time points (discrete event simulation), it can be changed continuously (continuous simulation), or it can combine both discrete and continuous changes (combined discrete/continuous simulation). In this thesis the term 'simulation' is used to refer to discrete event simulation. In this context, manufacturing systems with discrete processes are primarily considered, although there are manufacturing systems which are suited to the other two types of simulation.

The process of simulation consists of several stages. This is shown in Figure 1.1 together with the main directions of feedback information. In practice the simulation process is dynamic and iterative. Individual stages provide feedback information to other stages. For example, model verification might indicate errors in the computer model, which means that further modifications of this model are needed. A practical implementation of these stages is illustrated in Chapter 3, which addresses a simulation case study carried out in a real manufacturing environment.

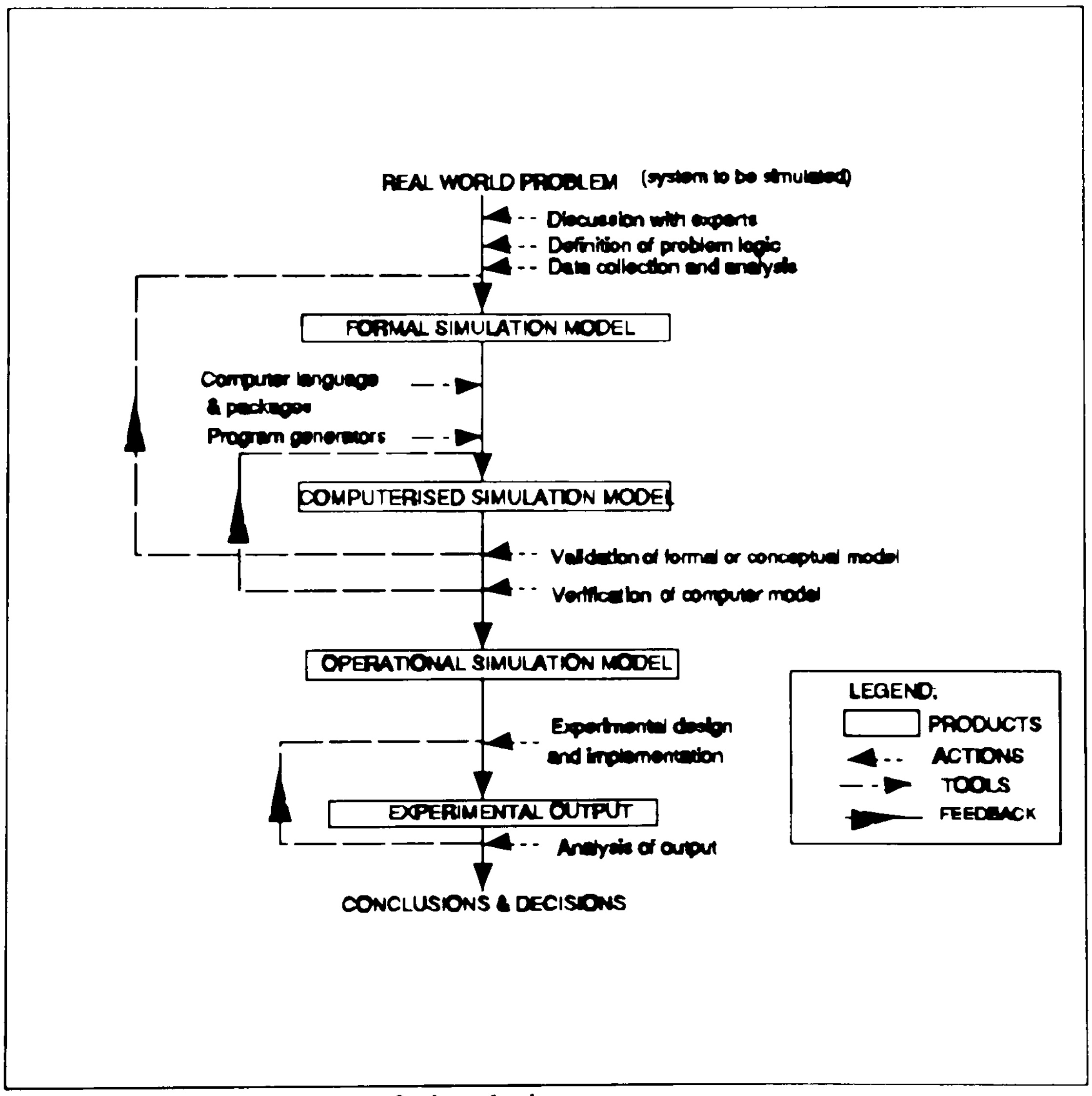


Figure 1.1 The stages of simulation process

1.1.2. Advanced Manufacturing Systems

A manufacturing system can be defined as a system in which raw materials are processed from one form into another, known as a product, gaining a higher or added value in the process (Parnaby, 1979). Since the beginning of the industrial revolution, and especially in the present competitive world, there has been a continuous attempt to improve manufacturing systems and make them more efficient. As a result of this, a number of new technologies and management concepts have emerged, generally known as Advanced Manufacturing Technology (AMT). AMT includes a variety of individual technologies such as Computer Aided Design (CAD), Computer-based production equipment, Group Technology (GT), Flexible Manufacturing Systems (FMS) and Computer Integrated Manufacturing (CIM). These technologies facilitate the following activities (Harrison, 1990):

(i) The transformation of materials through the physical operations of cutting, mixing, printing, fabrication and assembly.

(ii) The movement of materials by means of conveyors, robots, guided vehicles etc.

(iii) The examination and inspection of materials through the use of automated testing equipment.

(iv) The storage of materials and their fast retrieval.

(v) Product design in terms of shape and properties such as strength and weight.

(vi) Determining how a product should be manufactured.

(vii) Production management systems which schedule products and control the level of inventories.

Advanced manufacturing systems incorporate advanced manufacturing technologies, and aims to reduce operating costs, maintain high and consistent quality, and accommodate changes in product design (Hollocks, 1989).

1.1.3. Reasons for Simulation of Manufacturing Systems

Computer simulation has become the most widely used technique which facilitates the carrying out of experiments on models representing manufacturing systems (Kochhar, 1989). There are several reasons for using simulation in manufacturing environments, such as:

(i) Advanced manufacturing systems are very complex, and it is therefore difficult to predict their behaviour. Complexity is reflected in a variety of product types and quantities, diversity of production equipment, different possibilities for routing of parts, variety of operations etc. Due to such complexity, analytical methods are usually not appropriate for modelling these systems.

(ii) Advances in automation have resulted in manufacturing systems that involve large capital investments. This has engendered a need for careful modelling of any proposed system or change to an existing system (O'Keefe and Haddock, 1991).

(iii) Manufacturing systems are characterized by a stochastic behaviour. Various random events can occur such as equipment breakdown, variations in orders and machining times or blocking of transportation routes (Hlupić and Paul, 1992c). With the capability of sampling from statistical distributions, most simulation software products can cope well with randomness.

(iv) Manufacturing systems are dynamic, involving parallel activities. In these systems, discrete products move through the production process from raw material to the final product, which means that discrete event simulation should be used as the basic modelling paradigm (Heinonen at al, 1986).

(v) Advances in hardware, software, and simulation methodology have made simulation more accessible even by small sized companies.

(vi) The provision of visual interactive facilities in simulation software packages has resulted in a greater comprehension and application of simulation not only by simulation specialists, but also by production engineers and managers.

(vii) It is cheaper and easier to experiment with models (when they work), instead of experimenting with the real system.

There are many application areas and potential benefits of simulation in manufacturing environments (Hollocks, 1992). The most common applications and benefits obtained from simulation are identified by a study "Simulation in U.K. Manufacturing Industry" (Simulation Study Group, 1991). The information derived in this study, with regard to application areas and the benefits of simulation in manufacturing is summarized in Table 1.1.

Table 1.1 Application areas and benefits of using simulation in manufacturing

APPLICATION AREAS OF SIMULATION IN MANUFACTURING	BENEFITS OF SIMULATION IN MANUFACTURING
Plant layout and utilization Analyzing material control rules Analyzing required manning levels Short term scheduling and loading Capital equipment analysis Line balancing Inventory evaluation and control Information flow analysis Process definition and analysis	Risk reduction in managers' decision making Greater understanding of systems achieved by defining the manufacturing logic and supporting data Reducing operating costs by installation of right technologies Lead time reduction by determination of the appropriate operating strategies Reduction of capital costs by selecting the appropriate system configuration Faster configuration changes achieved by experimentation

1.2. SIMULATION SOFTWARE

This section provides basic information about simulation software. Since data-driven manufacturing simulators are the main subject of this research, this type of simulation software is additionally described. A classification of simulation software is presented in sub-section 1.2.1, whilst sub-section 1.2.2 provides further information about data-driven simulators.

1.2.1. Classification of Simulation Software

There are many different ways of classifying simulation software. Pidd (1992a) classifies simulation software in seven groups: general purpose languages (eg. FORTRAN, Pascal and C), pre-written libraries (eg. GASP, SIMON and FORSSIGHT), simulation programming languages (eg. SIMSCRIPT, SIMAN and SIMULA), flow diagram systems (eg. GPSS and HOCUS), program generators (eg. CAPS/ECSL, VS7 and DRAFT), visual interactive simulation systems (eg. GENETIK, SIMAN/CINEMA and SEE-WHY), and visual interactive modelling systems (eg. WITNESS, SIMFACTORY, ProModelPC and XCELL+).

A more general classification is proposed by Law and Kelton (1991), according to whom simulation packages can be generally classified as simulation languages and simulators. When a simulation language is used, the model is developed by writing a program using the modelling constructs of a language. This enables modelling of almost any type of system, but it might be tedious and time consuming. Simulation languages are general in nature, although some of them have special features for modelling manufacturing systems. For example, SIMAN and SLAM II have manufacturing modules for automated guided vehicles and conveyors.

On the other hand, a simulator allows the modelling of a specific class of systems with little or no programming, as it is a data driven environment for a limited problem domain. When a simulator is used for model development, models are typically developed by the specification of model parameters via menus. As little or no programming is needed, modelling time is usually significantly reduced. In this thesis, the term 'simulator' is adopted to represent data-driven manufacturing simulators (or visual interactive modelling systems in the context of Pidd's classification).

1.2.2. Data-driven Simulators

A simulator is a parameter driven system. The user only has to provide data to a simulator, instead of programming. These data together with simulation logic of a simulator form a basis for development of computer model. Some simulators enable additional programming in order to model specific logical features of the system under consideration. Although there are several general purpose data-driven simulators available on the market (eg. HOCUS and VS7), most of them are domain specific or generic (Carrie, 1988), designed to simulate a special class of systems.

There are many simulators that incorporate logic specific to manufacturing systems (eg. WITNESS, SIMFACTORY II.5, XCELL+, ProModelPC, AutoMod II etc.). Logical constructs within these simulators correspond to typical physical elements of manufacturing systems, and to their connection, interaction and behaviour. For example, a user has to provide only data relating to the number, type and performance of machines, operators, and materials handling system (if they are supplied within the simulator), and a simulation model can be quickly completed and run. If a system to be modelled does not fit within the logic of such a simulator, despite all approximations and ingenuity of the user, it is not possible to utilize the advantages of using simulators. In such a case other software tools will need to be used, most likely a simulation language. Although there might be many differences between data-driven simulators, they are all characterized by a common structure. The main components of a data-driven simulator are shown in Figure 1.2, as specified by Pidd (1992c).

The simulation model is produced on the basis of data provided by the user, and the simulation logic of a simulator. The simulation logic handles the change of system's state, according to the predefined operating modes (process, event, activity or three phase based). The model configurator controls and stores data entered by the user either graphically or textually. A general library stores routines which perform general simulation tasks such as scheduling of events or handling a list of future events. Sampling routines deal with randomness providing, for example, random number generation and sampling from statistical distributions. A graphics library provides facilities for graphical displays of the simulation model or of simulation results. A filer

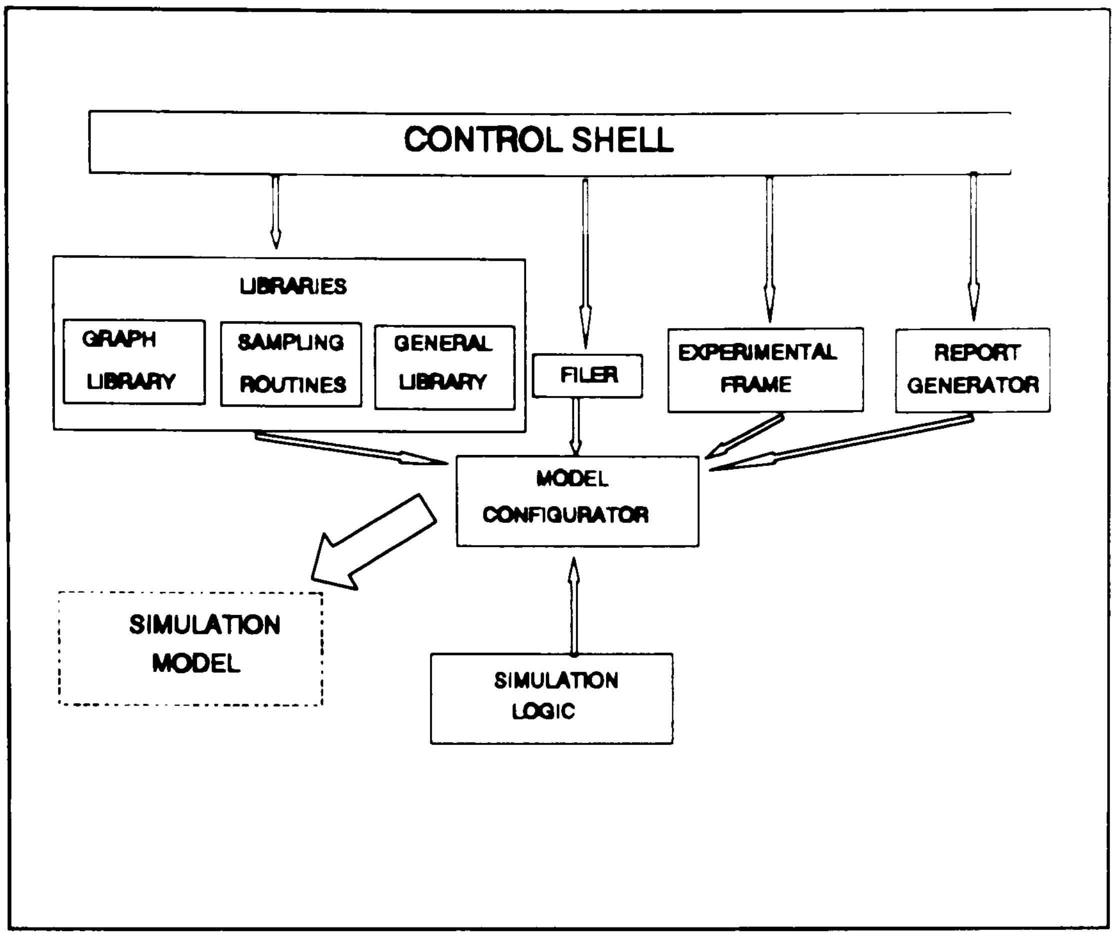


Figure 1.2 Components of a data driven simulator

handles the files which store model data or simulation results. An experimental frame deals with experimentation aspects of simulation such as specification of simulation parameters or the user's interactions. A report generator provides simulation reports during a run-time interaction or after a simulation run. Finally, a control shell performs an overall control of the system providing, for example, menus, help for the users, or error checks.

1.3. THE CASE STUDY

This section provides introductory information related to a case study carried out in a real manufacturing environment. The main reasons for a case study approach are discussed in sub-section 1.3.1. Factories visited during the search for an appropriate system for study are briefly described in sub-section 1.3.2. The main information about BICC-VERO Electronics, and the system being modelled is given in sub-section 1.3.3.

1.3.1. Reasons for a Case Study Approach

It was decided to take a case study approach at an early stage in this research. The main reason for this is a belief that any theoretical research can be improved by combining it with practice. The intention was to apply simulation methodology to model a real-life manufacturing system, derive results that could be utilized in practice, and to further develop research on the basis of the case study experience and findings. A more detailed specification of the reasons for this approach is embodied within the case study objectives presented in Chapter 3.

1.3.2. Factories Visited

It was not easy to find a suitable company for a case study. Many companies were contacted, and finally appointments were made to visit four factories around England. The first company visited was Yamazaki Mazak Machinery U.K. in Worcester, a Japanese owned leading manufacturer of CNC Machine Tools. The main characteristic of this impressive company is an implementation of the most advanced manufacturing technology, such as CAD, CIM, FMS with Automated Guided Vehicles (AGVs), Just-in-time manufacturing (JIT), Total Quality Control (TQC), an Automated Storage and Retrieval System (AS/RS), and an automated tool management system. Because of such high automation, the outstanding success and productivity of this company, and the utilization of simulation by some of its employees, it became obvious that there was not much additional simulation research to be done. The second factory visited was York International in Basildon, a manufacturer of heating, ventilation, air conditioning and refrigeration facilities. Although this company is not as highly automated as the previous one, it utilizes advanced manufacturing technologies such as JIT manufacturing, CIM with a Direct Numerical Control (DNC) system, Simultaneous Engineering and Statistical Process Control (SPC). Most of these technologies have been introduced recently and no major problems were evident.

The next company visited was Fred Lawton & Son Limited in Huddersfield, a manufacturer of carpet yarn. Although up-to-date machinery in the textile industry was utilized, this company is the least automated among the factories visited due to the nature of its production processes. In addition, the majority of production processes were not suitable for discrete event simulation modelling (for example, wool is continuously released to the premixing area, from which it is distributed to a carding machine, then to a spinning machine etc.). Due to these facts, this system was not considered as suitable for a case study.

The fourth company visited was BICC-VERO Electronics Limited in Eastleigh, a manufacturer of components for the electronics, electrical and communications industries. As the manufacturing processes in this company are fairly automated and suitable for discrete simulation, and because there were perceived to be problems in one part of the factory (the powder coating system), this company was chosen as a basis for a case study.

Whilst one company was selected for the case study, the detailed examination of the other three companies prior to selection provided an extensive backcloth to the research. This enabled broader conclusions to be drawn than the experience of just one company might have justified.

1.3.3. BICC-VERO Electronics Limited - Eastleigh

BICC-VERO Electronics Limited in Eastleigh, manufactures several thousand different types of products such as racks, cases, prototyping boards, connectors and cable carriers. The company utilizes advanced manufacturing technologies such as JIT manufacturing, CAD/CAM with CNC machines and computerized stock system.

Metal components of the manufactured products are made in different production cells. The last processing stage of these components is electrostatic powder coating in the powder coating system. This system is regarded as a bottleneck, because its productivity is smaller than in other parts of the factory. Due to this fact, and because simulation modelling has not been utilized in this company before, the managers welcomed the initiation of this thesis research. It was therefore decided to select this system for a case study. A detailed description of the powder coating system is given in Chapter 3.

1.4. RESEARCH OBJECTIVES

The aim of this thesis is to examine simulation modelling software approaches to manufacturing problems. In particular, to investigate issues related to the evaluation, selection and improvement of software for manufacturing simulation. This should result in more effective software selection, better satisfaction of the users' requirements for this type of software, and hence lead to a reduction in the amount of time, costs, and effort needed for simulation models development.

To accomplish these objectives, a case study was carried out at BICC-VERO Electronic, Eastleigh. This study involves using different manufacturing simulators to model a real manufacturing system in order to analyze these simulators, and to perceive the difficulties faced in modelling complex real systems. In addition, the experience acquired facilitated the establishment of an evaluation framework for manufacturing simulation software.

A survey was conducted involving a number of simulation practitioners both at universities and in industry. The purpose of the survey was to discover the users' requirements for manufacturing simulation software, and to perceive which features they require to be included in this software.

It was furthermore intended to derive a methodology for selecting simulation software. This should provide practical guidelines for the actions to be taken and the factors to be considered prior to final selection and purchasing of simulation software.

The final objective was to make proposals for the improvement of manufacturing simulators, according to their purpose. As every software product is unique, some of the improvement proposals would be more suitable for some particular software than for the others. Nevertheless, each individual improvement, when appropriately applied, should lead to faster and easier model development, augmenting the benefits of simulation modelling.

Several assumptions underline this research. With regard to simulation modelling in general, I am aware that the simulation process is adaptive, flexible, and iterative. Problem understanding is facilitated through several stages of the simulation process, each of which is provided with feedback information from other phases.

Considering simulation software, I believe that it should be easy to use, user-friendly, visual interactive, and flexible. Software evaluation in general should be structured, standardized, objective, and based on practical experience in using particular software products.

With regard to the nature of systems to be simulated, the main types of systems considered relate to manufacturing systems with discrete processes. In these systems parts are moved through the network of machines, storage areas, conveyors and other types of production equipment, until they become a final product. Several features typical of manufacturing characterize these systems such as part attributes, part loading and scheduling, shift patterns, breakdown of production equipment, labour requirements, inspection operation etc.

The summary of the thesis objectives is as follows:

(i) To identify essential features of manufacturing simulation software needed for the purpose of education or of modelling in industry.

(ii) To develop evaluation criteria for the assessment of software for manufacturing simulation.

(iii) To evaluate several widely used manufacturing simulators in terms of the established criteria on the basis of a case study in a real manufacturing environment.

(iv) To additionally investigate the users' requirements for manufacturing simulation software by conducting a survey involving both users at universities and users in industry.

(v) To derive a methodology for simulation software selection, which could facilitate this process and make it more efficient.

(vi) To determine how manufacturing simulators can be further improved in order to better satisfy users' requirements and make the simulation process more productive.

1.5. OUTLINE OF THE THESIS

This thesis addresses simulation modelling software approaches to manufacturing problems. In particular, it investigates which features manufacturing simulators should possess, and examines how they can be effectively selected and further improved. The thesis is structured as follows. Chapter 2 discusses background research material, which has been studied and used as a basis for establishing research objectives.

Chapter 3 presents a case study carried out in a real manufacturing environment, which has resulted in several simulation models. Chapter 4 derives a framework for the evaluation of simulation software, including over 310 evaluation criteria. An actual evaluation of several manufacturing simulators is presented in Chapter 5. Chapter 6 describes a survey conducted in order to find out the users' requirements for simulation software, derives a methodology for simulation software selection, and determines ways of improving manufacturing simulators. Chapter 7 summarizes the previous six chapters, draws conclusions from them and determines the lines for future research.

1.6. SUMMARY

This chapter provides introductory information with regard to this thesis. It gives a background for research presented in this thesis, and establishes the objectives of this research. The main points related to the simulation of manufacturing systems are addressed. The main types of simulation software are presented, with an emphasis on data-driven simulators, which are the main subject of this research. The case study carried out in a manufacturing company is introduced, together with information about other companies considered for research. Finally, the research objectives have been specified.

Despite all advances in hardware, software and simulation methodology itself, it still takes too long to develop and debug simulation models of considerable complexity. In addition, the amount of time that needs to be invested in learning and mastering simulation software is seen as too great, and the cost of the software is too high (Simulation Study Group, 1991). These facts, in addition to the findings derived from the analysis of the background research material in Chapter 2, support a need for further research in the area of manufacturing simulation software approaches.

CHAPTER 2. BACKGROUND RESEARCH MATERIAL

2.1. INTRODUCTION

Increased competition in many industries has resulted in greater emphasis on developing and using automated manufacturing systems in order to improve efficiency and reduce costs (Hlupić and Paul, 1992a). Due to the complexity and dynamic behaviour of such systems, simulation modelling is becoming one of the most popular methods of facilitating their design and appraising operating strategies.

The growing need for the use of simulation is reflected by a growth in the number of simulation software products in the software market. Although simulation software for manufacturing applications has many characteristics in common with simulation software designed for general purpose or other specific application areas, there are some special features that make manufacturing simulators unique. Some of these features include modelling of material handling systems (Law and Kelton, 1991), special types of machines, part routing etc. An extensive list of these features is provided in Chapter 4.

This chapter provides a description of the literature used in carrying out the research presented in this thesis. Research studies related to the use of simulation in manufacturing environments, and especially studies regarding simulation software are analyzed.

The chapter is structured as follows. The application of simulation in advanced manufacturing environments is addressed with examples of publications that review a number of different simulation studies, carried out in order to facilitate the solving of different problems that arise in advanced manufacturing systems. These survey studies illustrate the popularity of simulation and its strength in approaching manufacturing problems.

The next section deals with research studies related to simulation software. This section is the core presentation of background research material, because it relates most closely to the subject of the research in this thesis. It contains publications sub-divided in several groups, according to the main focus of a particular study. For example, studies that relate to the evaluation of simulation software are separated from those that address simulation software selection or describe surveys. Nevertheless, overlaps between some of the groups is inevitable. For example, there are studies presenting software evaluation on the basis of information obtained from a survey, whilst some studies that focused on the use of a certain method for software selection also provide a list of evaluation criteria.

The penultimate section of the chapter gives a critique of the literature presented, whilst the last section provides a summary of findings revealed during the analysis of the background research material.

2.2. SIMULATION IN ADVANCED MANUFACTURING ENVIRONMENTS

The number of simulation applications in advanced manufacturing environments is constantly increasing. Evidence can be found in literature both in descriptions of specific case studies and in surveys of manufacturing simulation studies.

Publications containing reviews and surveys which present the use of simulation for solving various problems that arise in manufacturing environments are reviewed below. Some of these studies analyze some other aspects such as mathematical or computer modelling of manufacturing systems, using artificial intelligence techniques in addressing management problems etc.

Kochhar and Ma (1989a) depict the major characteristics of simulation studies carried out in order to make decisions relating to production management problems and to assess the resulting benefits. Simulation applications are classified according to the control level of the manufacturing system (management control or production control), or according to the types of manufacturing systems being simulated (flexible manufacturing systems or just-in-time manufacturing systems).

O'Grady and Menon (1986) present a review of flexible manufacturing systems and FMS literature. Part of this paper relates to a description of solving FMS planning and control problems using one of the following techniques: simulation, queuing theory, integer programming or heuristic algorithms. The authors propose that the purpose of simulation modelling has been one of the following: to establish the viability of a given FMS configuration of machine and transport devices, to assist the system design process with respect to hardware choices, or test operational planning and control strategies.

Singhal *et al* (1987) discuss how models can play a major role in design and control of complex automated manufacturing systems. They describe publications related to the applications of various operational research methods for solving problems in automated manufacturing systems, such as system design, production planning, scheduling and control, steady state operations and system improvements. Methods described include queuing network theory, simulation and artificial intelligence techniques.

Several survey papers describe simulation research in production scheduling. Kiran and Smith (1983) report on numerous simulation studies carried out for production scheduling purposes. They classify all studies in two categories: studies comparing and/or developing scheduling rules which will give good performance under a given set of criteria, and studies investigating the sensitivity of manufacturing performance to changing of production parameters under a given set of scheduling rules.

Ramasesh (1990) provides a state-of-the-art survey of simulation-based research on dynamic job shop scheduling. A number of different simulation studies are described with a focus in their findings on the job shop performance measures such as time-based measures, work-in-process measures, due-date related measures or cost-based measures.

In the context of current and future issues concerning FMS scheduling, Hutchison (1991) discusses several simulation studies which were used in order to improve the performance of flexible manufacturing systems. A classification framework is provided that facilitates the identification of FMS types and types of scheduling strategies, as well as explaining interactions between these categories.

The above survey publications are chosen as an illustration of the extensive use of simulation as an analysis tool in the design and operation of manufacturing systems. With the increasing use of simulation, the number of simulation software tools is also increasing. As a consequence of this, the number of research studies related to analysis, evaluation and selection of simulation software is also growing, which is addressed in the next section.

2.3. RESEARCH STUDIES RELATING TO SIMULATION SOFTWARE

This section is a part of the presentation of background research material that is the most relevant for the research presented in this thesis. It contains a summary analysis of various research studies related to simulation software. Although many studies combine different aspects of research such as evaluation, selection, or descriptions of simulation software, they are grouped according to the main focus of a particular research topic.

Those studies which propose criteria for the evaluation of manufacturing simulation software and/or desirable features of this type of software are presented in sub-section 2.3.1. Sub-section 2.3.2 includes a number of studies regarding the evaluation of different simulation software tools. An analysis of simulation software selection is the main subject of publications presented in sub-section 2.3.3, whilst sub-section 2.3.4 contains the publications regarding survey studies. Finally, sub-section 2.3.5 comprises some publications related to simulation software descriptions and tutorials.

2.3.1. Simulation Software Evaluation Criteria

Simulation models of real manufacturing systems are often large and complex, requiring a considerable time and effort for their development, verification and experimentation. Because of this, the facilities provided in the available simulation software tools are important. Studies presented below address this issue, providing criteria for the evaluation of simulation software tools in general as well as the requirements for manufacturing simulation software.

(i) Kochhar and Ma (1989b) address the essential and desirable features of simulation software for its effective use in manufacturing environments, providing the criteria which should be used for the selection of manufacturing simulation software tools. These criteria relate to *modelling assistance* provided, *interactivity*, *graphics* and the *data handling capability*. Other proposed criteria include the *time scale for model development*, *the learning curve* and *the required skills* for the use of software, *ease of model editing*, *portability*, *simulation speed* and *interfacing* the simulation package *with external systems*. The study concludes with a remark that the final decision to select a particular software tool must be based on the requirements of the organisation, the applications for which it will be used, and the skills of the users.

(ii) Addressing the issues related to simulation software products for analyzing manufacturing systems, Haider and Banks (1986) establish the following desirable features for simulation software. *Input flexibility* should enable model development either in a batch mode or in an interactive graphical environment.

Syntax used in software should be user-friendly, consistent and unambiguous. *Structural modularity* should allow modular model development. *Modelling flexibility* should allow interfacing with lower level programming languages to handle specific logical features. *Modelling conciseness* should be achieved by powerful and concise commands incorporated in the software. *Macro capability and hierarchical modelling* should allow a user to develop macros of manufacturing system components and develop models hierarchically. *Materials handling modules* which enable modelling of elements such as trucks, AGVs, conveyors and robots should be provided. *Standard statistics generation* should provide comprehensive statistics on standard measures automatically, or their generation should be simple to specify. *Data analysis* should enable analysis of input as well as the output data generated by the model, whilst *animation* should be included to facilitate debugging and communication to clients. Further desirable features include *interactive model debugging*, *micro/mainframe compatibility*, the *support* provided by the supplier, and the *cost* of simulation software.

(iii) In discussion about the role of simulation in designing and scheduling manufacturing systems, Grant (1988) provides a list of features which manufacturing simulators should possess. These features are classified in two groups: the first group contains software requirements for manufacturing simulators used for scheduling and control, whilst the second group contains characteristics that a simulator used for the design of manufacturing systems should contain. In order to be used as a tool for scheduling and control, a manufacturing simulator has to comprise the following features: *an effective user interface* (to facilitate the definition of the manufacturing model and to generate production schedules), an *implemented set of algorithms* for sequencing production orders, *interactivity*, an *interface to external data sources* (integration with database management system), a *mechanism to store all input and output data in a database*, and *fast execution of simulation* to respond to the needs of the production scheduler. Software requirements for simulators that facilitate the design of manufacturing systems include: a *language-based interface* (to build a model and generate performance reports for alternative designs), *user-designed and* *coded algorithms* associated with queue ranking procedures, *standard and user-coded performance reports*, *storage of data* used for model design *in an external file*, and *orientation to the design issues* including randomness.

(iv) Kochhar (1989) presents criteria for the assessment of manufacturing simulators. These criteria include: *the world views* (event, activity or process based) adopted by the simulator, *modelling assistance provided*, *interactive capability*, *animation facilities*, *data handling capability*, *learning curve*, *ease of use*, *portability* of simulation software, *simulation speed*, *reliability and service*, *flexibility* and *facilities for data recording and output results*. Furthermore, the author discusses the advantages of using software tools for manufacturing simulation with interactive and animation capabilities, as well as the benefits of combining artificial intelligence/expert systems with simulation. Interactivity is regarded as a means of eliminating or reducing demands on computer programming skills and simulation expertise in many simulation exercises. It is also mentioned that graphic and animated simulation enhance the credibility of simulation models and improve the utility of simulation results. On the other hand, artificial intelligence techniques can help the process of simulation model development by ensuring that the models are correct, logical and complete, or facilitate the design of experiments and the interpretation of simulation results.

(v) Bright and Johnson (1991) discuss the intrinsic nature of visual interactive modelling (VIM) software. Three main features of this type of software have been addressed: *speed and adaptability*, *width of application* and *ease of use*. Regarding speed and adaptability, they require fast development of models, which should quickly and easily adapt to any changes. On the other hand, there should be no constraints on the size or complexity of models. The width of application should be enhanced with an interface to other software systems such as data bases and general-purpose languages. Finally, software for visual interactive modelling should be easy to learn and use, providing as much help as possible during modelling. Their conclusions state that VIM software has not yet achieved the 'critical values' of user-friendliness and power.

(vi) Szymankiewicz *et al* (1988) specify desirable features of simulation software. They claim that many of these features are already established though no single simulation package currently provides all of them. The main features which should be more extensively used in simulation software include: *suitability for a wide range of problems, portability, lower price, ease of learning, additional programming, user-friendliness, integration with real-life control systems, built-in debugging facilities, high resolution graphics, panning and zooming, statistical facilities for multiple runs, and breakpoints setting capability.*

The above studies show which features of the simulation software are considered as important and as such should be incorporated into simulation software, and/or could be used for software evaluation. The actual software evaluation is addressed in the studies presented below.

2.3.2. Evaluation of Simulation Software

Publications presented in this sub-section relate to an explicit evaluation of various simulation software tools on the basis of certain criteria. Most of the studies concern the evaluation of software tools for manufacturing applications. Some studies have analyzed both special and general purpose simulation packages. A few earlier studies that evaluate only general purpose simulation languages are also included as an illustration of a long interest in research on simulation software evaluation.

(i) Several manufacturing simulators have been evaluated in terms of a number of different criteria by Banks *et al* (1991). Four manufacturing simulators are examined, with a remark that their features indicate the types of considerations involved in selecting software. The following manufacturing simulators have been evaluated: SIMFACTORY II.5, *XCELL+*, *WITNESS* and *ProModelPC*. The criteria for the evaluation are classified in five groups. The first group relates to the *basic features* such as routes, schedules, capacities, downtimes or transporters. The *robust features* (within the second group) include programming, conditional routing, part attributes, global variables and interface to other software. *Qualitative considerations* include ease of learning and using, the quality of the interface, animation and documentation, output reports, on-line help and system trace. Robots and cranes are mentioned within the section on *special constructs*.

Finally, the last criterion relates to the *cost* of the package. The evaluation reveals that SIMFACTORY II.5, WITNESS and ProModelPC are similar in their basic features, whilst XCELL+ does not model downtimes and requires the user to construct transporters and conveyors from available elements. Those simulators that were found to be similar differed in their operational procedures. Whilst in SIMFACTORY II.5 and ProModelPC, the complete route is specified directly on the screen, in WITNESS the user builds the route one step at a time when specifying other characteristics. SIMFACTORY II.5 and XCELL+ do not have robust features, whilst WITNESS and ProModelPC have most or all of them, enabling the user to add programming constructs to the model. The paper concludes with the statement that a simulator which can model every situation does not exist yet. However, simulators are considered as a step in the right direction, and as vendors realize their limitations, they can begin to improve them in succeeding versions of the software.

(ii) A critical evaluation of manufacturing-oriented simulation packages is provided by Law and Kelton (1991). They analyze *AutoMod II*, *ProModel*, *SIMFACTORY II.5*, *WITNESS* and *XCELL+*. Following an explanation of the major modelling elements of each package, their positive characteristics and shortcomings are addressed. For example, the main strength of AutoMod II is considered to be its three dimensional animation capability and a comprehensive set of material-handling modules. On the other hand, this package has very limited statistical capabilities, for example, a small number of input probability distributions and a lack of an easy mechanism for making multiple replications of experiments. ProModel is regarded as one of the most flexible simulators currently available, due to its programming-like constructs and its ability to call C or Pascal routines to model complex decision logic. They admit that the shortcomings are not currently known since ProModel is a relatively new product. However, they state that its animation is based on character graphics. The greatest advantages of SIMFACTORY II.5 are its ease of use and good statistical capabilities, eg. a variety of input probability distributions, automatic multiple replications of simulation experiments and confidence intervals for output measures. The main shortcoming is its inadequate modelling flexibility for certain manufacturing applications. WITNESS is regarded as a very flexible manufacturing simulator, due to its programming-like input/output rules and actions. As the main shortcoming, they mention the lack of an easy mechanism for making multiple replications of simulation, and an incorrect modelling of machine downtimes in the "calendar-time" approach (this approach allows a machine to break down when it is idle). According to this report, XCELL+ is easy to learn and use, with menus being employed to place and connect predefined graphical representations of system components. But, its statistical facilities are poor, there is no explicit modelling of transport and accumulating conveyors, and modelling flexibility is very limited.

(iii) A critical evaluation of the simulation languages *ECSL* and *SLAM* II is provided by Ekere and Hannam (1986). They claim that it is intrinsically difficult to compare simulation languages because it takes time to learn a language proficiently and a user's view of a language can be distorted by which language was learnt first. The following criteria have been used for evaluation and comparison: *static structure of the language, dynamic structure and system conceptualization, features of the language and constraints, language utilities, debugging and editing*, and *use for experimentation*. The results of evaluation reveal that ECSL is in many respects easier and simpler to use than SLAM II. On the other hand, SLAM II offers better data collection, data and result presentation facilities, and better error diagnostics.

(iv) Ekere and Hannam (1989) present an evaluation of the event, activity and process-based approaches for modelling manufacturing systems as well as an evaluation of three software tools for manufacturing simulation. They evaluate the simulation language *SLAM*, the program generator *CAPS/ECSL*, and the data driven simulation package *HOCUS*. The criteria specified for the evaluation of software features are classified into four categories. The first group relates to *model characterisation and programming*, which includes criteria such as precision of commands and syntax, programming effort and ease of use, static structuring features, dynamic structuring, algorithm capability, data manipulation, program readability, self-documentation, and language flexibility. The second group relates to *model development features* such as data type and logic diagnostics, ease of verification and debugging, and execution diagnostics. *Experimental and reporting features* comprise criteria such as input facilities, user interface, interactive use, model saving and restart, interactive graphic capability, built-in data collection, automatic standard report generation, and database interfacing. Finally, the fourth group concerns *commercial and technical* features such as availability and transportability, technical support, and documentation. The evaluation in terms of presented criteria reveals that SLAM's structure and syntax in the event version are not user-friendly, but it is a powerful language for modelling complex problems. On the other hand, the network version of SLAM is only suitable for models with limited complexity. ECSL is ranked highly with its code structure matching to a system's activity cycle diagram (ACD). Its ACD based structure and English-like syntax make the code of any problem readable and verifiable. Further features relate to its modularity and good facilities for static structure description. The menu-driven front-end entry of data provided by HOCUS is regarded to be easy to use, enabling the user to avoid learning the complexities of language syntax. The interactive graphical capabilities are judged as adequate, whilst the attribute manipulation options are regarded as limited.

(v) Law and Heider (1989) present a simulation software survey and evaluation on the basis of information provided by vendors. Twenty two software tools have been included in the evaluation. One half of these software tools relate to simulation languages (*AutoMod II, CADmotion, GPSS/PC, INSIGHT, PCModel, RESQ, SIMAN/Cinema, SIMPLE_1, SIMSCRIPT II.5, SLAM II* and *SLAMSYSTEM*), whilst the other half relates to manufacturing simulators (*FACTOR, HEI RTSS, InterFaSE, MAST, MIC-SIM, Micro SAINT, PROMOD, SIMFACTORY, STARCELL, WITNESS* and *XCELL+*). Six groups of criteria considered to be important for manufacturing simulation software are used for the analysis of the above mentioned software tools. *General features* of the software include modelling flexibility, part attributes, ease of model development, debugging aids, model execution speed, maximum model size, and portability. Desirable *animation* features comprise ease of development, creation of high-resolution icons, and smooth movement of icons across the screen. *Statistical capabilities* relate to a provision of standard distributions, user-defined distributions, multiple random number streams, automatic replications of experiments, specification of warm-up period, and providing confidence intervals for measures of performance. *Material handling modules* should include easy-to-use modules for modelling transporters, AGVs, conveyors, AS/RS, cranes, and robots. *Customer support* comprises training, technical support, and good documentation. Finally, criteria related to *output reports* include standard and user-defined reports, graphical displays of reports, and access to the individual model output observations. Instead of commenting on the presented information about the software, the paper concludes with a statement that there is no simulation package which is completely convenient and appropriate for all manufacturing applications.

(vi) A similar approach to software evaluation has been taken by Grant and Weiner (1986). They analyze several simulation software products (*AutoGram, BEAM, Cinema, Modelmaster, PCModel, RTCS, SEE WHY, SIMFACTORY, SIMPLE1* and *TESS)* with the main emphasis on their graphical and animation features. The analysis is done on the basis of information provided by vendors. The features examined are grouped in three main groups. *The simulation model building system* group includes the main orientation of the software and flexibility. *Animation graphics* related features determine the type of graphics and animation, and evaluate whether interactive display generation, zooming, panning, user created menus, and help screens are provided. Criteria within the *operational considerations* include the cost of the software, platforms on which software can be run and determination of need for a specialized VDU. The authors do not comment on the provided features of software tools. They conclude with a specification of general trends regarding simulation software tools, such as the implementation of software on microcomputers, manufacturing oriented preprocessors, lower priced systems, and interactivity both for model building and model animation.

(vii) Several FMS simulators have been examined by Bevans (1982). The following simulators are considered: *COL* (Carts On Line), *GCMS* (General Computerized Manufacturing System Simulation), *GFMS* (General Flexible Manufacturing Systems Simulator). *HABMS* (Advanced Batch Manufacturing System Model), *K&T FMS Simulator*, *MAST* (Manufacturing System Design Tool), *SPEED* and

Variable Mision Simulation Model. A little information about each simulator is provided. Most of this information is technical, such as *the length of source code, the language in which a simulator has been written, platforms on which simulator can be run, cost and availability*, information about the *vendor* and *latest revision*. Some other information is also provided such *the level of the detail* that can be accommodated by each simulator, *the types of FMS* that can be simulated and type of *documentation* provided. The main details about these simulators are not further discussed.

(viii) A comprehensive evaluation of fifteen simulation languages is provided by Cellier (1983). Languages examined are *ACSL, DARE-P, SIMNON, DYMOLA, SYSMOD, FORSIM-IV, SIMULA'67, PROSIM, SIMSCRIPT-II, GPSS_FORTRAN-II, GPSS_FORTRAN-III, SLAM-II, GASP-V, GASP-VI* and *COSY*. The evaluation criteria are classified in six groups regarding *expressiveness of the language, numerical behaviour, structural features, status of implementation, portability*, and *documentation*. Features within each group are assessed according to their availability and quality. Data presented is analyzed and software tools compared and ranked on the basis of evaluation. The results obtained indicate, for example, that ACSL should be used for continuous simulation, SYSMOD and SLAM-II for discrete simulation, and GASP-V for combined discrete-continuous simulation.

(ix) Perhaps one of the best known early simulation software evaluation and comparison was carried out by Tocher (1965). The simulation languages analyzed were: *GPSS, SIMPAC, SIMSCRIPT, SIMULA, CSL, ESP, GSP, MONTECODE* and *SIMON*. These languages are examined on the basis of the following groups of criteria: *the organization of time and activities in a simulation programming system, naming and structure of entities and generalized activity specification, testing of conditions in activities, test formation facilities, naming of variables in the simulation system, procedure facilities, sampling procedures, statistics collection procedures, output facilities, magnetic tape handling, initialization and simulation facilities*, and *development facilities*. It is estimated how well the languages under consideration satisfy the criteria within each group. Subsequently, each language is briefly described with an emphasis on its main qualities and weaknesses. The languages evaluated have not been ranked nor particular ones were recommended for use. The paper concluded that "...it is not possible to recommend the 'best buy', and the potential user cannot avoid the responsibility of studying the systems for himself".

The studies presented in this section relate to simulation software evaluation. The authors of many studies have chosen simulation software for manufacturing applications as a subject of research and evaluation. However, not many of them provide a critical evaluation of the software products under consideration nor suggest which particular software can be considered to be 'the best' to use.

2.3.3. Simulation Software Selection

Studies presented in this section generally relate to the selection of simulation software. They either provide general guidelines or approaches to simulation software selection, or demonstrate the use of a particular technique for software selection.

(i) Pidd (1989) provides some *general advice regarding selection of discrete simulation software*. Concerning the assessment of vendors' claims, the author warns of several facts that the potential users should be sceptical about. For example, one should not believe any vendor who claims that his product is better then everyone else's for any application or that the software can run on any computer under any operating system. In addition, when asked about the support they can provide in case of problems caused by bugs, the majority of vendors would probably deny the possibility of the existence of bugs. Furthermore, the author claims that the type of simulation software to be chosen depends on the intended application, and discusses which basic facilities should be provided in simulation software. Finally, general advice for simulation software selection is provided, which includes: development of a preliminary model of application, consideration of available resources and future applications, examination of the available software, and asking the vendors for assistance if possible.

(ii) *A structured approach to selection of simulation software* is proposed by Holder (1990). This approach suggests that software selection should commence with a consideration of the available resources within the organization, and a determination of the simulation objectives (potential users and types of applications). Subsequently, the essential features of the software are to be determined in order to eliminate software products that would certainly not be suitable. This should result in a 'short list' of products that are to be evaluated using the evaluation table provided. This table comprises evaluation features categorized in six groups: technical features, user needs (system development), user needs (end user), future development, functionality and commercial features. No weighting of the proposed criteria is established. These criteria are to be used to determine whether the products have the features required, and on the basis of this, a recommendation as to which software seems to be most suitable is to be derived. The paper concludes with the suggestion that vendors should be asked to assist both in the evaluation of software features, and in test modelling.

(iii) A discussion on *factors to consider before selecting manufacturing simulation software* is provided by Deaver (1987). He identifies a need to thoroughly analyze system requirements before selecting simulation software, as simulation packages vary widely in capability. Some of the factors that should be considered before simulation software selection include: identification of potential simulation users, consideration of future training for employees, determination of types of systems to be simulated, analyzing the resources currently available, and consideration of the amount of time that is to be dedicated to simulation. In addition, several criteria are presented that can be used for software evaluation. These criteria include features such as graphics, interaction, statistical data gathering and analysis, flexibility, support provided by vendor, ability for discrete-event and continuous-processes modelling, and ability to simulate both push and pull processes. The conclusions outline the benefits of simulation, if adequate data is provided.

(iv) Davis and Williams illustrate the evaluation and selection of simulation software using *the analytic hierarchy process method* (1993). They evaluate five simulation software systems using this method, in order to recommend suitable simulation software for a U.K. company. The chosen criteria include: cost, comprehensiveness of the system, integration with other systems, documentation, training, ease of use, hardware and installation, and confidence related issues (mainly regarding the reputation of the vendor). An illustration of the main phases of software evaluation and comparison using the analytic hierarchy process method is provided. In the first stage, the criteria are ranked according to their relative importance when selecting a simulation package. Several other steps follow, finally producing an overall ranking for each package being evaluated. It is emphasized that it is not possible to derive absolute measures of how well any package performs against a given criterion. Only its relative performance compared to the other packages can be obtained. The conclusions outline that the method used should be considered only as a decision aid, although the authors were satisfied with the overall results obtained.

(v) A simple *three step method for the selection of simulation software* is proposed by Bovone *et al* (1989). The purpose of using this method is to obtain the weights which can express the importance of software evaluation criteria with regard to the simulation objectives. The applicability of this method is illustrated using the following criteria: flexibility, learning and use, modelling speed, running speed, report features, debugging, stochastic capacity, ease of transport, service, and reliability. Separate evaluation tables are constructed both for the conceptual design (ie. initial phase in which ease of modelling is the most important) and for the detailed design (where the system under consideration is to be studied in detail), which emphasizes the importance of flexibility. On the basis of evaluation of several simulation packages using this method, the authors conclude that no product is superior to the others with regard to both software purposes (conceptual and detailed design).

The above studies illustrate some research carried out in the domain of simulation software selection. A general outcome of these studies is that there is no simulation package which can perform equally well for any application or purpose.

2.3.4. Survey Papers

The publications presented in this sub-section are survey studies carried out in order to investigate the issues related to the use of simulation and simulation software.

(i) Van Breedam *et al* (1990) have conducted a survey in order to evaluate several simulation software tools. They distributed a questionnaire to experienced users of simulation, who were asked to rate a sample of simulation packages on the proposed criteria. These criteria include flexibility, learning time, run-time observation, run-time alterations, statistics, data input/output facilities, on-line analysis, animation, customer support, literature, and price. On the basis of received answers, they classify the evaluated software into clusters according to their main features, because there is no package which scores highly on all the criteria. They propose the use of these clusters for the segmentation of the simulation software market. With such a division, each user can choose one cluster (and software within that cluster), according to the representative software features specified for a particular cluster.

(ii) Kirkpatrick and Bell (1989) have used a survey approach to *investigate the issues related to visual interactive simulation in industry*. These issues include the identification of the users of visual interactive simulation, the types of problems being addressed, reasons for using visual interactive modelling, and the ways in which this type of modelling affects problem solving. The results reveal that visual interactive facilities in simulation software enhance the interaction with the decision maker, provide more useful and easier-to-understand models, and improve decision making in general. Although some of the participants are aware of the significant set-up costs and the need for learning new software and a new methodology, most of them agree that visual interactive modelling is much more beneficial then the traditional approach to simulation.

(iii) Christy and Watson (1983) have used a survey of nonacademic users to explore issues such as the *functional areas that use simulation, the method of selecting simulation software, the popularity of various software tools for simulation applications, the problems associated with the application of simulation etc*. The response obtained reveals that, of the total applications of simulation, 59% are in the area of manufacturing systems. With regard to simulation software, the results show that, in general, there is a reluctance to implement and learn new programming languages for simulation applications. In addition, most companies attempt to utilize new software with current employees, instead of employing somebody with previous experience in any particular software. The authors consider these findings as support for the practice some vendors have, of providing attractive software arrangements to universities as a means of market penetration. Some of the main obstacles to the implementation of simulation include a lack of a quantitative education of managers, a lack of good data, a lack of time and a lack of end user education. Finally, suggestions for improving the use of simulation are presented. The main suggestions include easier-to-use and less expensive software, and improving the simulation knowledge of end users and managers.

(iv) In a report on the role of simulation in FMS, Carrie (1986) presents the main findings of the survey in which users of different software tools for manufacturing simulation were asked to *specify desirable features of these simulators.* The response indicated that a FMS simulator should be modular, user friendly, of low costs, enable easy and fast development of FMS models, give a variety of performance measures, and include financial analysis of the proposed investment and operational costs. On the other hand, users have found the following inadequacies in current simulators: program generation and compiling, speed of graphics programs, high costs, limitations of micro-based systems, and the length of time needed for model development. It is claimed that the effectiveness of FMS simulators depends on their flexibility. On the other hand, they are rarely capable of modelling special features such as model cutting tools or empty pallet problems. Conclusions outline that users should be able to evaluate and criticise proposals of software suppliers. Furthermore, users are looking for user-friendly simulators that can enable even non-specialists to be involved in model development. It is admitted that at present there is a gap between the expertise required by simulation packages and that available among users.

(v) An early survey carried out by Kleine (1970 and 1971), has *examined users' views of several discrete simulation languages.* A questionnaire distributed to the users comprised questions related to familiarity with the language, the language preferred for writing simulation programs, an evaluation of the difficulty in learning and using the features provided by the language, and capability evaluation. The analysis of results shows that the responses obtained are inconsistent for the majority of questions. It is difficult to interpret the results mainly because a limited number of respondents were proficient in more than one language. In addition, the expertise of some respondents is difficult to specify. A general conclusion drawn by the author is that one should try to conclude very little from opinion surveys.

The majority of the above survey studies deal with the investigation of issues regarding simulation software, either for the purpose of evaluation or examination of users experience and opinions about the software.

2.3.4. Simulation Software Descriptions and Tutorials

The review of research studies related to simulation software is completed with a brief presentation of publications dealing with descriptions or tutorials on simulation software. The majority of publications examined can be included in this sub-section. Because of this, and because of the relatively less significant importance of these studies to the scope of this research, these studies are only briefly mentioned and their general analysis provided.

A comprehensive simulation software survey produced by Swain (1991), provides a basic description of fifty six simulation software tools. Information presented is derived from the information supplied by software vendors. Many of these tools relate to data driven manufacturing simulators.

A survey of simulation software provided by Pollacia (1990) includes a discussion of manufacturing oriented simulation software products such as SIMFACTORY, ModelMaster, XCELL, WITNESS, FACTOR, and SIMAN.

An overview of simulation software is provided by Carrie (1988). In addition to a general overview, a detailed description of the manufacturing simulators WITNESS and MAST is given.

A concise overview of some manufacturing simulators is given by Paul (1991). In addition to the description of several general purpose simulation packages, a brief delineation of the manufacturing oriented packages EPSIM, PROPHET, SIMFACTORY II.5, and WITNESS is provided.

As a part of the discussion on visual interactive modelling and visual interactive software, Bell (1991) describes several packages for manufacturing simulation such as SIMAN/CINEMA, XCELL+ and Modelmaster.

A description of Modelmaster is given by Greenwood (1988), who declared that this simulator is designed for people who have little or no previous experience in computer simulation, to create and simulate models of complex manufacturing systems.

Within a discussion on simulation software, Pidd (1992a) provides a description of the manufacturing simulator XCELL+. In another publication (Pidd, 1992b), the same author describes a prototype simulation package SKIM developed in order to facilitate the design of quasi-continuous manufacturing plants.

A software vendor representative Clark (1991) describes the basic characteristics of the visual interactive manufacturing simulator WITNESS, emphasizing the superiority of the WITNESS interactive and animation capabilities in comparison to the features of some other simulation packages. Another report on a manufacturing simulator written by a vendor is a description of MAST given by Lenz (1983).

The main endeavour of a paper by Bollino (1988) is to highlight the use of the simulation package FACTOR as a powerful scheduling tool. A flexible manufacturing system simulator FLEXSIM is presented by Gelenbe and Guennouni (1991), whilst Thome (1988) provides a description of the manufacturing simulator GISA.

Valcada and Masteretta (1984) give a description of several FMS simulators: FIST (Flexible Integrated Simulation Tool), FMSSIM, HABMS, and K&T Simulator. O'Keefe and Haddock (1991) provide a description of two FMS simulators: the data-driven generic simulator RENSAM (Rensselaer Simulator for Automated Manufacturing) and RENVIS (Rensselaer Visual Interactive Simulator).

In addition to the above software descriptions, a number of publications related to software tutorials were found, written by software vendors. Some examples include the tutorials written by: Gilman and Billingham (1989) - SEE WHY and WITNESS; Murgiano (1990) - WITNESS; Farnsworth *et al* (1987) - AutoMod; Conway and Maxwell (1987) - XCELL+; Rohrbough (1989) and Goble (1990) - SIMFACTORY II.5; Harrell and Tumay (1990) - ProModelPC; Grant (1989) - FACTOR; Lenz (1989) - MAST; Suri *et al* (1990) - ManuPlan and SimStarter; Sturrock and Pedgen (1989) - SIMAN; and Poorte and Davis (1989) - CINEMA.

The main characteristics of many publications related to software descriptions, and of all software tutorials, is a lack of criticism. With a few exceptions, these studies mainly provide a delineation of software with discussion on its capabilities. Software limitations and problems that are likely to be experienced during modelling are not addressed. This is especially the case for publications written by software vendors or developers, which inevitably show a certain level of bias.

2.4. A CRITIQUE OF THE LITERATURE

When the background research material was selected for analysis, publications related to commercially available packages were considered, although numerous examples were found where individual institutions developed their own software for specific needs.

Most publications on the use of simulation in manufacturing environments reflect the views of individuals whose expertise in one software product is difficult to relate to publications that present expertise in another simulation language or simulator. Furthermore, there are many reports on software packages written by its vendors, who are not critical at all. They all try to advertise the software product. I also obtained a similar notion during some presentations at conferences. Similarly for the reports written by the members of research teams involved in software development, in various commercial or research institutions.

Independent users that were involved in a number of case studies, produced more critical and unbiased reports on software characteristics. A general impression gained is that the more practical experience users have gained with different packages for modelling real systems, the more pertinent criticisms they have.

In general, users of software tools usually describe the better part of the simulation study - success. Mistakes, learning efforts, frustrations, difficulties they had using the software, software limitations, logical features that they were not able to model adequately (therefore they had to make some assumptions in order to simplify the models) are rarely described.

It seems likely that in these cases where some authors provide a review of a number of different simulation packages, they do not have an opportunity to test all these packages and use them for developing real complex models. Instead they had to rely on information about the software products supplied by vendors.

When publications relevant for this research were analyzed, it was discovered that there are more papers describing case studies, than software tools used in these studies. The majority of the papers that describe one particular tool are either written by their vendors, or by a member of the research team that developed a particular simulator. A recapitulation of the findings drawn from the background research material analysis, that supports the objectives of research of this thesis, is as follows:

(i) In software evaluation, many factors are to be assessed, and their significance weighted in order to evaluate, compare and select adequate software.

(ii) Although several studies provide criteria for the evaluation of simulation packages in the manufacturing domain, these criteria are not as comprehensive as are those provided in this thesis.

(iii) An analysis of the studies that provide evaluation criteria or desirable features of simulation software reveals that these features mainly relate to graphics and animation, interactivity, modelling flexibility, ease of use, ease of learning, modelling assistance, portability, execution speed, price, data analysis capability, and supplier support.

(iv) Several evaluation studies are based on information provided by vendors, and are lacking criticism.

(v) The significance of software evaluation has been identified in several studies. For example, O'Keefe and Haddock (1991) declare that "there have been few rational efforts to evaluate the development and use of data driven simulators in manufacturing modelling. As the number of tools continue to grow, such evaluation is necessary if simulation users are going to make sensible informed choice".

(vi) Although some of the evaluation studies consider WITNESS, SIMFACTORY II.5, XCELL+ and ProModelPC, none of these evaluations and comparisons are comprehensive.

(vii) None of the research studies have found that a particular simulation package is superior to other packages for any purpose of simulation and for any application.

(viii) Many studies have recognized the importance of software selection. For example, one of the studies (Haider and Banks, 1986) claims that "selection of an appropriate simulation software product can make a significant difference in how well simulation analyses support managerial decision making".

(ix) None of the studies focus on the methodology of simulation software selection.

(x) Although the majority of the survey studies investigate issues related to simulation software, none of them examine users' opinions about possible ways to improve software.

(xi) As the existing simulation packages are continuously being revised, and new ones are being released on the market, the significance of some of the earlier studies is diminished.

(xii) A final conclusion to be drawn from the literature review, which justifies the objectives of this research, is the following. Due to the intensive use of simulation in manufacturing environment, and the costs involved in purchasing simulation software, hardware and training, further research in software evaluation, comparison, and selection seems to be needed and applicable.

2.5. SUMMARY

This chapter presents the background research material. Publications discussed have been studied in order to gain a better understanding of the subjects that became the research objectives. In addition, the intention was to attain an insight into contemporary reports on research in the field of simulation of advanced manufacturing systems, and on research in simulation software evaluation, comparison and selection.

The review of survey studies illustrates the popularity of simulation for modelling automated manufacturing systems. Research studies related to simulation software provide a basis for a critique of the background research material and a basis for setting up the objectives of this research.

The important point that comes out of the literature critique is that the majority of papers discuss what a certain package can do, but they rarely indicate what it cannot do. The exception are these few publications which presented a critical evaluation (although never in too much detail) of several packages, in terms of various criteria.

Nevertheless, in several publications a need for evaluation and establishing criteria for the evaluation of simulation software is evident: "Companies embarking on the use of simulation for the first time are seeking guidance on what they should look for in a simulation language, on the criteria to use in selecting a language, on the features needed in a language for modelling manufacturing systems and on the ease of learning and using a language, amongst other factors. The choice of an inappropriate language can be expensive both in terms of the software purchased and in the training of personnel." (Ekere and Hannam, 1989).

Publications analyzed reflect a particular cultural view, limited by availability and the use of different manufacturing simulators. In an attempt to overcome this issue, a case study approach was used in this research. The main purpose of this was to critically evaluate some of the most popular manufacturing simulators, by investing effort in modelling a real system of substantial complexity. The endeavour was to produce an unbiased report, with a description of all the problems experienced by using a particular package and/or experienced by trying to model a particular logical feature of the system.

The maximum number of evaluation criteria listed in the publications analyzed is moderate. Therefore, this research represents an attempt, among the other objectives, to establish more comprehensive guidelines for the assessment of software packages for manufacturing simulation, and to propose a structured approach to simulation software selection. Finally, this research should determine which characteristics a proper simulator has to possess in order to cope with the complexity that arises in automated manufacturing systems, and how the existing simulators can be further improved to make simulation more beneficial and accepted.

CHAPTER 3. A CASE STUDY: SIMULATION MODELLING OF THE POWDER COATING SYSTEM AT BICC-VERO ELECTRONICS

3.1. INTRODUCTION

This chapter presents a case study carried out in a real manufacturing environment. The study relates to simulation modelling of an automated system for the electrostatic powder coating of metal components installed at BICC-VERO Electronics, Eastleigh.

A description of the system being studied is provided in section 3.2, with an emphasis on its configuration and the logic of the production process. The objectives of the modelling are elaborated in section 3.3, whilst section 3.4 presents a description of data collection and analysis. A conceptual model developed using activity cycle diagrams is presented in section 3.5 together with a discussion on the suitability of this graphical method for modelling real complex manufacturing systems.

The computer models developed using several manufacturing simulators are presented in section 3.6 together with a brief description of each simulator. The WITNESS simulator was used for modelling the system under consideration with a substantial level of detail, and the model thus obtained through an iterative process was experimented with. Other simulators such as SIMFACTORY II.5, XCELL+ and ProModelPC were also used for modelling the same system, although mainly for the purpose of software evaluation. These models do not, therefore, contain such a level of detail as the WITNESS model, but they are presented so that differences among the models because of the features of the packages can be discussed.

The process of establishing confidence in the model is described in section 3.7. A variety of production strategies were tested in experimentation. Following the description of experimentation, the results obtained are presented and analyzed in section 3.8. Section 3.9 provides a summary of information relating to the case study, which includes a review of the simulation process implemented and an analysis of the main findings obtained in this research.

3.2. AN AUTOMATED SYSTEM FOR ELECTROSTATIC POWDER COATING

This section presents basic features of the system being modelled. Some general information about the powder coating system are presented in sub-section 3.2.1, while a more detailed description of this system is presented in sub-section 3.2.2.

3.2.1. General Features

The automated system for the electrostatic powder coating of metal components is a part of factory that produces racks, card frames and enclosures for electronics products. These components are produced in flexible manufacturing cells installed in the same factory, and after coating are assembled with other components to create final products. Whilst a description of this system is provided in this section, some more information about the entire factory is presented in Chapter 1.

Due to the specific characteristics and functions of the system for electrostatic powder coating, it can be considered as a separate unit and such an approach has been taken in this study. The system paints various metal components using the method of electrostatic powder coating. Powder coating is applied through a special gun which electrostatically charges the particles and deposits them on to an earthed article. The charge allows the particles to remain adherent to the substrate and when baked the particles fuse together and crosslink the polymers to form hard abrasion resistant coatings (Bassett, 1989).

Metal components (made from steel, aluminium or zinc) are coated in powder booths. The powder booth is designed as an enclosure in which component parts constantly moving at a fixed speed in front of powder guns are electrostatically powder coated.

Figure 3.1 shows a standard powder booth. Extended inlet and exit vestibules allow components to enter and leave the powder booth. Gun slots are provided on each side of the booth to allow powder guns to move to cover the height of the components. The number of gun slots (and guns) depends upon the complexity of the components and the speed at which they move in front of the guns. A manual spraying aperture with a lift out door is provided to allow for a manual finish of the components if needed.

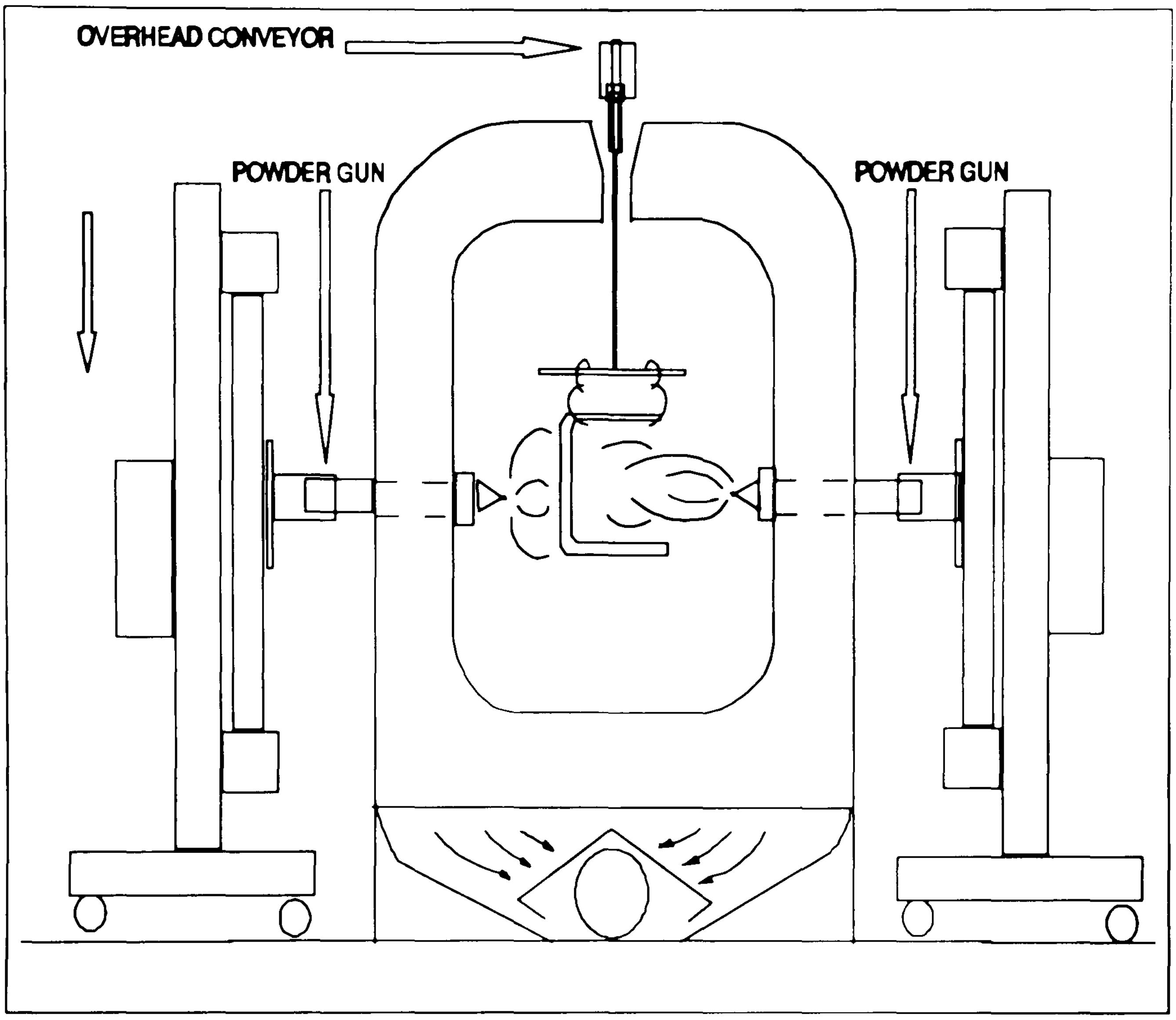


Figure 3.1 Powder spraying booth

3.2.2. Description of the system

The entire system consists of a large overhead conveyor chain passing through several processing areas. Components to be coated are attached to flight bars, which are mounted on the conveyor. The number of parts per flight bar depends on the product type of which there is a range of approximately two thousand different part types. For example, one large part may need two flight bars, whereas on the other hand five hundred small parts (screws) can be jigged together and hung on one flight bar.

Batches of parts to be painted are stored in the storage area near the powder coating system. Each batch has several special characteristics such as batch number, which determines the part type, batch size, colour, number of parts per flight bar, masking requirements, priority, and manual finish if complex parts are to be sprayed automatically.

All parts within one batch have the same features, and are treated equally. When parts are jigged together prior to loading on the conveyor or loaded directly on flight bars, all parts from one batch will be taken before parts from the subsequent batch are loaded in order to avoid mixing of batches. In case that the number of parts per flight bar is large (around half a dozen or more), parts are jigged together off-line, pre-hung in a special area, and then loaded on flight bars when it is convenient.

There are some types of components which do not have to be coated completely. In that case all components in the batch have to be masked, which means parts that should remain uncoated are covered by paper. This operation is performed by all available workers and can be done on-line and off-line. When masking is on-line, parts are already loaded on the conveyor, and special stickers are placed on the components while the conveyor is moving. If masking requirements are more complex, then masking is performed off-line when convenient, whilst another batch is being loaded on the conveyor.

After the loading of one batch is completed, a certain number of flight bars should remain empty. If the subsequent batch should be coated in the same colour, then two flight bars are left empty. This gap is needed for readjusting the guns due to the different shape of the parts in the next batch. On the other hand, if the next batch is to be painted in a different colour, then ten flight bars should stay empty between these two batches. This has to be done in order to avoid the touching of parts from different batches (coated in different colours) in particular parts of the oven.

After the parts have been loaded onto the flight bars in the loading area they are transported through several processing areas, prior to their unloading after the last processing stage. The first stage of processing is pretreatment, where parts are prepared for coating. This process consists of spraying the components with a special chemical solution to clean and condition metallic surfaces uniformly. The components are then given a spray water rinse and treated again with another solution which enhances the corrosion resistance.

Following pretreatment, the parts are transported to the oven where they pass through a drying area in order to remove moisture. After drying, the parts are transported to the powder coating area. There are two automatic and two manual booths for coating. If the batch size is small (by the convention accepted in the plant, a batch is small if it has less than 20 parts) and the batch has normal priority, or if the batch size is larger (up to 80 parts) and the priority of the batch is high, then such a batch is coated manually on the first booth, with one gun on each side. Manual spraying is performed in these cases, because the time needed for cleaning this booth before each colour change is smaller (30-40 minutes, whilst automatic booths need 40-50 minutes for setup).

If the part has a normal priority and the batch has more than 20 parts, then parts are sprayed on one of the two automatic booths. The first automatic booth is dedicated to a special colour (light grey), whilst all other colours are sprayed on the second automatic booth.

There are six spraying guns on every automatic booth, three on each side. The number of guns used for spraying depends on the size and complexity of parts within a particular batch. When parts within a particular batch are complex, then a worker has to perform re-coating ie. manual finish by a spraying gun. If there are no more than two parts within the batch, then this sample batch is sprayed manually on the last booth.

Following coating, parts are then transported to the oven, where they are baked in order to preserve the coating. After this last stage of processing, parts go to the unloading area, where they are unloaded from the flight bars, separated and moved out of this system.

The operation of the system is facilitated by eight workers, divided in four groups. First, there is one worker dedicated to painting, performing both coating on manual booths or re-coating (manual finish) on automatic booths. Second, there are three workers dedicated to part jigging and loading the flight bars. Third, there are two workers assigned to unloading the parts from flight bars after the last operation (baking). Finally, there are two 'floating' workers, who can perform any kind of job, depending on the current priorities in the system.

Each worker in addition to his dedicated task has also to participate in the following activities: daily, weekly and monthly maintenance, masking, cleaning the booths and guns (setup) after each colour change, or repairing equipment in the case of equipment breakdown.

Breakdowns of the conveyor chain occur in average every three months, and it takes about 35 minutes to repair it. Once a month on average, there is a breakage of one or more spraying guns, and it takes about 15 minutes to repair one gun. The summary of the main model components is as follows:

(i) overhead chain conveyor
(ii) pretreatment area
(iii) two booths for automatic spraying
(iv) two booths for manual spraying
(v) oven with separate drying and baking areas
(vi) eight workers

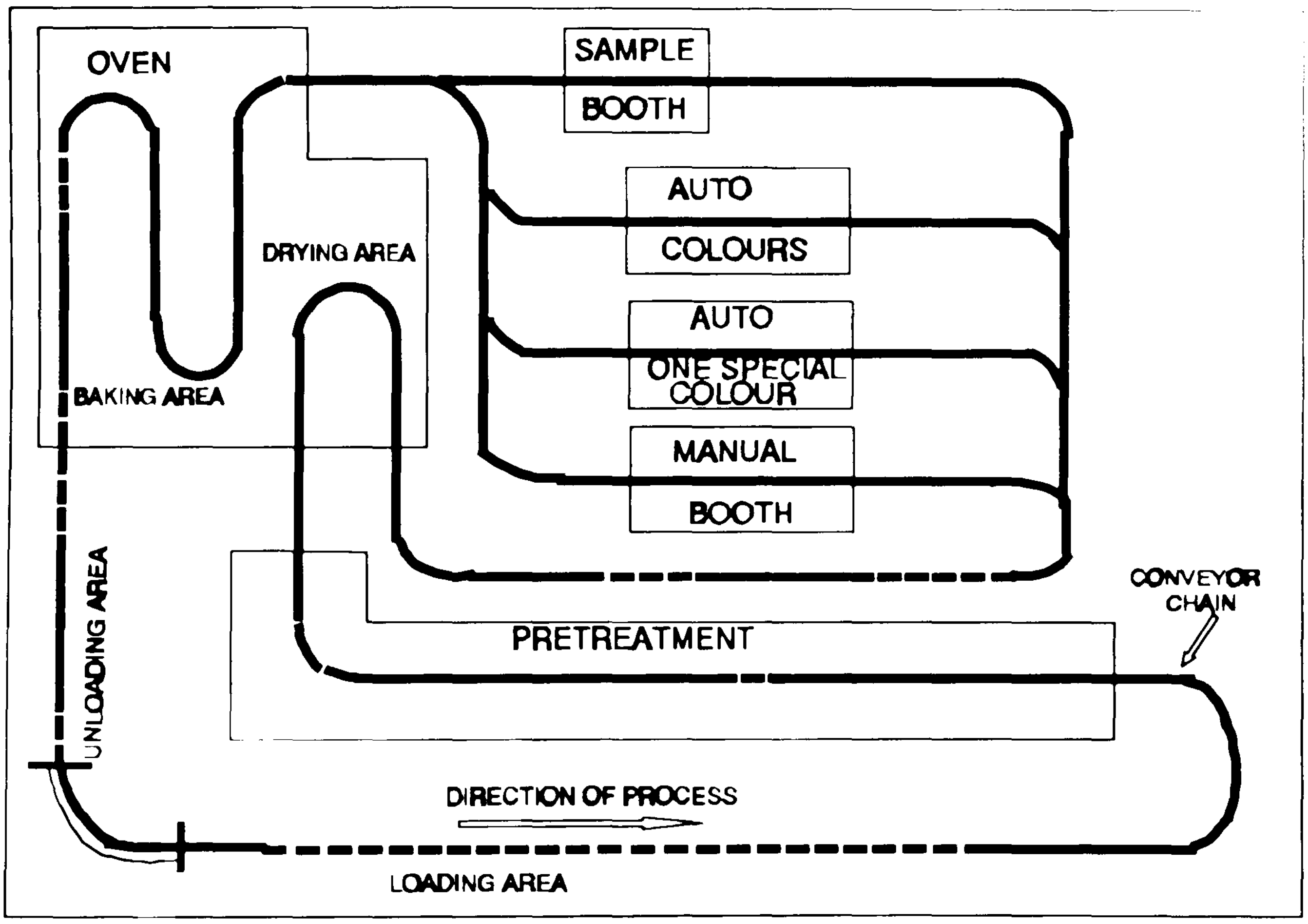


Figure 3.2 The layout of the powder coating system

The list of the components does not include control system for conveyors and spraying guns, powder recycle unit, extraction filters, detailed equipment for pretreatment or oven areas, sources of energy, ventilation etc. These components were not considered, because they are not relevant for the flow of components and measures of system performance analyzed in this study.

The layout of the system is shown in Figure 3.2, whilst Section 1 of the thesis supporting material shows photographs of the main parts of the system.

3.3. OBJECTIVES OF THE MODELLING

The objectives of this simulation study differ from the objectives of the thesis, described in Chapter 1. Whilst the thesis objectives generally relate to simulation software evaluation, selection and improvement, several specific objectives were to be satisfied in carrying out a case study. These case study objectives are classified in two groups. The first group comprises my personal objectives. These objectives are presented in sub-section 3.3.1. The second group contains objectives defined by the company in which the case study was carried out. Sub-section 3.3.2 presents the objectives included in this second group.

3.3.1. Personal Objectives

The objectives included in this group express what I tried to achieve by working on this project and why a case study approach was taken. The following objectives were established:

(i) Analysis and modelling a real life complex manufacturing system.
(ii) Acquiring some practical experience in addition to the theoretical research.
(iii) Improving knowledge about simulation.
(iv) Application of simulation methodology in a real manufacturing environment.
(v) Use of different software tools in a real manufacturing environment.
(vi) Producing the results of research that would be applicable in practice.

3.3.2. Company's Objectives

The objectives included in this group are defined from the point of view of the company in which the research was carried out. These objectives specified in agreement with production managers, include:

(i) The main objective of the modelling was to understand how the system works and to discover how the performance of the system can be improved, especially regarding the throughput. This system is a bottleneck in the factory. There is always a substantial number of parts to be painted stored in the storage area. These parts represent raw materials for the powder coating system, but they also represent work in process for the entire factory (which should be reduced as much as possible). At the moment, the other parts of the factory are more productive than the powder coating system, so the efficiency of this system has to be improved.

(ii) Another objective is to assess whether the current manpower level is adequate, or some additional workers would have to be employed.

(iii) Further objective is to analyze the utilization of painting booths.

(iv) Since there is a possibility of transferring components to be coated from another factory to the system being studied, the simulation model to be produced should enable testing different configuration alternatives. The results of such experiments should help managers to decide what changes they have to make in the system in order to cope with an increased number of components to be coated.

3.4. DATA COLLECTION AND ANALYSIS

Although there is a formal method of data collection in the factory, information already available for the powder coating system at the beginning of the project was not sufficient for the purpose of modelling. An example of such a report is a list of various product types (distinguished by an unique order number) painted in the specific colours in a period of six months together with an estimated number of parts per flight bar (for most of the products this number was not available), and total number of flight bars required for particular product. This information is regularly collected on the shop floor, when operators transfer the information about the products (batches) to the computer system via bar code.

Information in the six months report that was used in the model relates to the proportion of parts painted in one of 19 colours, because the six months sample was more representative than it would be a sample with new data collected for a short period.

Section 2 of the thesis supporting material shows what colours were used for coating, and the percentage of parts painted in certain colours in the mentioned period.

It has already been mentioned that available data were not sufficient for the modelling purpose. For example, information about the number of parts per flight bar was not complete, there was no information about the masking requirements, manual finish, repainting etc.

In order to obtain this information, special forms were created and then filled in by workers on the shop floor for a period of four weeks. These forms comprised information about the part types (order code), batch quantity, number of parts per flight bar, masking requirements, prehanging, type of spraying (only automatically, automatically with manual finish, and only manually), number of empty flight bars between batches, the colour and special comments such as an indication of whether the batch has to be repainted. Section 3 of the thesis supporting material shows a copy of the form, completed for one working day.

The data collected on the shop floor was statistically analyzed and used in the model. The values recorded for the batch size and number of parts per flight bar were fitted to the theoretical distributions using the statistical package STATGRAPHICS (STSC, 1986). Although it was possible to use empirical user-defined distributions that would comprise only collected data, distribution fitting was chosen in order to provide sample values outside the range of the collected data. By applying the heuristic method of graphically comparing fitted distributions with the empirical values together with the Chi-Square Goodness-of-Fit Test, the most appropriate theoretical distributions were found for variables representing batch size and the number of parts per flight bar. Section 4 of the thesis supporting material provides more information about distribution fitting.

The two fitted distributions for batch sizes and the number of parts per flight bar were applied to 98.27% of the normal type parts (type 1). For the special type of parts (type 2), where on average 500 small screws were jigged together and loaded on the flight bars, empirical distributions were used.

For other data such as masking requirements, batch priority, manual finish, and repainting which either applied to a particular batch or not, the percentage of parts with specific characteristic was calculated. This data was analyzed using the QUATTRO PRO (Borland, 1991) spreadsheet package. An analysis has produced a percentage of parts with a specific feature, which was needed for modelling. Section 5 of the thesis supporting material presents an example of this table, whilst Section 6 shows the final results obtained and then used in computer models.

Information about the system was also obtained by interviewing managers, production engineers, and the shop floor foreman. They provided information about the operating procedures, system maintenance, breakdowns of equipment, shift patterns, and many other useful information related to the system in general. For example, information obtained for the shift pattern revealed that the system had operated in only one shift, from 7.30 to 16.30. During that time plant stopped for 12 minutes only during the tea break. In this shift pattern, workers are divided in two groups in order to have a lunch break at a different time, so that the coating does not need to be stopped for lunch.

Some previous reports on the analysis of the powder coating system were also useful sources of information. A group of production engineers investigated production in the powder coating system for the period of one month (July 1991), and produced a report showing an analysis of the throughput and costs of production. In addition, several other publications and internal reports were analyzed (Termoset Powder Coatings, 1989), (GEMA, 1985), (PYRENE, 1990), (BICC-VERO, 1985), (BICC-VERO, 1991).

Finally, the duration of some operations (loading, unloading, and masking) were measured several times during the actual production process and the mean values derived used in the model. In addition, information about physical characteristics (the number of flight bars in specific processing areas and parts of the conveyor chain) was counted on the shop floor.

To summarize, the following methods of data collection were used in the research:

(i) Data collection on the shop floor, where workers filled up the forms.

(ii) Interviews with the managers, production engineers and foreman.

(iii) Analysis of the previous studies on the system and the literature related to powder coating.

(iv) Measuring the durations of operations on the shop floor.

(v) Observation of the system on the shop floor.

3.5. CONCEPTUAL MODEL

There are many methods available for conceptual modelling, especially in the European literature. Activity cycle diagrams, which feature prominently (Tocher, 1961), (Carrie, 1988), (Hlupić and Paul, 1992b), (Čerić and Hlupić, 1993), (Holder and Gittins, 1989), (Szymankiewicz *et al*, 1988), were used as a formal method for modelling the powder coating system. A formal method concerns the use of standard notation and a set of rules by the analyst to depict the problem logic in a systematic way (Au, 1990).

This section presents the activity cycle diagram developed for the powder coating system (sub-section 3.5.1.) together with a discussion on the applicability of this method for modelling complex manufacturing problems (sub-section 3.5.2), whilst a methodological overview of activity cycle diagrams is presented in Appendix A.

3.5.1. Conceptual Model of the Powder Coating System

The development of conceptual simulation models is a stage of the simulation process that precedes the development of computer models. Conceptual models refer to a representation of a simulation models' logic and structure.

A conceptual model of the powder coating system was developed for several reasons. As first of all, to provide an understanding of the system, prior to computer model development. Not only did the model developed using the activity cycle diagrams show the basic features of the system related to its structure and logic, but also it provided a good medium for discussion with the managers and production engineers. Such a model provided the skeleton with the basic features to be included into the computer model, which facilitated computer model development.

Due to the complexity of the system being studied the activity cycle diagram of the powder coating system was developed hierarchically. Figure 3.3 presents the main activity cycle diagram of the powder coating system. Two activities from this diagram shown with double boxes, that represent loading and unloading, and painting (coating) are further modelled in more detail as shown in Figures 3.4 and 3.5.

Figure 3.3 shows the main activity cycle diagram of the powder coating system. This diagram consists of the life cycles of classes of entities representing parts to be

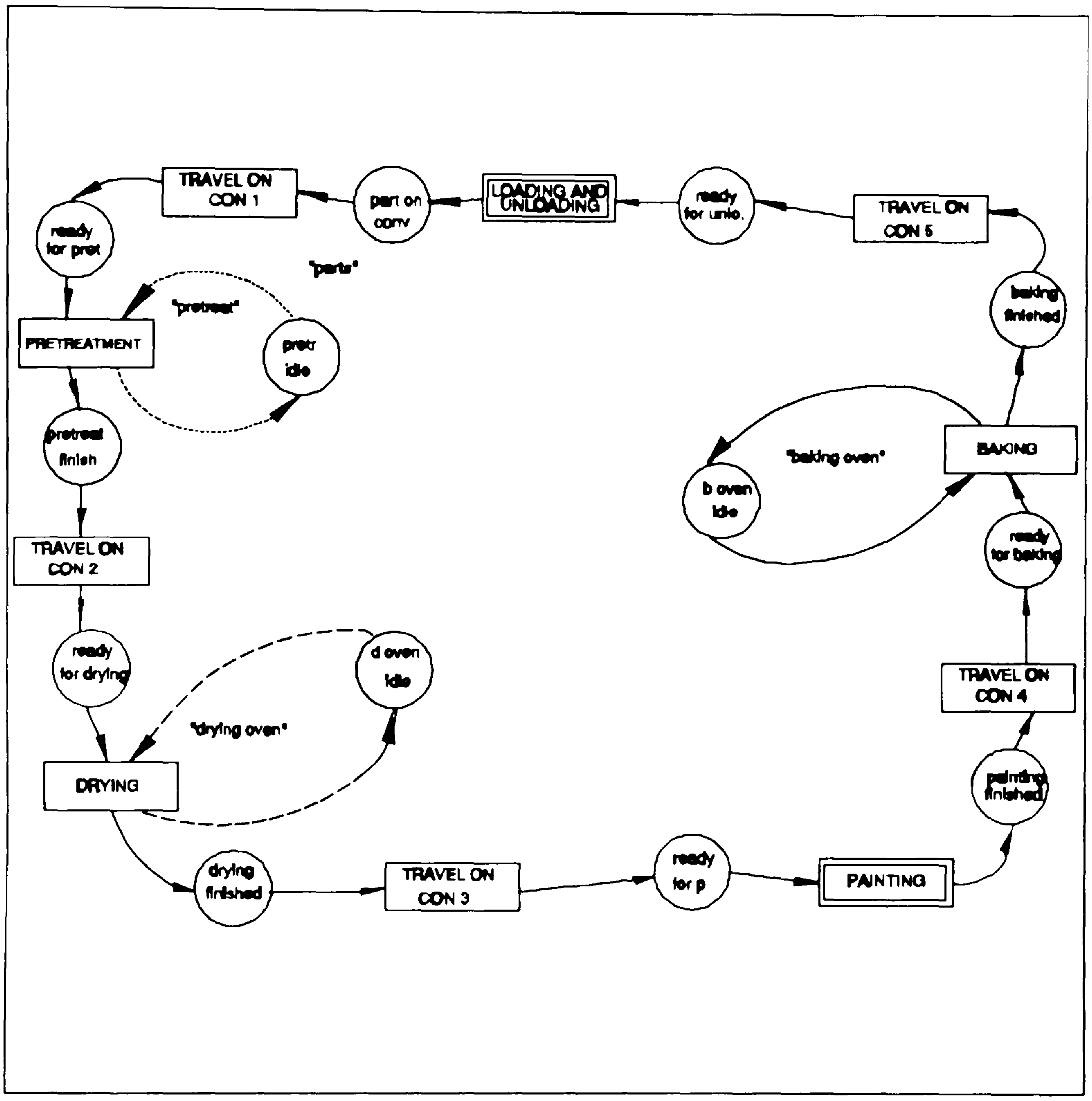


Figure 3.3 Activity cycle diagram for the powder coating system

painted, the pretreatment area, and separate areas of the oven used for drying and baking.

Activity "LOADING AND UNLOADING" represents the first and the last phase of the part handling in the system. After the parts are loaded on the conveyor flight bars, they are transported (activity "TRAVEL ON CON 1") to the pretreatment area for preparation for coating. Here, parts are actually represented by flight bars. Each flight bar holds a specific number of parts jigged together, which is recorded by an attribute. Accordingly, the number of entities in the pretreatment area, and in the oven areas for drying and baking, correspond to the number of flight bars accommodated in these areas. These areas were explicitly modelled in order to distinguish and emphasise processes performed before and after coating.

Following pretreatment, parts are transported ("TRAVEL ON CON 2") to the area of the oven where parts are dried before coating. After drying, parts are moved further ("TRAVEL ON CON 3") to the painting area (activity "PAINTING") which is further modelled as shown in Figure 3.5. Subsequently to painting, parts are further transported ("TRAVEL ON CON 4") to the oven for baking, and then transported again ("TRAVEL ON CON 5") to the unloading area (activity "LOADING AND UNLOADING").

Figure 3.4 shows loading and unloading of the parts modelled in more detail. All activities and queues between queues "part on conv" and "ready for unlo", represent the logic behind activity "LOADING AND UNLOADING" that is included in the main activity cycle diagram shown in Figure 3.3. This is the reason why both the diagrams shown on Figures 3.4 and 3.5 are not constructed as closed loops.

The activity cycle diagram developed for loading and unloading shows a part of the life cycle of classes of entities representing parts and labour. Another part of the labour cycle is modelled in the activity cycle diagram for painting, whilst the life cycle of parts is shown on both of the other diagrams (Figures 3.3 and 3.5). The queue "labour idle" which belongs both to diagrams representing loading and unloading and painting is represented by a double circle. A section of the parts life cycle shown here relates to the following activities. Firstly, parts arrive in the system, which is represented by activity "PARTS ARRIVE" in which the logical entity "arrival" participates together with parts ensuring that only one batch arrives at a time. On arrival of the parts, several attributes have to be defined such as batch size, batch number, number of parts per flight bar, priority of the batch, manual finish if the parts are to be coated automatically and masking requirements.

If parts within a certain batch have to be masked, they go firstly to the "MASKING" activity, after which they are placed to the storage area again (queue "wait in storage"). It is assumed that there is always enough space in the storage area. If there are several parts (three and more) to be placed on the flight bar, they are jigged together ("JIGGING"), and then hung on the flight bar ("HANGING"), or hung directly if there are less than three parts on the flight bar.

In this model a conveyor chain is represented by flight bars which are loaded by components. In the life cycle of class of entities representing flight bars, entities are positioned in the queue "fb at entry". If a flight bar is going to be loaded with parts, it

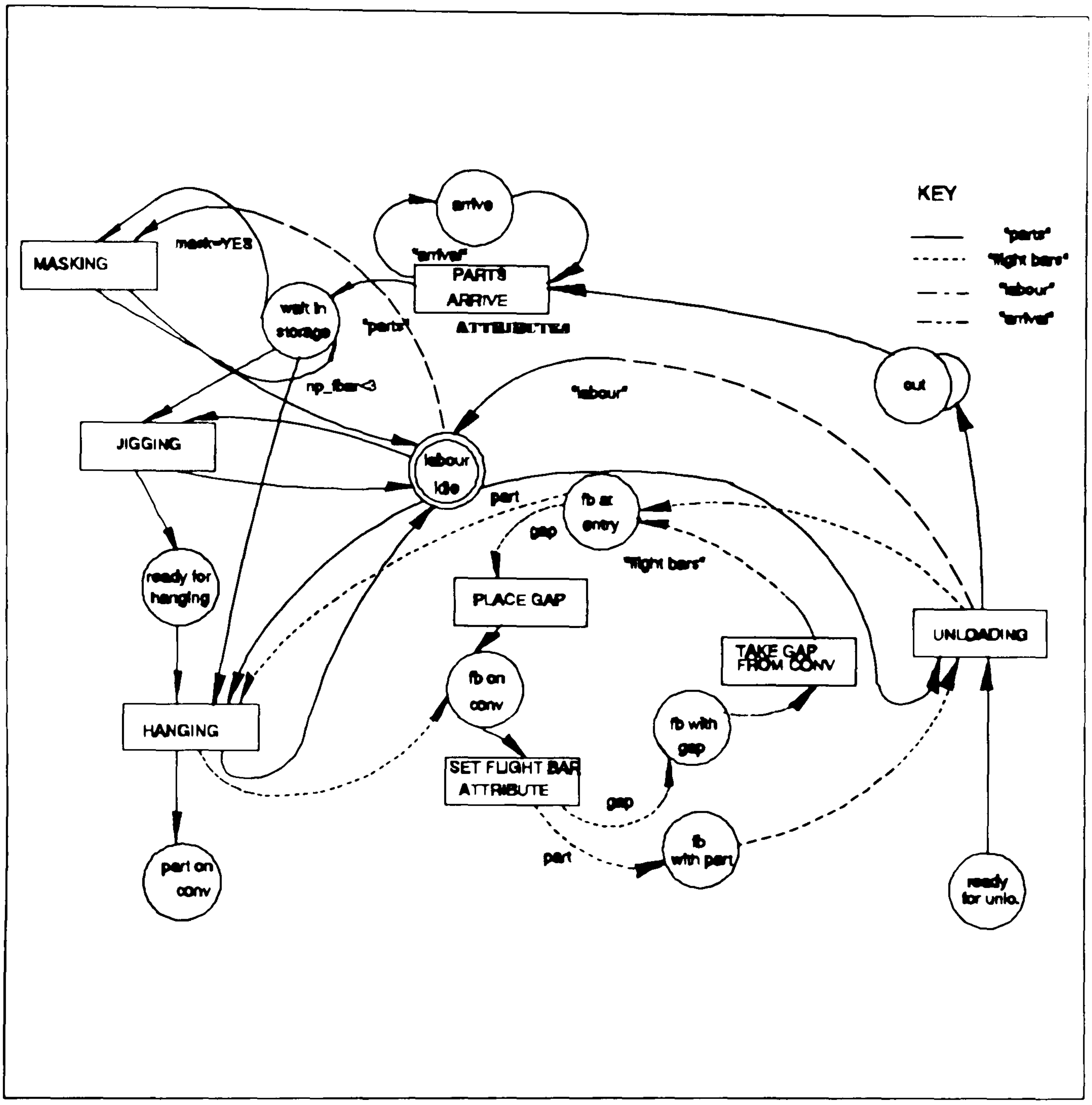


Figure 3.4 Activity cycle diagram for loading and unloading

participates in the activity "HANGING", if not, it moves to the activity "PLACE GAP" which means that it stays empty. In both cases, flight bars are subsequently placed in the "fb on conv" queue with a FIFO queuing policy, so that all flight bars are kept in strict order. In the real system flight bars move throughout various processing areas together with parts after loading and before unloading and behave like a single unit. Therefore flight bars and parts were not modelled separately after loading. After loading the number of parts that move through the system represent the number of loaded flight bars, carrying the attribute which records how many parts are jigged together on each flight bar.

On the other hand, entities representing flight bars are positioned in the queue "fb with part", waiting to be released for new loading. This happens after activity

"UNLOADING" if the flight bar contains parts and after activity "TAKE GAP FROM CONV" if the flight bar has a gap on it.

Comparing the activity cycle diagram shown in Figure 3.4 with its description, it is apparent that several details related to the logic of the system are not included in the diagram. For example, for the sake of model readability, attributes to be defined on parts arrival are not listed separately. The number of parts in the storage queue is not recorded, together with the number of parts participating in masking (a complete batch should go to the "MASKING" activity, and several "labour" entities can participate in this activity, depending on their availability and on the number of parts to be masked).

The number of parts to be jigged together is not shown (it depends on the part attribute), together with a counter, which should determine when a new batch has to be loaded. This counter should also indicate when gaps ought to be attached to the flight bars. The number of gaps depends on the attributes of the subsequent batch, which is also not shown on the diagram.

Each entity within the class of entities representing labour has an attribute which distinguishes different types of labour. According to these types, only a specific type of labour can participate in specific activities. This has again also not been shown on the diagram.

There are other details not shown on the activity cycle diagram. For example, when activity "UNLOADING" starts (parts are unloaded from the flight bar), it should be monitored how many parts have actually been unloaded. This is not displayed on the activity cycle diagram. The details related to breakdowns of the conveyor chain, spraying guns, pretreatment or oven areas as well as details related to shift patterns have also been omitted from the model.

Figure 3.5 shows part of the model related to powder coating, modelled in more detail. Here, all activities and queues between queues "ready for p" and "painting finished" represent the logic behind activity "PAINTING" which has been included in the main activity cycle diagram shown on Figure 3.3.

The activity cycle diagram developed for painting shows a part of the life cycle of the entities representing parts, labour and painting booths. This diagram includes another part of the labour life cycle (the life cycle of parts is shown on all diagrams). Therefore queue "labour idle" belongs to both diagrams related to loading and unloading,

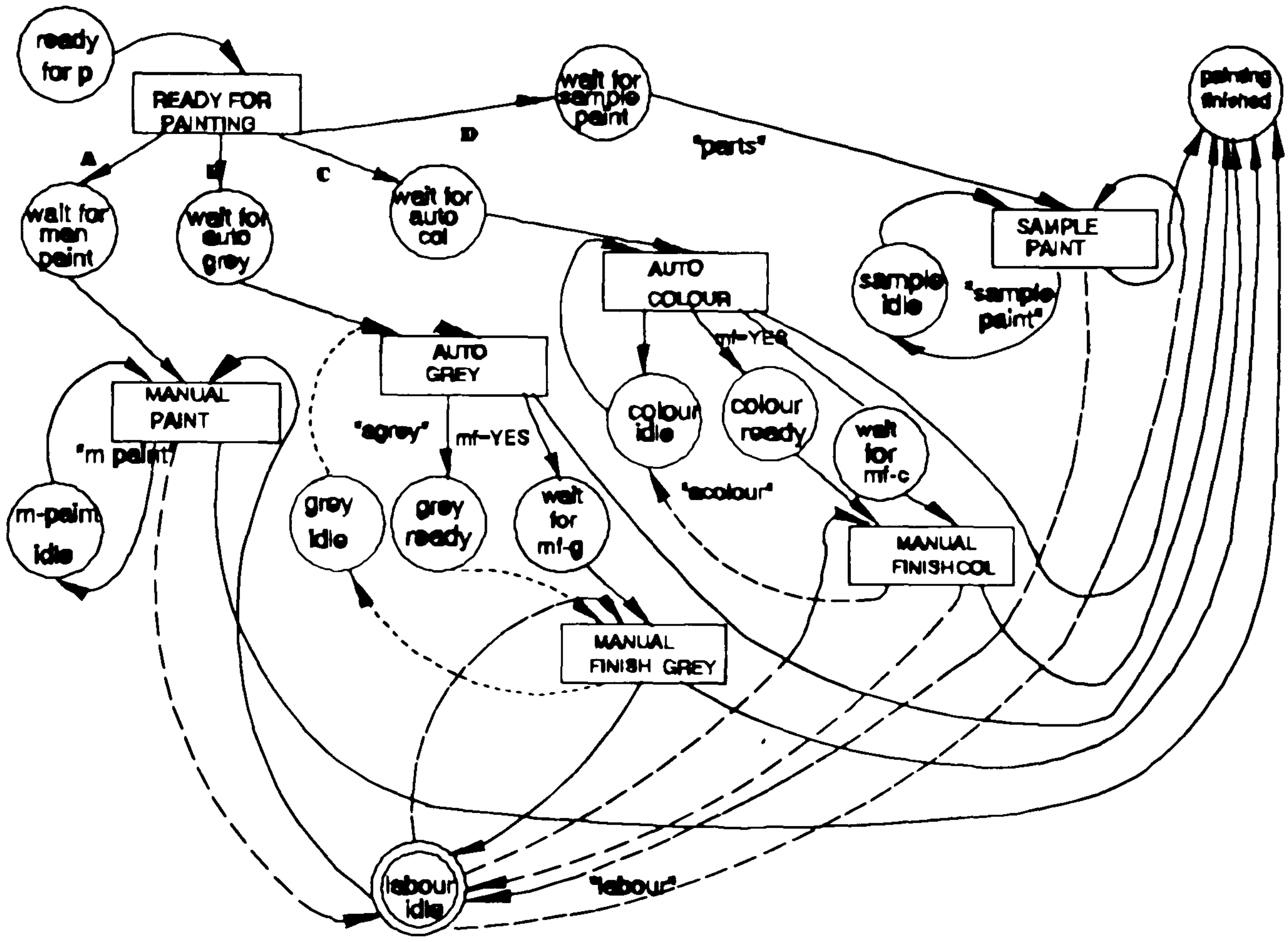


Figure 3.5 Activity cycle diagram for painting

and painting. One common queue for entity "labour" has been defined, because there are workers that can participate in any activity, as well as workers dedicated to specific tasks. This part of the logic can be handled by attributes (although not shown on the diagram). Such an approach has simplified the structure of the model, but it could not show much of the logic behind it.

When parts are transported to the coating area (queue "ready for p"), they are firstly placed in a dummy activity "READY FOR PAINTING" which enables branching to four different directions, leading to one of the four painting booths. These branches are labelled with conditions A, B, C or D, because again it was not possible to specify all conditions on the diagram in a readable and organized manner. Condition A is satisfied when the batch size is greater than 2 and smaller than 20 parts and the batch has a normal priority, or the batch size is between 20 and 80 parts and the batch has high priority. When the batch size is larger than 80 and parts have to be painted to a specific colour, parts are routed via a branch labelled by condition B. Condition C is satisfied when the batch size is larger than 80 and parts have to be painted in any other colour. Finally, when the batch size is smaller than or equal to 2, then parts are painted on the fourth booth which handles these samples (condition D).

Painting on the booths was modelled in as little detail as possible in order to produce an understandable diagram. When coating is performed on automatic booths, there is the possibility of a manual finish in these cases that require it. If not, after coating parts go directly to "painting finished" queue. Painting booths could be modelled in much more detail. For example, details that relate to the set up of the booths and cleaning of the spraying guns whenever the next batch is painted into a different colour on a specific booth could be included. These would result in much more complicated activity cycle diagram, not understandable and readable. It was assumed that it is better to construct a diagram that comprises only the main logic of the system, instead of including many details that can only contribute to confusion and make activity cycle diagrams not usable.

3.5.3. Applicability of the ACD for Modelling Complex Manufacturing Systems

Manufacturing simulators used for the development of computer models do not require activity cycle diagram as would some packages based on the ACD, such as HOCUS (Szymankiewciz *et al*,1988) or VS7 (SysPack, 1990). The simulators used in this research are based on the network concept, which means that the models developed by these simulators are regarded as a network of production equipment (eg. machines, buffers, transporters and conveyors) through which parts are moved requiring processing. Such an approach does not 'naturally' support the concept of activity cycle diagrams. Because of this, it is not possible to link directly the logic of the model expressed by activity cycle diagrams to a computer model developed by a data-driven simulator. Despite this fact it is believed that the activity cycle diagram of the powder coating system as a software independent modelling tool provided several benefits.

Firstly, it gave an insight into the problem and understanding of the basic structure and the logic of the system being modelled. As such, this conceptual model provided a structured way to determine the basic model elements and their interaction. Secondly, this model facilitated communication with clients at early stages of the simulation process. Production engineers and managers participated in the development of the conceptual model discussing its structure and possible modifications. Thirdly, although it was not directly linked to computer models, activity cycle diagram provided help in structuring computer models.

Regardless of the advantages when using activity cycle diagrams for development of conceptual models, it is obvious that this method becomes too limited as the complexity of the systems arises. In order to model a complex, real automated manufacturing system, a number of approximations have to be made.

The automated system for powder coating was modelled hierarchically. For two parts of the main diagram, two separate diagrams were constructed. This resulted in the splitting of some life cycles and modelling the same queues in different diagrams. Such an approach was taken as it was not possible to include all details in one single diagram in an organized manner.

Even when this hierarchical modelling approach was taken, many logical details were not displayed such as queuing policies, conditional assignment of attributes, attribute matching, recording of variables etc. The activity cycle diagram provided a static display of model structure and interaction between its elements, but without showing dynamic changes of the models' states. This demonstrates the limitations of activity cycle diagrams and their inability to handle the full complexity of real life manufacturing systems. It is claimed that the same applies to other graphical methods for conceptual modelling such as petri nets or flow diagrams, which indicates that a graphical method that can handle the full complexity of real manufacturing systems does not exist.

Nevertheless, activity cycle diagrams can be considered as an important graphic modelling tool, but only when simple models without much detail are to be developed. This makes them more appropriate for educational and training purposes than for modelling real manufacturing systems.

It was realised as far as modelling the powder coating system is concerned, that although the activity cycle diagram was not able to incorporate full complexity of this system, some benefits were obtained. Due to these benefits, it was wise to construct the conceptual model prior to computer model development, independently of the software tool used.

3.6. COMPUTER MODELS

Several computer models have been developed, one for each simulator being evaluated. A preliminary investigation was carried out in order to chose one simulator which would enable detailed modelling of the system. At that moment in time WITNESS (AT&T ISTEL, 1991) appeared to be the most suitable package due to its ability to allow additional programming for modelling specific features of the system. This was necessary owing to the complexity of the system being modelled. When some other simulators with similar characteristics became available, it was realised that probably similar results could be obtained.

The WITNESS model has been used for achieving the objectives of the simulation study, defined from the perspective of the company in which the study has been carried out. Thus information presented in the subsequent sections regarding model verification, experimentation and analysis of simulation results apply to this model.

Other models were developed using the packages SIMFACTORY II.5 (CACI, 1990), XCELL+ (Conway *et al*, 1990) and ProModelPC (PMC, 1991). These models are 'quick and dirty', because they were developed only for the purpose of the software analysis. There was no need to include the level of detail as in the WITNESS model, which had to provide results as the basis for decision making in the real system being studied.

Information about the models provided in the following sub-sections relates to a description of simulators used in the case study and the analysis of the features modelled from a software point of view, whilst a general assessment of the simulators is presented in Chapter 5. A particular emphasis is given to the description of the problems experienced in modelling the complex real system being studied. Technical information about the computer models developed are provided in the corresponding appendices. The WITNESS model is discussed in sub-section 3.6.1, whilst the SIMFACTORY II.5 model is addressed in sub-section 3.6.2. Sub-section 3.6.3 presents the XCELL+ model, whilst the ProModelPC model is described in sub-section 3.6.4.

3.6.1. WITNESS Model

WITNESS is a data-driven manufacturing simulator supplied by AT&T ISTEL. Version 3.07 has been used for the evaluation of this simulator, as well as for the development of the model for the case study. WITNESS employs a visual interactive approach to modelling using pre-defined elements to represent manufacturing processes. Models are developed through three main phases: Define, Display and Detail.

Physical and logical elements are created in the Define phase. Each element has a separate form, which has to be filled in with relevant data. The names and quantity of the majority of elements have to be specified in this phase.

In the Display phase, a graphical display of the model is created. Icons representing elements are designed using the Icon Editor. The model layout is built up by positioning elements on the screen using a mouse. Characteristics such as the position of Parts and Labour, icons for physical elements and directional flows are chosen from menu forms. There is also a Screen Editor which enables the drawing of text, and different lines and shapes to enhance the graphical display of entire model.

In the Detail phase, the user has to specify how elements operate and how they interact with other elements. For each element, pre-defined forms are filled in with information such as cycle times, labour requirements, speeds, capacities and breakdown patterns. Data provided in this phase might include functions, statistical distributions and variables.

Pre-defined physical elements represent manufacturing equipment such as Parts, Machines, Labour, Buffers, Conveyors, Vehicles, Trucks, Tanks and Fluids. Logical elements handle the model's logic. For this purpose, the users can define part Attributes, Variables, use in-built Functions or write their own, specify Input and Output rules for part routing, write Actions to describe changes in the status of elements or specify Shift patterns.

There are also some elements used for storing and manipulating data and the simulation results: Files for reading the data or storing simulation results, Part Files used to store delivery schedules of the parts, Timeseries showing changes in a specific value over time, and Histograms showing the distribution of an expression or value.

Models can be run with animation or in the Batch mode. Multiple experiments can be performed automatically from the specially programmed Command File. Statistics for each element are collected automatically, and can be stored in the file. Any changes in the model can be incorporated immediately without the need for compilation of the entire model, which enhances the interactivity of WITNESS.

The model developed using simulation package WITNESS is the most comprehensive one. It includes a substantial level of detail incorporated in the model during the six months period of gradual model development. The final version of this model was presented to the clients and used for experimentation.

Information about the model presented here relates to the modelling aspects regarding the used software. Experience obtained in modelling certain features of the system being studied is discussed below. A technical description of the model is provided in Appendix B, whilst a program listing is provided in Section 7 of the thesis supporting material.

The first impression of the windows-based manufacturing simulator WITNESS is that it is user friendly. The menu driven modelling environment, graphical facilities, embedded manufacturing concepts and programming flexibility seemed to be suitable for modelling the powder coating system.

The first modelling step was to define the basic physical elements of the system. These elements relate to machines (painting booths), conveyors (an overhead conveyor chain was split into a number of smaller conveyors in order to distinguish different processing areas and enable the specification of conditions at branching) and workers. Following the definition of model elements, some additional features were specified. The data relating to the capacity, type of element and common basic features of elements were completed in the appropriate forms.

The next step was to construct a graphical display for each physical element using the Icon Editor. These icons were then combined together into a graphical display of the entire model. The Screen Editor was used for further refinement of this graphical display. These early steps of model development were quite straightforward. The data relating to the physical characteristics of the system were available, and these were fairly easily incorporated in the model, by filling in predefined forms.

More details were gradually added to the model. Three assembling machines were affixed to represent jigging of parts prior to loading on the conveyor, then two disassembling machine to represent unloading the parts from the flight bars, and seven masking stations. The part attributes were specified with fixed deterministic values. The graphical facilities of the package were very useful for testing the behaviour of the model.

The subsequent task was to begin development of the logical features. The logical elements had to be additionally programmed, using a special programming language accommodated by WITNESS. Thus many additional functions and conditions for routing were written, and stochastic values were assigned to part attributes.

The part attributes relate to the batch characteristics such as colour, batch size, batch priority, batch number, number of parts per flight bar, masking requirements, the number of booth on which the batch is to be painted, the total number of flight bars required for a particular batch, and the requirements for manual finish for painting if the parts within a batch are complex.

The routing conditions, in the form of input/output rules, were programmed for several model elements. For example, when the batch has to be routed to one booth for painting, attributes such as colour, batch size and priority are tested, and depending on the values of these attributes the batch is routed in the appropriate direction.

For many details of the model it was necessary to write functions (Hlupić and Paul, 1993b). Many of these functions invoke built-in functions provided by WITNESS. This part of model development was not easy and straightforward. One of the reasons for this is the improper description and illustration of the functions in the User manual. One of the examples of a use of a function was to assign part attributes. When stochastic values for part attributes were introduced, the functions were used to assign the same value of the part attribute for every part within a particular batch. Otherwise every part would have a different characteristic.

The development of logical features which enhance the power of the package takes time to learn. This development was a trial-and-error process, time consuming and sometimes discouraging. Several features of the system have not been modelled completely or have been omitted from the model due to the software capabilities (Hlupić and Paul, 1993c). Examples of these features are as follows:

(i) WITNESS allows the commencement of the machine setup only when the part arrives at the machine. This caused a lot of problems when this feature was modelled. In the system under consideration, setup (ie. cleaning of the painting booths and spraying guns) starts as soon as one batch is finished, and when it is known that the next batch to arrive at a particular booth is to be coated in a different colour. The production in the real system does not stop because of the setup. The batch that requires an alternative booth for coating is loaded whilst another booth is being cleaned. Modelling of the setup that was initiated when a new batch had already arrived at the machine caused an unacceptable delay in processing which had a significant impact on throughput. After many unsuccessful attempts to model this feature according to the situation in the real system, setup modelling was abandoned. This did not influence the validity of the results regarding throughput, but it had some impact on labour utilization. Since setup was omitted from the model, the performance of labour was slightly underestimated. This did not represent a problem, because the number of colour changes on each machine was monitored, and the time spent on setup during the simulated time could be calculated and added to the value of labour utilization (obtained in the report).

(ii) Another problem experienced occurred in an attempt to model a search of the buffer content (the number and type of parts positioned in the storage area), and pulling out from the buffer a batch with a specific characteristic. For example, after one batch was loaded, the attributes of the batches placed in the buffer ought to be investigated and then the batch with the same colour (as the previously loaded batch) loaded next. Such a selection is usually done in the real system except when there are batches of high priority, which have to be coated as soon as possible. This practice reduces the number of colour changes. When this feature was modelled, only the first part within a particular batch that satisfied the condition (the same colour) was pulled out from the buffer, and after that the program stopped. This part of the logic was abandoned and parts were pulled out from the buffer on a FIFO basis, which in the end did not influence the results significantly.

(iii) It is not possible to pull a part from a specific position of an element (e.g.

machine), although it is possible to push a part to a specific position. In the system, after baking in the oven, the parts are unloaded from any position on the conveyor apart from the newest three (which must be left for cooling of the parts after baking, prior to unloading). The parts are unloaded simultaneously from final positions. In the model, unloading stations that represented unloading activity could pull parts only from the front position on the conveyor, namely from one flight bar at a time. This did not have an important influence on the final results, but showed a weakness of the package.

The above mentioned examples describe particular features of the system being studied that have not been possible to model corresponding to the situation in the real system. Although in the end their abandonment has not influenced the results significantly, they are a good illustration that even such a flexible package as WITNESS does not allow the modelling of all specific details.

There are many other features of the package which have not eased the modelling process. Some of them will be probably eliminated in subsequent versions of the package. However, a lot of modelling time could have been saved, had not the shortcomings listed below been present:

(i) Buffers are passive, which means that it is neither possible to pull parts from, nor to push parts to, buffers. This caused problems when parts in the buffer had to be sent to masking stations. Several dummy machines were used to pull parts from the buffer and send them to masking stations, which additionally complicated the logic of the model.

(ii) There is no automatic increase of buffer capacity. This sometimes meant that only part of a batch was placed in the buffer and that another part was lost or the model simply stopped. A separate function was therefore written in order to check for free space in the buffer, before the batch was placed in it.

(iii) The maximum length of lines in the coding editor is 256 characters, which caused problems for some complex features of the model, such as testing whether a new batch could be pulled from the buffer, which should happen when a previous batch has been loaded completely, and all loading stations were empty and idle. In addition, there is no indicator of the cursor position within the line, so it is not possible to know when the limit is reached. Exceeding the limit is reported when the code is to be saved. In this case this can not be done, and it is not easy to determine which part of the line is surplus.

(iv) The modelling process could be speeded up if the package allowed the copying of physical and logical elements. Similar elements could then be modelled by copying ones already defined and making some minor alterations.

(v) The version of the package used for model development did not have copy, cut and paste facilities within the editor. So, a lot of similar and repetitive code had to be typed character by character.

(vi) Another example of the package weaknesses listed here, and perhaps the most important one, is the problem of software reliability. That is, the program stopped many times, and the only thing that could be done was to restart the computer. Examples of the occasions when this happened are as follows. In the case of logical mistakes, when a part attribute had to be accessed in the element which was empty. When the specification of the drive on which model had to be saved together with its status after experimentation was wrongly typed. In both cases, the message that occurred on the screen was "Application error. Terminating current application".

(vii) Similar problems occurred due to a shortage of computer memory. The software itself has a significant hardware requirement (4 MB of memory). In addition, the model developed was quite complex and during experimentation the memory space necessarily increased. Due to this memory problem, the program 'crashed' many times during experimentation, or when the buffer content had to be listed using the EXPLODE function.

These problems relating to software reliability and memory problems, which could have probably been eliminated by using a more powerful computer, cost a lot of time.

Despite all the problems experienced and deficiencies of the software discovered, the majority of important features were at the end successfully modelled. Those that have been omitted have emerged to be of no significant importance to model credibility and usability.

3.6.2. SIMFACTORY II.5 Model

SIMFACTORY II.5 is also a data-driven manufacturing simulator, based on the concept of visual interactive simulation. It is supplied by CACI Products Company. For the purpose of this research, version 5.0 has been used. SIMFACTORY II.5 is supplied with a mouse driven graphical user interface that enables the user to build graphical representations of models. Interactivity is provided both during the model development and during experimentation.

A simulation model developed by SIMFACTORY II.5 is considered as a network of stations, buffers, transporters and conveyors through which parts are moved requiring various operations, according to rules specified in the workflow or process plan.

The first step in designing the model is to define the layout of the factory. The layout consisting of processing stations, buffers or queues, transporters, transportation paths and receiving areas which accommodate arriving parts, is created by selecting and positioning icons that represent these components. As each icon is positioned on the screen the data that describe its characteristics are entered. These characteristics refer to the name of the element, its capacity, setup time etc. In addition, information such as labour requirements, breakdowns, shift patterns and interruptions might also be added to the model.

The next step is to specify processes performed on specific stations and to design a workflow of the parts. The workflow defines the path that each part takes through the model. It is created by building a list of processes and connecting their inputs and outputs. Once the workflow is specified, the model can be simulated. During simulation, a variety of reports are collected automatically and presented both in textual and graphical form at any time when the experiments are stopped.

SIMFACTORY II.5 also possesses the Expression Builder and the Data Graph. The Expression Builder enables pieces of code to be added into the model, which increases modelling flexibility. The Data Graph evaluates alternative distributions, fits real-world data to a theoretical statistical distribution, and analyzes output from SIMFACTORY II.5.

The SIMFACTORY II.5 model is 'quick and dirty', developed for the analysis of this simulator. This sub-section deals with the modelling aspects of SIMFACTORY II.5

and experience obtained in modelling the powder coating system. Appendix C includes a technical description of the SIMFACTORY II.5 model, whilst a program listing is included in Section 8 of the thesis supporting material.

The first modelling step was to define physical elements of the system. These elements include a receiving area, stations for loading/unloading, stations for coating, conveyors, and queues positioned at merging and branching points of conveyors (these queues were used for logical decisions regarding the routing of parts). The basic data related to a performance of these elements was specified in the appropriate forms. Graphical representations of the model elements were designed using the Icon Editor.

The subsequent step was to define processes performed on particular locations, as well as the workflow plan. The workflow plan specifies the flow of the parts through the system, from loading on the flight bar, to painting and unloading. Additional logical details were subsequently added to the model using the Expression Builder. These logical constructs were used for the assignment of part attributes, and for the conditional routing of parts according to the values of these attributes. For example, at the painting booth an expression on the entry condition is used to check the value of the attribute representing the batch size and colour in order to determine which parts (loaded flight bars) will be painted in a particular booth.

Modelling of the basic features was straightforward. However, it took some time to model the logical expressions properly. The main reason for this is inadequate description and illustration of the Expression Builder capabilities in the user manual. Furthermore, the Expression Builder allows the user only to select from available commands for code development without editing any of them or adding to them, which restricts the modelling flexibility.

Although it was not intended to develop a model of great complexity and level of detail, the limitations of this simulator, at least concerning the case study problem, were quite easily detected. Some of these limitations are as follows:

(i) It is not possible to model merging and branching of conveyors, and no push/pull rules can be defined for these elements. In order to model merging and branching, and to control the flow of the parts, dummy queues or machines have to be used to further guide the parts.

(ii) Setup starts only when a part arrives at a station (machine). This means that the situation regarding the setup in the case study system cannot be modelled properly.

(iii) Assembling processes (at assembling stations) can only have a constant number of inputs parts. In the system under consideration this number (number of parts per flight bar) is variable, depending on the characteristics of a particular batch.

(iv) Similarly, disassembling processes can only have a constant number of output parts.

(v) User-defined distributions cannot be used within the Expression Builder. A user-defined distribution was needed for the attribute giving the colour in which the part is to be painted. Since it was not possible to use a user-defined distribution, an approximation was made using the uniform distribution.

(vi) It is not possible to manipulate the code within the Expression Builder, and only the last command can be deleted. So, for example, if the beginning of the expression is to be changed, then all the following commands have to be deleted.

(vii) As there is no Screen Editor, dummy elements have to be defined in order to display icons on the screen. Since they are not needed for the logic, and are not used in the workflow plan, every compilation gives a warning about these elements. For example, pretreatment and oven areas were modelled as conveyors, but were defined only for the purpose of display.

(viii) Reliability is another shortcoming of SIMFACTORY II.5. On several occasions the program stopped for no apparent reason, and it was needed to restart the computer.

A general impression about this simulator is that it is very easy to use when the model is not too complex. Although a certain level of complexity can be handled by the Expression Builder, it becomes difficult to use this simulator when the complexity of logic rises.

3.6.3. XCELL+ Model

XCELL+ is a visual interactive data driven manufacturing simulator, supplied by the Pritsker Corporation. Release 4.0 has been used for this research. XCELL+ is, by its designer's own admittance, simple and capable of producing 'quick and dirty' models.

Simplicity and ease of use are favoured instead of generality and power. The user interface consists of a hierarchy of menus, accessible through the function keys F1 to F8. The action of each key is displayed at the bottom of every screen as it changes.

Elements of the model are represented by symbolic graphics. The factory is represented by a uniform grid of 'cells', and each element of the model occupies exactly one of these cells. Models are constructed by placing the elements in cells of the factory floor and connecting them by entering the appropriate data. Data such as input, output and cycle times is entered into the system by selecting the appropriate options from the menu and responding to the prompts given in the dialogue line.

Models can be developed by selecting the following elements: Receiving Area, which receives material and releases it to the model, and Shipping Area which receives material from the model and 'ships' it to the outside world. Process is an activity performed on a certain type of material, whilst Workcenter represents a facility required to perform a Process. Buffer is used to store material which is not currently processed. Auxiliary Resources are resources of a particular type used to perform particular Processes. Maintenance Facilities are used to model the maintenance of Workcentres. Links enable the modelling of routes for material movement.

The materials handling system is modelled by a Path, which is a connected sequence of cells over which a Carrier can move, a Control Point which is a cell that serves as the end-point of one or more Paths, and a Carrier, which is the vehicle that travels over the network of Paths.

The computer model developed using XCELL+ was also made only for software evaluation purposes. This 'quick and dirty' model does not contain much detail. The reason for this is not only the modelling purpose. Another reason is the characteristics of this simulator. Namely, XCELL+ is very user friendly and easy to learn, but not very robust.

This sub-section addresses modelling aspects regarding XCELL+ and the experience obtained in modelling certain features of the system being studied. Appendix D contains a technical description of this model, whilst a program listing is included in Section 9 of the thesis supporting material.

Whilst reading the documentation, it was immediately perceived that the main characteristic of XCELL+ is simplicity rather than flexibility and power. This makes this simulator much more suitable for 'quick and dirty' than for detailed modelling.

The first step of modelling the powder coating system was actually to determine the features that could be modelled. Since there is no explicit facility to model conveyors, an adequate approximation had to be found. After an analysis of several alternatives, it was finally decided to model conveyors as conveyor buffers, because they allow the specification of a Minimum Holding Time.

Once the basic logic and approach to modelling were established, physical elements of the model were designed. These elements include Buffers, Workcentres, Receiving Area, and Shipping Area. A graphical symbol for each element was positioned on the screen until the complete layout was obtained. Following design, characteristics of each element were specified, such as the capacity and conveyor time for Buffers, or details of Processes at Workcentres. Finally, all elements were connected by the specification of input and output Links, and the model was ready for experimentation.

The model was developed very quickly once the basic logic was determined. Some of the features were deliberately omitted such as labour requirements, breakdowns of Workcentres or shifts modelling, because they were not considered as relevant for the modelling purpose. There are also several characteristics that had to be excluded due to the software features. Examples of these features are as follows:

(i) Part attributes had to be eliminated due to the features of XCELL+. Because of that, it was not possible to model routing of batches to different booths according to their colour, batch size and priority. Therefore, only one painting booth was modelled. Although XCELL+ enables assigning one process to different Workcentres, modelling several booths would not provide useful information because it could not represent the real situation. It could only cause problems, by allowing different booths to be used at the same time.

(ii) Due to the inability to define part attributes, fixed deterministic values were used for the batch size and number of parts per flight bar. In addition, masking requirements and masking were not modelled. Since loading, unloading and masking are the main activities performed by labour (in addition to manual painting), labour was not included in the model. Every part that entered the system was regarded as a loaded flight bar, and the final results regarding throughput were multiplied by the fixed number of parts per flight bar.

(iii) Conveyors were modelled as the special type 'conveyor buffers'. For each segment of the conveyor chain, a buffer was defined with a capacity corresponding to the number of flight bars in that particular segment, and with a holding time representing the time which the flight bar needs to travel from the beginning to the end of that conveyor (segment of conveyor chain). An alternative approach could be to model conveyors with Paths and Carriers. In that case, for each flight bar on the conveyor, a separate Carrier would have to be defined, which could complicate the model but not provide better results.

(iv) Due to the symbolic graphics provided by XCELL+, and due to the approach taken to modelling conveyors, the graphical display of the model is not very realistic.

(v) Shifts were also not modelled. A certain approximation could be obtained by using Maintenance Centres to occupy Workcentres during the breaks. But, as the labour was omitted from the model, there was no need to model different shift patterns for different types of labour.

(vi) Breakdown of conveyors was not modelled because buffers (used to model conveyors) can not include this feature. However, it is not very likely that this feature would have influenced the results very much had it been included.

Despite all the limitations of the software, the model was obtained rapidly, and was capable of producing some useful information. This model could, for example, be used for flow analysis. The changes in throughput and flow time can be estimated due to a change in conveyor speed or capacity. The model developed showed that XCELL+ is indeed suitable for 'quick and dirty' modelling, and is also more appropriate for modelling flow lines rather than job shop types of manufacturing systems.

3.6.4. ProModelPC Model

ProModelPC is also a data driven, discrete-event manufacturing simulator. This simulator is supplied by the Production Modelling Corporation. For the purpose of this research, release 5.0 has been used. The main characteristics and a description of model development using ProModelPC are provided in this section.

ProModelPC is a visual interactive simulator. It provides a combination of a menu driven modelling environment and the flexibility of additional programming in the form of Turbo Pascal subroutines. Built-in modelling constructs enable the user to select specific logical features such as routing strategies or control logic for materials handling system.

The main types of physical modelling elements are Parts, Resources, Transporters and Conveyors. Parts refer to items that are processed in a system. Resources include items that are used for producing parts: routing locations (i.e. machines, storage areas, queues etc.), where parts are sent for performing operations, and general resources (i.e. operators, tools, fixtures etc.), which may be used during an operation or part movement. Transporters refer to mobile resources such as AGVs, robots, and cranes, that are used for moving parts between locations on a defined path. Conveyors (accumulating and non-accumulating) are also used for moving parts between routing locations.

When models are defined using the Automatic Model Build mode, ProModelPC guides the user step-by-step through each of the definition modules. Each of these steps is performed using the menus which have to be completed. These menus are automatically invoked on the basis of information supplied in a previously completed menu. Once all menus are completed, a graphical layout can be optionally defined.

Models may be first defined graphically as well. In this case, locations of the model such as machines, conveyors or transporter paths are positioned on the layout screen, and after that the flow of parts through the model is defined. Graphical representations of locations can be either chosen from the supplied icons or they can be developed using the Icon Editor.

The definition of the model both in the Automatic Model Build mode and in non-automatic mode begins with a specification of the Routing menu. This menu defines the part flow logic including location and operation sequences, operation times, part input-output relationships such as assembling or disassembling, routing conditions, moving times etc.

Following the specification of routing, the Part Scheduling menu, which refers to the mechanism for introducing parts to the system, has to be completed. Data to be supplied includes part name, location where the parts enter the system, batch size, arrival frequency, number of arrivals, and the starting time of the first arrival.

The next step in model development is to define capacities of model elements, downtimes for any location, resources, user defined distributions, characteristics of transporters and conveyors if they are included in the model, and to specify simulation parameters.

The ProModelPC model is also 'quick and dirty', developed for the evaluation of this simulator. The details included in this model are similar to those incorporated in the SIMFACTORY II.5 model. The main reason for this are that the modelling purpose, and level of flexibility are similar for these two simulators.

This sub-section provides information related to modelling the case study system using the ProModelPC. Appendix E contains a technical description of this model, whilst a program listing is provided in Section 10 of the thesis supporting material.

With regard to the modelling approach, this simulator differs from the other three simulators analyzed. Instead of filling in forms with relevant data for each model element and connecting the elements via input/output or push/pull rules, in ProModelPC all information regarding the flow of parts is provided within a routing section in the form of a table.

Completion of a routing section was the first step in model development. This relates to a specification of part flow logic including a definition of processing and dummy locations, duration of operations, routing conditions, time needed for moving the parts from one location to another etc. The subsequent modelling steps relate to a specification of distributions (used in the routing section), part scheduling data, conveyors data, and a specification of simulation parameters. Once the logic of the model was completed, a graphical display was designed using the standard icons provided within the Icon Editor.

Some features of the system under consideration were deliberately omitted because of the modelling purpose (as was the case for the SIMFACTORY II.5 model). For example, masking, labour requirements, breakdowns of machines and conveyors, and shifts patterns were not included in these models. Those features that were incorporated provided enough experience to perceive the main strengths and weaknesses of the simulators analyzed. Several software limitations and problems were discovered during the development of the ProModelPC model, such as:

(i) Setup can be activated only when the part to be processed arrives at a particular routing location, which means that the setup circumstances that apply to the case study system cannot be adequately modelled.

(ii) Although ProModelPC is in general reasonably flexible, unless external Turbo Pascal subroutines are used, it is not easy to handle complex logic. For example, there is a limitation in the length of commands (less than 20 characters) that can be specified within the routing module, so each condition or command has to be written on a subsequent line. This makes complex logic not easy to specify and follow.

(iii) An internal code that can be used is also somewhat limited. For example, it is not possible to use multiple IF statements (IF-THEN-ELSEIF-...) and loops. Such programming constructs can be designed only by using Turbo Pascal subroutines.

(iv) Another problem perceived was the impossibility of building a partially developed model. At least a simplified version of a model has to be completed in one session (the last row of the routing section has to contain an EXIT as the next location for parts).

(v) ProModelPC has a reliability problem. On several occasions the program stopped for no apparent reasons.

(vi) Perhaps one of the most significant difficulties experienced was model debugging. There are no error messages during model development. Only when model is compiled, is there an indication that there is an error in the model, but there is no information on error type and position.

Overall ProModelPC is similar to SIMFACTORY II.5, in that it is reasonably easy to use for simple models. The possibility of linking to Turbo Pascal subroutines provides, in theory, substantial flexibility. However, it is questionable how effectively this flexibility can be utilized, especially because of the inadequate testing facilities.

3.7. MODEL CONFIDENCE

Establishing confidence in the model is one of the most difficult and time consuming phases of the simulation process. At this stage it is necessary to test if the model is valid ie. an appropriate representation of the system being studied (Law and Kelton, 1991). In practice, this actually means that the analyst tries to find out whether the model is incorrect rather than trying to prove otherwise.

Some general information about model validation and verification are given in sub-section 3.7.1, whilst an analysis of the methods applied and the results obtained are presented in sub-section 3.7.2.

3.7.1. Model Validation and Verification

Two different issues are relevant for establishing a model confidence: validation and verification. Whilst *validation* refers to determination whether a conceptual simulation model is an appropriate representation of the real world system under study, *verification* relates to testing that the program does as it is supposed to.

The conceptual model of the powder coating system developed using the activity cycle diagrams was thoroughly checked not only by modeller, but also by several production engineers and simulation specialists. The final version was accepted as adequate to provide basic information about the system. The reasons for developing the conceptual model with that level of detail have been described in section 3.5.

However, the structure of the conceptual model cannot be transferred directly to the computer model due to the features of the software used. Verification was not performed in order to test the correspondence between the conceptual and computer model (because the computer model was not directly developed from the conceptual model), but in order to give confidence that the computer model (ie. WITNESS model) could be accepted as an adequate representation of the real system being modelled.

The importance of verification of the model of the powder coating system was emphasised by its intended use for decision making in the real system. Results of the experimentation should have been used for implementing the changes in the system, which could provide better productivity of the system. Due to these facts, special concern was given to model testing and numerous verification techniques were applied. Appendix F presents a description of the verification techniques used for testing the model of the system under consideration.

3.7.2. Analysis of Verification

Special attention was paid to establishing confidence in the model of the system being studied. One of the main reasons for this was its intended use for decision making by production managers. The financial implications of these decisions can be considerable as well as their importance to future modification of the powder coating system. The results of the experiments with the model should indicate which changes are to be made in the existing powder coating system in order to increase its productivity.

A variety of verification tests have been applied and all of them together provided considerable confidence that the computer model could be accepted as an approximation of the real system being modelled. The final, and the most important test, concerned a comparison of the output from the model with the real data, which revealed that these two sets of data differed by less than 1%. According to such results, it was strongly believed that the model had a practical use and could provide useful information about the system.

3.8. EXPERIMENTS AND RESULTS

The final version of the WITNESS model was used as the basis for analysis of various production alternatives. The process of computer model verification confirmed the assumption that the WITNESS model (as the most detailed one) could provide the results that are the closest to the real values obtained in the powder coating system. Whilst the preliminary results obtained from the WITNESS model regarding the throughput differed from the real values by approximately 1%, the results from the other three simpler models differed from the real data by 30% on average. Because of this, it was assumed that the results obtained by experimentation with the WITNESS model should be sufficient for satisfying the simulation objectives defined by BICC VERO Electronics.

The production engineers proposed testing several changes to the system, which resulted in the simulation of five different versions of the model, in addition to simulating the present situation (Hlupić and Paul, 1993d).

Consequently, six different models were simulated. The main aim of experimentation was to see the impact of changes on throughput in the system, but machine and labour utilizations were also considered. Model 1 represented the present situation. In model 2, the number of parts loaded on flight bars was doubled, because it is possible to jig parts together, and paint them only on one side.

Model 3 simulated the use of three automatic spray booths, with a new colour mix. At present, there are 19 different colours which complicates the operation of the system, because after each colour change booths have to be cleaned for about 45 minutes. In this new version, only 5 colours were used.

Model 4 simulated the automatic spraying of 25% of the parts that are sprayed manually at the moment. Similarly, model 5 simulated the automatic spraying of 50% of the parts that are now sprayed manually, whilst model 6 simulated the automatic spraying of 75% of the parts that are now sprayed manually.

All models were simulated under the same conditions. There are 22 random variables in the model, and for each model three experiments with different random number seeds were run. The average values from these three experiments were considered. The model simulated the performance of the system over 40 days, with a warm up period of 7 days. Table 3.1 shows the average total throughput obtained for the six models, whilst some additional results are presented in Appendix G.

Table 3.1 Average total throughput obtained for 6 models

	MODEL 1	MODEL 2	MODEL 3	MODEL 4	MODEL 5	MODEL 6
TOTAL THR.	39753	62358	42057	41352	41252	41204
INCREASE	-	56%	5.7%	4%	3.7%	3.6%

The last row of the table shows the increase in throughput obtained in models 2-6 in comparison with the throughput obtained by model 1. The best results were achieved by model 2, with an increase in throughput of 56%. Even higher increase in throughput could be expected in this model, because the number of parts per flight bar was doubled. The anticipated increase was reduced because it takes longer to load and unload larger numbers of parts. The model stops the conveyor to allow time for load and unload. In order to avoid this, loading should be modelled as off-line jigging, utilizing more workers to make sure that the conveyor does not wait to be loaded or unloaded. This probably means that in the real system employing more workers might be needed if the number of parts per flight bar is increased.

The other four models also provided an increase in throughput, though less significantly. Model 3 with three automatic booths and the new colour mix provided an increase in throughput of almost 6%. Models 4,5 and 6, where a part of the components were sprayed automatically on manual booths, gave an increase in throughput of about 4%.

The results presented show that different changes in the way of running the system have different impacts on the measures of performance. Since the main goal of the study was to investigate the ways of increasing the throughput, the results regarding this measure of performance are presented and discussed.

The results demonstrate that the greatest improvement will be achieved by jigging the parts together on flight bars, as was simulated in model 2. Although this can probably be easily accomplished for the majority of part types, the production engineers will have to find a way to jig the large components which currently require two flight bars. Even if only some components can be doubled on flight bars, a significant improvement in throughput might be gained (Hlupić and Paul, 1993e). Further experimentation could perhaps provide more precise estimates of those improvements and the number of workers needed.

Introducing the third booth for automatic spraying requires some additional investment in production equipment, whilst reducing the range of colours may be opposed by customers. Nevertheless, an improvement of 6% can have a significant impact on system efficiency, especially in the long run.

The last three models gave an increase in throughput of about 4%. Such a level of production can be achieved without any additional investment by transferring some batches to one of the automatic booths.

These are some of the changes that could improve the performance of the system being modelled. The results obtained have given a good insight to production managers as to which direction they would have to follow in efforts to improve the performance of the system.

3.9. SUMMARY

This chapter presents the case study carried out at BICC-VERO Electronics, Eastleigh. This study is of a simulation model of an automated system for the electrostatic powder coating of metal components. The study involves an implementation of all phases of the simulation process: identification of a real world problem, definition of problem logic, data collection and analysis, development of a formal or conceptual simulation model, development of computer models, model validation and verification, experimentation, and analysis of simulation results. The main purpose of this study, within the context of this thesis, was to use and analyze several manufacturing simulators by modelling a real manufacturing system. Another purpose, defined by the company in which the study was carried out, was to examine the possibilities for improving the throughput of the system under consideration.

The system being modelled was described from the modelling point of view. Logical features and physical components which were estimated as important for modelling discrete processes performed in this system were considered. Other features, such as recycling of coating powder or sources of energy were omitted from consideration because of their irrelevance for the purpose of this study.

Several methods of data collection were used in this research: data collection on the shop floor, where workers filled up the forms especially designed for this purpose, interviews with the employees, observation of the system and measuring the durations of operations on the shop floor, and analysis of the relevant literature.

A conceptual model of the electrostatic powder coating was produced using activity cycle diagrams. Due to the complexity of the system under consideration, the conceptual model was developed hierarchically. Two activities from the main activity cycle diagram (loading/unloading and painting) were further modelled in more detail. The analysis of the applicability of activity cycle diagrams for modelling complex manufacturing systems reveals that, despite all the advantages of this method, it becomes too limited as the complexity of the systems rises.

Several computer models were developed. The most detailed was the model developed using the WITNESS manufacturing simulator. This model was experimented with in order to investigate the possibilities of improving the throughput of the powder coating system. Other models developed using the manufacturing simulators SIMFACTORY II.5, XCELL+ and ProModelPC are 'quick and dirty', and were produced only for the purpose of software analysis.

The models developed reveal similarities and differences between the software tools analyzed. Although all software packages used for modelling are visual, interactive and belong to the class of data-driven manufacturing simulators, it is evident that the models developed are different. There are differences in the modelling approaches. For example, the ProModelPC model was developed by specifying the relevant data in the form of tables. For other simulators, model elements had to be separately defined, and each element had a separate form into which data was entered.

With regard to the graphical representation of models, the XCELL+ model comprised symbolic graphics, whilst graphical layouts of other models included icons. Similarly, the physical layout of the XCELL+ model is the least realistic mainly due to an inability to explicitly model conveyors. In addition, the XCELL+ model is the only one in which part attributes were not modelled, due to the features of this simulator.

The ProModelPC model was the most difficult to debug, but it was the only one that could handle merging and branching of conveyors, which simplified the modelling process. On the other hand, the WITNESS model was the easiest to verify due to the many testing facilities provided by this simulator.

WITNESS and ProModelPC could model, for example, a variable input quantity for the assembling machines to represent loading the parts on the flight bars, whilst XCELL+ and SIMFACTORY II.5 models could only have a constant input quantity to the machines. On the other hand, none of these simulators could initiate setup after a certain batch has been painted nor could they pull a specific batch from any position within the storage. In addition, all simulators had a reliability problem.

Experience obtained in modelling the powder coating system using manufacturing simulators indicates that these software tools can easily become too limited for complex real life problems. This fact supports a need for further improvement of these simulation software products, which is addressed in Chapter 6.

CHAPTER 4. CRITERIA FOR THE EVALUATION OF SIMULATION PACKAGES

4.1. INTRODUCTION

This chapter establishes a number of criteria that can be used for the evaluation of simulation packages in general, and especially for the evaluation of packages for simulation of manufacturing systems.

According to Law and Kelton (1991), simulation packages can be classified as simulation languages and simulators. However, in this research the term 'simulation package' is mainly used when discussing data driven simulators, and all criteria are derived and described from the perspective of this type of simulation software. On the other hand, some of the criteria (eg. criteria related to pedigree, user support or financial and technical features) might also be used for the evaluation of simulation languages, and this was the main reason why the more general expression 'simulation package' has been used.

Criteria listed in this chapter represent a comprehensive evaluation framework that can be used for package selection by potential buyers as well as for guidance in further software development and improvement.

The publications on software evaluation analyzed in Chapter 2 provide a limited number of software evaluation criteria. This study is believed to be more comprehensive than earlier studies, providing a list of more than 310 evaluation criteria. These criteria were derived mainly from practical experience obtained by using different manufacturing simulators for a case study, from common sense, and some of them were arrived at during analysis of the literature.

The results of the survey (see Chapter 6) also provided several criteria. The survey has been carried out when the main groups of criteria have already been established, and survey results did not initiate a need for a new group of criteria. Those individual criteria that were derived from the survey could be easily included within the existing groups of criteria. An information about the origin of each criteria is provided in Appendix H.

Some of the criteria do necessarily overlap, for example ease of use and quality of documentation. It may be arguable therefore, as to why a specific criterion is included in one sub-section and not in another. There are also some criteria that are more general, comprising several specific criteria. For example, ease of use of the package depends on many factors such as the quality of documentation, on-line help and tutorials, availability of a help line and the experience of the user.

However, all these criteria are listed separately to emphasize their importance. In addition, the aim was to derive a comprehensive list of the evaluation criteria that can be of practical use rather than to invent a strict classification of criteria.

The evaluation criteria are classified into two main groups. The first group (section 4.2.) contains general criteria, which can be applied to the evaluation of any simulation package, regardless of its application area. Thus, these criteria can be used in part for the evaluation of manufacturing simulators, but they can be also used for the evaluation of any general purpose package or specialised simulator. Criteria within this group are further classified into sub-sections, according to their character (sub-sections 4.2.1-4.2.13). The second group of criteria, presented in section 4.3, comprises criteria specific to the evaluation of manufacturing simulation packages. These criteria are also further grouped in sub-sections (4.3.1-4.3.5), corresponding to their function. Criteria in each group are discussed generally, from the perspective of their classification in sub-sections, whilst a brief description of every individual criterion is provided in Appendix I.

Section 4.4 provides a discussion on a proposed hierarchy of criteria and their usefulness for the selection either of a package to be used for education or a package for use in industry. Whilst this section provides some initial basic ideas on the use of criteria, Chapter 6 presents a more detailed discussion of this issue.

4.2. GENERAL CRITERIA

These criteria can be applied to the evaluation of any general or special purpose simulation package. However, they will be described from the point of view of manufacturing applications, to fulfil the thesis objectives.

The criteria within this group are 'naturally' grouped according to their character. Figure 4.1 shows a proposal for mapping these groups of criteria to phases of the simulation process. This shows where criteria representing certain features of simulation software can influence each phase of this process. For example, criteria regarding *general*

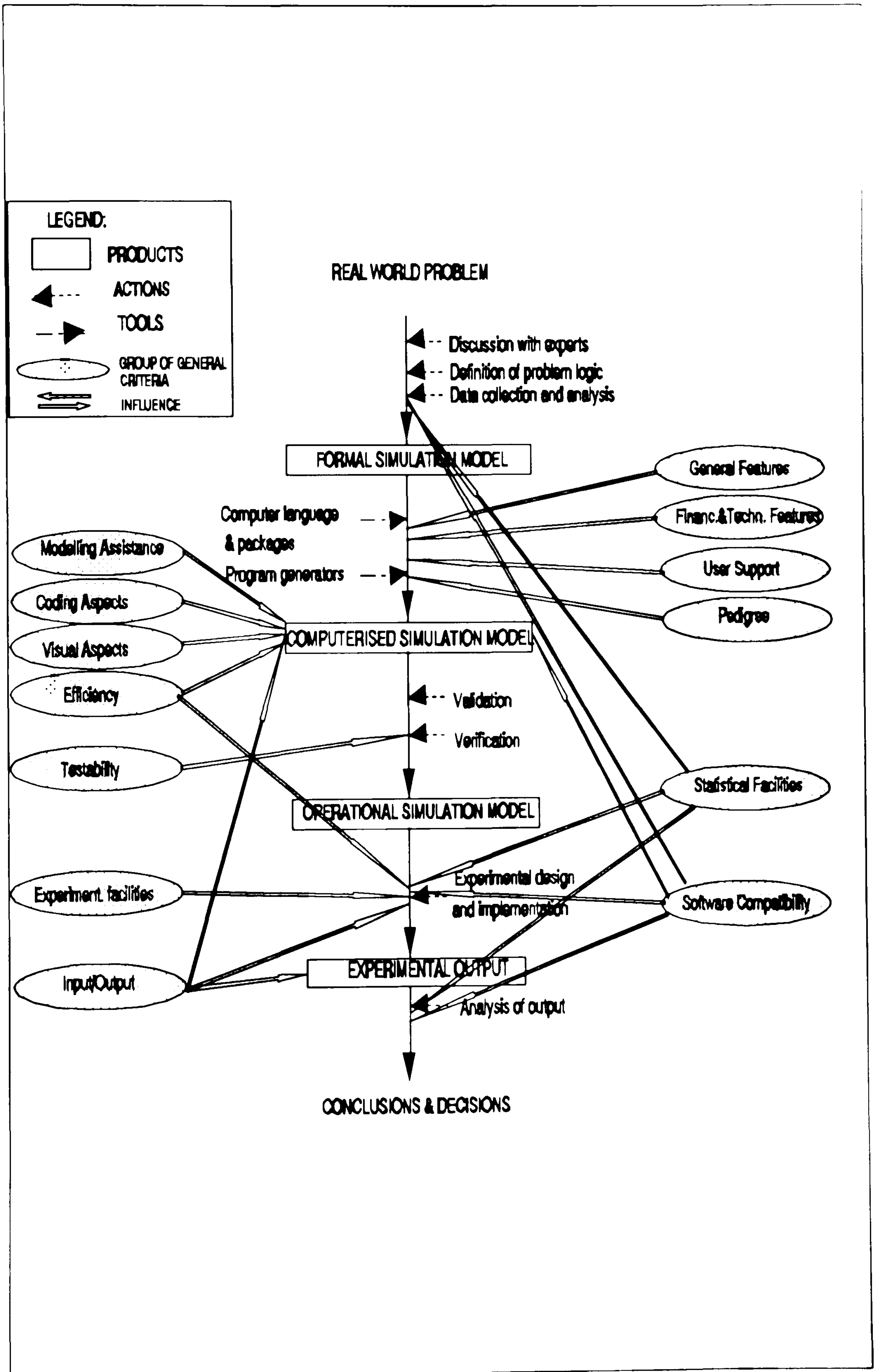


Figure 4.1 Mapping of groups of general criteria to phases of simulation process

features, *financial and technical features*, *user support* and *pedigree* could initially influence selection of the software tool for computer model development. Characteristics and speed of computer model development are influenced by criteria concerning *modelling assistance*, *visual aspects*, *coding aspects*, *efficiency* and the manner of data *input*. Software features regarding *testability* have a significant impact on model verification, whilst *experimentation facilities* might affect experimental design and experimentation. *Input/output* features are in charge of output reports both during and after experimentation. *Statistical facilities* might be used for data analysis, experimentation (eg. generation of random numbers) and for analysis of simulation output. Finally, *software compatibility* might be useful for the visual appearance of model (integration with CAD software), for data analysis (integration with statistical packages), for experimental design and experimentation (integration with expert systems and data base management systems) or for output analysis (integration with spreadsheet packages, expert systems, statistical packages).

This chapter is set out as follows. Sub-section 4.2.1 contains criteria regarding general modelling features, whilst sub-sections 4.2.2 and 4.2.3 comprise criteria related to visual and coding aspects respectively. Efficiency and modelling assistance related criteria are presented in sub-sections 4.2.4 and 4.2.5, whilst testability and software compatibility associated criteria make up sub-sections 4.2.6 and 4.2.7.

Sub-section 4.2.8 contains criteria connected to model input/output, whilst experimentation and statistical facilities are included in sub-sections 4.2.9 and 4.2.10. Finally, user support, financial and technical features and pedigree of the software make up sub-sections 4.2.11, 4.2.12, and 4.2.13 respectively.

4.2.1. General Features

Criteria included in this group describe general features of the package. Most of these criteria relate to modelling aspects such as the type of formal logic needed for modelling (if any), the method of changing the state of the model, the level of modelling transparency etc.

There are also some criteria that evaluate the level of experience and formal education in simulation needed from the user, and examine how easy it is to learn and use the package. Table 4.1 comprises the criteria categorized in this group together with a possible classification of the packages from the point of view of each criteria.

Table 4.1. Criteria for general features

GENERAL FEATURES		
	CRITERIA	**CLASSIFICATION**
1.	Type of package	- Data driven simulator - Data driven simulator with additional programming - Programming language
2.	Type of simulation	- Discrete event - Continuous - Both
3.	Purpose	- General purpose - Manufacturing oriented - Other special purpose
4.	Terminology	- Manufacturing terminology - Other
5.	Modelling approach	- Process based - Activity based - Event based - Three phase - Combination
6.	Formal logic	- Required - Not required
7.	Representativeness of models	- High - Medium - Low
8.	Ease of conceptualization of simulation logic	- Easy - Not easy
9.	Modelling transparency	- High - Medium - Low
10.	Hierarchical model building	- Possible - Not possible
11.	Run-time applications	- Provided - Not provided

12. Conceptual model generator	- Provided - Not provided
13. The length of entity name	- Long - Medium - Short
14. Entity name	- User defined - System defined
15. Experience required for software use	- None - Some - Substantial
16. Formal education in simulation required for software use	- None - Some - Substantial
17. User friendliness	- High - Medium - Low
18. Ease of learning	- Easy - Not easy
19. Ease of using	- Easy - Not easy
20. Initialization	- Possible - Not possible
21. Specification of time units	- Possible - Not possible
22. Specification of length measures	- Possible - Not possible

4.2.2. Visual Aspects

Graphical presentation of simulation models and animation of simulation are very important characteristics of simulation software. Criteria included in this group concern the type and quality of graphical facilities provided by the package.

These criteria evaluate, for example, whether it is possible to perform animation of simulation experiments, the types of animation provided by the package, and whether it is possible to manipulate icons. Table 4.2 shows the criteria included in this group and a feasible classification of the packages in respect to these criteria.

Table 4.2 Criteria for visual aspects

VISUAL ASPECTS		
CRITERIA		CLASSIFICATION
1.	Animation	- Possible - Not possible
2.	Type of animation	- Full animation - Semi-animation (state-to-state)
3.	Timing of animation	- Concurrent animation - Post-processed animation
4.	Type of graphical display	- Icons - Symbols - Characters
5.	3-D graphics	- Provided - Not provided
6.	Integrity of graphics	- Integrated to the package - Separate
7.	Animation layout development	- Concurrent with model development - Before model development - After model development - Flexible
8.	Multiple screen layout	- Possible - Not possible
9.	Animation with visual clock	- Provided - Not provided
10.	Icon editor	- Provided - Not provided
11.	Screen editor	- Provided - Not provided
12.	Ease of icon development	- Easy - Not easy
13.	Ease of using screen editor	- Easy - Not easy

14.	Type of icons	- Bit mapped - Pixel based
15.	Icon library	- Provided - Not provided
16.	Merging icon files	- Possible - Not possible
17.	Resizing of icons	- Possible - Not possible
18.	Rotating of icons	- Possible - Not possible
19.	Changing the colour of the icons	- Possible - Not possible
20.	Zoom function	- Provided - Not provided
21.	Panning	- Provided - Not provided
22.	Switching on/off the graphic	- Possible - Not possible
23.	Switching between screens	- Possible - Not possible
24.	Switching between character and icon graphics	- Possible - Not possible
25.	Print screen facility	- Provided - Not provided
26.	Virtual screen	- Provided - Not provided
27.	Indication of the elements' status	- Provided - Not provided
28.	Changing the colour of the elements' status display	- Possible - Not possible
29.	Limitation on number of displayed icons	- Exists - Does not exist
30.	Number of icons stored in icon library	- Large - Medium - Small
31.	Change of icons during simulation	- Possible - Not possible

32.	Icons with multiple colours	- Provided - Not provided
33.	Easy copying of icons	- Possible - Not possible

4.2.3. Coding Aspects

The possibility of additional coding might be very important feature of a package. This feature determines the flexibility and robustness of the software, which is especially valuable when complex systems are to be modelled.

Criteria included in this group determine whether the package allows additional programming, if access to the code is possible, the characteristics of the added code, the programming concepts supported etc. Table 4.3 comprises the criteria included in this group and a viable classification of the packages regarding these criteria.

Table 4.3 Criteria for coding aspects

CODING ASPECTS		
CRITERIA		CLASSIFICATION
1.	Programming flexibility	- Provided - Not provided
2.	Program generator	- Provided - Not provided
3.	Access to source code	- Possible - Not possible
4.	Readability of source code	- High - Medium - Low
5.	Readability of added code	- High - Medium - Low
6.	Self-documentation of added code	- High - Medium - Low

7.	Precision of added code	- High - Medium - Low
8.	Comprehensiveness of added code	- High - Medium - Low
9.	Link to a lower language	- Possible - Not possible
10.	Data storage, retrieval and manipulation facilities	- Provided - Not provided
11.	Quality of data storage, retrieval and manipulation facilities	- High - Medium - Low
12.	Built-in functions	- Provided - Not provided
13.	User functions	- Possible - Not possible
14.	Global variables	- Provided - Not provided
15.	Names of functions, variables and attributes	- User defined - System defined
16.	Writing comments for logical elements	- Possible - Not possible
17.	Type of time variable	- Real - Integer
18.	Type of translation	- Compilation - Interpretation
19.	Text/code manipulation	- Possible - Not possible
20.	Length of the lines in coding editor	- Large - Medium - Small
21.	Support of programming concepts	- Provided - Not provided
22.	Quality of the support for programming concepts	- High - Medium - Low

23.	Object oriented programming concepts	- Provided - Not provided

4.2.4. Efficiency

Criteria classified in this group determine the effectiveness and the power of simulation software. Efficiency is expressed both by the capability of the software to model a variety of complex systems and by the characteristics which can save time needed for modelling and improve the quality of modelling such as model reusability, reliability, compilation and execution time and multitasking. Table 4.4 includes the criteria categorized in this group and a possible classification of the packages considering these criteria.

Table 4.4 Criteria for efficiency

EFFICIENCY		
CRITERIA		CLASSIFICATION
1.	Robustness	- High - Medium - Low
2.	Level of detail	- High - Medium - Low
3.	Number of elements in the model	- Large - Medium - Small
4.	Model reusability	- Possible - Not possible
5.	Model status saving	- Possible - Not possible
6.	Automatic saving	- Possible - Not possible
7.	Interaction	- Possible - Not possible

8.	Adaptability	- High - Medium - Low
9.	Multitasking	- Possible - Not possible
10.	Model chaining ie. linking outputs from different models	- Possible - Not possible
11.	Exit to the operating system within the package	- Possible - Not possible
12.	Compilation time	- Long - Medium - Short
13.	Model execution time	- Long - Medium - Short
14.	Case sensitivity	- Provided - Not provided
15.	Conversion of numbers (real v integer)	- Provided - Not provided
16.	Queuing policies	- Provided - Not provided
17.	Number of queuing policies	- Large - Medium - Small
18.	Time scale for model building	- Large - Medium - Small
19.	Reliability	- High - Medium - Small
20.	Pre-existing generic models	- Provided - Not provided
21.	Merging of models	- Provided - Not provided
22.	Editing partially developed models	- Possible - Not possible
23.	Automatic model building	- Provided - Not provided

24.	Ease of model editing	- Easy - Not easy
25.	Specification of part flow by a mouse	- Provided - Not provided

4.2.5. Modelling Assistance

Criteria systematized in this group evaluate the type and level of assistance provided by the package during modelling. For example, these criteria examine the comprehensiveness of prompting, on-line help if it is provided, whether the package enables modular model development and writing the documentation notes (this feature enables writing a documentation concurrently with the model development), and whether the model and data can be separated. Criteria related to modelling assistance are listed in Table 4.5, together with a feasible classification of the packages regarding these criteria.

Table 4.5 Criteria for modelling assistance

MODELLING ASSISTANCE		
CRITERIA		CLASSIFICATION
1.	Prompting	- Provided - Not provided
2.	Quality of prompting	- High - Medium - Small
3.	Modularity	- Possible - Not possible
4.	Model and data separation	- Possible - Not possible
5.	Use of mouse	- Possible - Not possible
6.	On-line help	- Provided - Not provided

7.	Quality of on-line help	- High - Medium - Low
8.	Documentation notes	- Provided - Not provided
9.	Quality of facility for documentation notes	- High - Medium - Low
10.	Text editor as integral part of the package	- Provided - Not provided
11.	Automatic editing of data	- Provided - Not provided

4.2.6. Testability

This group comprises criteria that examine which facilities for model verification are provided by the package. These facilities include error messages, displays of the values of logical elements such as functions and variables, the possibility of obtaining special files for verification such as list, trace and echo files, provision of step function etc. Table 4.6 contains the criteria and classification of the packages regarding testability of the models.

Table 4.6 Criteria for testability

TESTABILITY		
CRITERIA		CLASSIFICATION
1.	Logic checks	- Provided - Not provided
2.	Error messages	- Provided - Not provided
3.	Quality of error messages	- High - Medium - Small

4.	Moment of error diagnosis	- Model entry - Compilation - Model execution - Combination
5.	Ease of debugging	- Easy - Not easy
6.	Display of function values	- Possible - Not possible
7.	Display of attributes	- Possible - Not possible
8.	Access to attributes	- Possible - Not possible
9.	Display of variables	- Possible - Not possible
10.	Display of element's state	- Possible - Not possible
11.	Dynamic display of capacity	- Possible - Not possible
12.	Display of the workflow path	- Provided - Not provided
13.	Display of events on the screen	- Provided - Not provided
14.	Display of part position within element	- Provided - Not provided
15.	Facility for immediate user actions	- Provided - Not provided
16.	List files	- Provided - Not provided
17.	Echo	- Provided - Not provided
18.	Trace files	- Provided - Not provided
19.	Explode function	- Provided - Not provided
20.	List of used elements	- Provided - Not provided
21.	Backward clock	- Provided - Not provided

22.	Step function (event to event jumping)	- Provided - Not provided
23.	Flow analysis	- Provided - Not provided
24.	Audible alarms	- Provided - Not provided
25.	Rejection of illegal inputs	- Provided - Not provided

4.2.7. Software Compatibility

These criteria evaluate whether the package can be interfaced to other software systems, in order to exchange data with these systems. This feature can considerably enhance the capabilities of the package, especially when complex real systems are modelled. Table 4.7 contains the criteria together with classification of the packages regarding software compatibility. The criterion related to integration with programming languages is not included in this group of criteria, because it is contained in the coding aspects.

Table 4.7 Criteria for software compatibility

SOFTWARE COMPATIBILITY		
CRITERIA		CLASSIFICATION
1.	Integration with spreadsheet packages	- Possible - Not possible
2.	Integration with statistical packages	- Possible - Not possible
3.	Integration with word processors	- Possible - Not possible
4.	Integration with CAD software	- Possible - Not possible
5.	Integration with DBMS	- Possible - Not possible

6.	Integration with expert systems	- Possible - Not possible
7.	Integration with MRP II software	- Possible - Not possible
8.	Integration with scheduling software	- Possible - Not possible

4.2.8. Input/Output

Criteria included in this group investigate how the user can present the data to the package and the type and quality of output reports provided by the package. These criteria evaluate, for example, whether the package has a menu driven interface, whether static and dynamic output reports are provided, and how understandable these reports are. Table 4.8 shows the criteria categorized in this group and a classification of the packages regarding input and output of data.

Table 4.8 Criteria for input/output

INPUT/OUTPUT		
CRITERIA		**CLASSIFICATION**
1.	Menu driven interface	- Provided - Not provided
2.	Pull down menus	- Provided - Not provided
3.	Type of menu selection	- By mouse - By keys - Other
4.	Selection buttons	- Provided - Not provided
5.	Dialogue boxes	- Provided - Not provided
6.	Multiple inputs	- Possible - Not possible

7.	Multiple outputs	- Possible - Not possible
8.	General output reports	- Provided - Not provided
9.	Static graphical output	- Provided - Not provided
10.	Dynamic graphical output	- Provided - Not provided
11.	User defined output	- Possible - Not possible
12.	Automatic rescaling of histograms and time series	- Provided - Not provided
13.	Quality of output reports	- High - Medium - Low
14.	Understandability of output reports	- High - Medium - Low
15.	Periodic output of simulation results	- Provided - Not provided
16.	Availability of results before end of simulation	- Provided - Not provided
17.	Input data reading from files	- Provided - Not provided
18.	Writing reports to files	- Provided - Not provided
19.	Writing reports to printer	- Provided - Not provided
20.	Writing reports to plotter	- Provided - Not provided
21.	Snapshot reports	- Provided - Not provided
22.	Summary reports for multiple runs	- Provided - Not provided

4.2.9. Experimentation Facilities

Criteria classified in this group evaluate the variety and characteristics of experimentation facilities. These facilities are required for improving the quality of simulation results and for speeding up the process of designing experiments and of the experimentation itself. Criteria included in this group and a classification of the packages regarding experimentation facilities are shown in Table 4.9.

Table 4.9 Criteria for experimentation facilities

EXPERIMENTATION FACILITIES		
CRITERIA		**CLASSIFICATION**
1.	Automatic batch run	- Possible - Not possible
2.	Warm-up period	- Provided - Not provided
3.	Re-initialization	- Provided - Not provided
4.	Re-start from non empty state	- Possible - Not possible
5.	Breakpoints	- Provided - Not provided
6.	Speed adjustment	- Provided - Not provided
7.	Experimental design capability	- Provided - Not provided
8.	Quality of experimental design facility	- High - Medium - Low
9.	Accuracy check	- Provided - Not provided
10.	Automatic determination of run length	- Provided - Not provided

4.2.10. Statistical facilities

Due to the randomness that is present in the majority of simulation models, good statistical facilities are very important. Criteria included in this group examine the range and quality of statistical facilities provided by the simulation package. Table 4.10 comprises the criteria included in this group together with a classification of the packages regarding statistical facilities.

Table 4.10 Criteria for statistical facilities

STATISTICAL FACILITIES	
CRITERIA	**CLASSIFICATION**
1. Theoretical statistical distributions	- Provided - Not provided
2. Number of theoretical statistical distributions	- Large - Medium - Small
3. User-defined distributions	- Possible - Not possible
4. Random number streams	- Provided - Not provided
5. Number of different random number streams	- Large - Medium - Small
6. User specified seeds of random number streams	- Provided - Not provided
7. Antithetic sampling	- Provided - Not provided
8. Distribution fitting	- Provided - Not provided
9. Goodness-of fit tests	- Provided - Not provided
10. Output data analysis	- Provided - Not provided

11. Quality of data analysis facility	- High - Medium - Low
12. Confidence intervals	- Provided - Not provided

4.2.11. User Support

The following criteria evaluate the type and quality of user support provided by the software supplier, which can facilitate learning and using the package. These criteria not only include technical support in the form of documentation, demo disks etc. They also include a variety of services provided by the software supplier which ease the use of the package and keep the user informed about plans for future software improvements. Table 4.11 embraces the criteria included in this group and a possible classification of the packages regarding user support.

Table 4.11 Criteria for user support

USER SUPPORT	
CRITERIA	**CLASSIFICATION**
1. Documentation	- Provided - Not provided
2. Quality of documentation	- High - Medium - Low
3. Reference card	- Provided - Nor provided
4. Demo disks	- Provided - Nor provided
5. Tutorial	- Provided - Not provided
6. Training course	- Provided - Not provided

7.	Duration of training courses	- Long - Medium - Short
8.	Frequency of training courses	- High - Medium - Low
9.	Demo models	- Provided - Not provided
10.	Help-line	- Provided - Not provided
11.	User group meetings	- Provided - Not provided
12.	Frequency of user group meetings	- High - Medium - Low
13.	Newsletter	- Provided - Not provided
14.	Package maintenance	- Provided - Not provided
15.	Consultancy	- Provided - Not provided

4.2.12. Financial and Technical Features

Criteria included in this group examine features of the package related to its costs and technical characteristics. Some of the issues considered here are: how expensive it is to purchase a certain package, to install and maintain it, whether any additional hardware would have to be purchased for installation of the package etc. Table 4.12 shows the criteria incorporated in this group and a possible classification of the packages regarding the financial and technical features.

Table 4.12 Criteria for financial and technical features

FINANCIAL AND TECHNICAL FEATURES	
CRITERIA	**CLASSIFICATION**
1. Portability	- Provided - Not provided
2. File conversion	- Possible - Not possible
3. Price	- High - Medium - Low
4. Installation costs	- High - Medium - Low
5. Ease of installation	- Easy - Not easy
6. Hardware requirements	- High - Medium - Low
7. Availability of package on standard hardware	- Provided - Not provided
8. Availability of package on standard operating systems	- Provided - Not provided
9. Version of software for network	- Provided - Not provided
10. Virtual memory facility	- Provided - Not provided
11. Security device	- Needed - Not needed
12. Free software trials	- Provided - Not provided
13. Free technical support	- Provided - Not provided
14. Types of contracts available	- Many - Not many
15. Educational discount	- Provided - Not provided

16. Quantity discount	- Provided - Not provided
17. Life cycle maintenance costs	- High - Medium - Low
18. Price of training course	- High - Medium - Low
19. Consultancy fees	- High - Medium - Low
20. Frequency of update	- High - Medium - Low
21. Comprehensiveness of update	- High - Medium - Low

4.2.13. Pedigree

Criteria shown in Table 4.13 refer to the origin of the package and its prominence. They also evaluate how widely the package is used, and judge the reputation of the software supplier. A supplier's reputation is a general criteria which depends on many factors such as the length of the time the supplier is present in the software market, the number of employees and representative offices the supplier has and the type and level of user support that is provided.

Table 4.13: Criteria for pedigree

PEDIGREE	
CRITERIA	**CLASSIFICATION**
1. Age	- New - Medium - Old
2. Genealogy	----------

3.	Spread	- High - Medium - Low
4.	Success	- High - Medium - Low
5.	Availability of references	- High - Medium - Low
6.	Reputation of supplier	- High - Medium - Low
7.	Sources of information about the package	- Literature - Other users - Supplier - Demonstration - Combination of several sources

4.3. CRITERIA SPECIFIC TO MANUFACTURING SIMULATION PACKAGES

Criteria listed in this section relate to the features specific only to packages dedicated to manufacturing simulation. Criteria within this group are further classified into sub-sections, from the perspective of their nature. Sub-section 4.3.1 comprises criteria related to general software features specific to manufacturing simulation. Typical physical elements to be modelled in manufacturing systems are included in sub-section 4.3.2. Sub-section 4.3.3 contains criteria related to scheduling features, whilst sub-section 4.3.4 comprises criteria related to manufacturing performance.

4.3.1. General Manufacturing Modelling Features

Criteria included in this group concern the general features related to manufacturing modelling. They evaluate whether the package allows modelling of logical elements such as part attributes, shifts modelling, and modelling of machine breakdowns. Some special operations typical for manufacturing systems are also included such as assembling, palletization and fluid composition. Table 4.14 presents the criteria included in this group and a possible classification of the packages concerning general manufacturing modelling features.

Table 4.14 Criteria for general manufacturing modelling features

GENERAL MANUFACTURING MODELLING FEATURES	
CRITERIA	**CLASSIFICATION**
1. Problem areas tackled	- Traditional manufacturing systems - Special types of manufacturing systems
2. Applicability for manufacturing systems	- High - Medium - Low
3. Equipment breakdown modelling	- Possible - Not possible
4. Type of breakdowns	- Clock based - Usage based - Cycle based - Shift based - Combination
5. Machine setup modelling	- Possible - Not possible
6. Machine teardown modelling	- Possible - Not possible
7. Rejects modelling	- Possible - Not possible
8. Capacity of manufacturing equipment	- Provided - Not provided
9. Shifts modelling	- Possible - Not possible
10. Maintenance modelling	- Possible - Not possible
11. Automatic increasing of buffer capacity	- Provided - Not provided
12. Buffer delays	- Provided - Not provided
13. Job lists	- Provided - Not provided

14.	Part attributes modelling	- Possible - Not possible
15.	Frequency of part arrival modelling	- Possible - Not possible
16.	Arrival of parts in batches	- Provided - Not provided
17.	Type of part arrival	- Generation - Creation
18.	Variable conveyor speed	- Provided - Not provided
19.	Assembly operation modelling	- Possible - Not possible
20.	Disassembly operation modelling	- Possible - Not possible
21.	Containerization modelling	- Possible - Not possible
22.	Fixturing modelling	- Possible - Not possible
23.	Palletization modelling	- Possible - Not possible
24.	Evaporation of fluids modelling	- Possible - Not possible
25.	Precipitation of fluids modelling	- Possible - Not possible
26.	Fluid composition modelling	- Possible - Not possible
27.	Inspection operation modelling	- Possible - Not possible

4.3.2. Physical elements

The following criteria examine which physical elements typical to different types of manufacturing systems can be modelled by a particular package. These criteria presented in Table 4.15 mainly relate to different types of machines and means of transport that can be modelled by a specific package.

Table 4.15 Criteria for physical elements

PHYSICAL ELEMENTS		
	CRITERIA	**CLASSIFICATION**
1.	Single machines	- Provided - Not provided
2.	Batch machines	- Provided - Not provided
3.	Production machines	- Provided - Not provided
4.	Assembly machines	- Provided - Not provided
5.	Multi-cycle machines	- Provided - Not provided
6.	Multi-station machines	- Provided - Not provided
7.	Buffers	- Provided - Not provided
8.	Workstation buffers	- Provided - Not provided
9.	Labour	- Provided - Not provided
10.	Automated guided vehicles (AGVs) and trucks	- Provided - Not provided
11.	Conveyors	- Provided - Not provided
12.	Types of conveyors	- Queuing conveyors - Conveyors with carriers - Both
13.	Branching and looping of conveyors	- Possible - Not possible
14.	Conveyor buffers	- Provided - Not provided
15.	Fork-lifts	- Provided - Not provided
16.	Robots	- Provided - Not provided

17. Automated storage retrieval system	- Provided - Not provided
18. Tools	- Provided - Not provided
19. Automated tool storage	- Provided - Not provided
20. Pallets	- Provided - Not provided
21. Fixtures	- Provided - Not provided
22. Fixture stores	- Provided - Not provided
23. Pallet shuttles	- Provided - Not provided
24. Carousel-type magazines	- Provided - Not provided
25. Cranes	- Provided - Not provided
26. Tanks and fluids	- Provided - Not provided

4.3.3. Scheduling Features

Criteria embraced in this group investigate the variety of scheduling strategies that can be modelled by the package. These criteria are dominated by a variety of features needed for part and vehicles scheduling. Table 4.16 shows the criteria included in this group and a classification of the packages regarding scheduling features.

Table 4.16 Criteria for scheduling features

SCHEDULING FEATURES	
CRITERIA	**CLASSIFICATION**
1. Scheduling rules	- Provided - Not provided

2.	Number of scheduling rules provided	- Large - Medium - Small
3.	Remaining processing time calculation	- Provided - Not provided
4.	Conditional routing	- Possible - Not possible
5.	Priority	- Provided - Not provided
6.	Preemption	- Provided - Not provided
7.	Push/pull from specific positions within the element	- Possible - Not possible
8.	Specification of quantity of parts to be moved between elements	- Possible - Not possible
9.	Batch index	- Provided - Not provided
10.	Predefined part routing	- Provided - Not provided
11.	Routing restrictions	- Exist - Do not exist
12.	Type of part sequencing	- Probabilistic - Conditional - Deterministic - Combination
13.	Departure scheduling for shipping area	- Provided - Not provided
14.	Vehicle scheduling	- Provided - Not provided
15.	Vehicle acceleration and deceleration	- Provided - Not provided
16.	Scheduling optimization	- Provided - Not provided

4.3.4. Manufacturing Performance

Whilst criteria listed in section 4.2.8 examine the type and quality of general output reports, criteria included in this section relate to reports typical for manufacturing. Criteria presented in Table 4.17 provide the standard reports needed for an insight into the performance of the manufacturing system being modelled.

Table 4.17 Criteria for manufacturing performance

MANUFACTURING PERFORMANCE	
CRITERIA	**CLASSIFICATION**
1. Throughput	- Provided - Not provided
2. Work in progress	- Provided - Not provided
3. Utilization of production equipment	- Provided - Not provided
4. Makespan	- Provided - Not provided
5. Special user-defined reports	- Provided - Not provided
6. Due dates monitoring	- Provided - Not provided
7. Manufacturing costs analysis	- Provided - Not provided
8. Schedule related report	- Provided - Not provided
9. Transportation time of the parts	- Provided - Not provided
10. Rework and scrap level	- Provided - Not provided
11. Interruption reports	- Provided - Not provided
12. Production sequence summary	- Provided - Not provided

4.4. USE OF CRITERIA

Whilst sections 4.2. and 4.3 establish an evaluation framework for manufacturing simulation packages, this section provides a discussion on the use of this framework. Information originated here is primarily based on work with students both at undergraduate and postgraduate level and practical experience gained by work for industry. Findings were also derived from many conversations with simulation software users at conferences, seminars and user group meetings.

A distinction is made between users of software for educational purposes and users in industry. The main reasons for this are differences in the purpose of modelling, complexity of the systems being modelled, and most likely the experience of the users.

It is not possible to draw an absolute line between these two user groups, because there are many variations in real dynamic manufacturing environments. However, some general guidelines regarding the use of criteria for software selection are provided in this section, whilst Chapter 6 provides more detailed discussion on this issue.

Sub-section 4.4.1 addresses the hierarchy of criteria regarding their importance to particular groups of users. Use of the evaluation framework for the selection of a package for education is discussed in sub-section 4.4.2, whilst its use for the selection of a package for industry is discussed in sub-section 4.4.3.

4.4.1. Hierarchy of Criteria

An important issue to be considered concerning the use of the established criteria is the differing importance of specific criteria to different types of users. From the point of view of this research, a hierarchy of criteria applicable for the selection of a package for education will differ from a hierarchy that concerns the selection of a package to be used in industry.

Although some of the criteria might be considered to be of equivalent importance, such as criteria regarding pedigree or efficiency of the software, there are also many dissimilarities concerning the relevance of criteria for each group of users. These issues are addressed in the next two sub-sections.

Tables presented in these sub-sections rank the groups of criteria according to their importance for specific purposes of software use. For each group, a subset of the group's individual criteria are listed which are believed to be the most relevant. The level of importance for each group of criteria is established in the range from 1 to 5, where level 1 represents a very important group of criteria and level 5 represents an irrelevant group of criteria for certain type of users. The proposed levels of importance are qualitative and relative, based on the personal experience and judgement. As such, they cannot be quantitatively justified.

4.4.2. Selection of a Package for Education

It was assumed that the users in this group would use simulation packages mainly for educational purposes in manufacturing simulation, and that the main users are students with little or no previous experience in simulation modelling. The use of software at educational institutions for modelling complex real life manufacturing problems was excluded. For that purpose, as well as for the research, a hierarchy of criteria proposed for users in industry can be applied. Table 4.18 summarizes a proposed hierarchy of criteria regarding selection of a package for education.

Table 4.18 Hierarchy of criteria for selection of a package for education

EDUCATION		
GROUP OF CRITERIA	**CRITERIA**	**LEVEL OF IMPOR-TANCE**
Modelling assistance	- On-line help - Prompting - Quality of prompting - Logic checks	1
General features	- Data driven simulator with or without additional programming - Modelling transparency - User friendliness - Ease of learning - Ease of using	1
Visual aspects	- Animation - Icon editor	1

Efficiency	- Interaction - Reliability - Time scale for model building - Specification of part flow by a mouse	1
Testability	- Error messages - Quality of error messages - Step function - Display of events on the screen - List files	1
Input/Output	- Menu driven interfaces - Dialogue boxes - Understandability of output reports - Static graphical output	1
Physical elements	- Single machines - Production machines - Assembly machines - Buffers - Labour - AGVs and trucks - Conveyors	1
User support	- Quality of documentation - Tutorial - Demo models - Package maintenance	1
Manufacturing performance	- Throughput - Work in progress - Utilization of production equipment	1
Financial and technical features	- Educational discount - No security device - Hardware requirements - Version of software for network	1
Scheduling features	- Conditional routing - Priority - Scheduling rules	2
Coding aspects	- Program generator - Readability of added code - Comprehensiveness of added code	2
General manufacturing modelling features	- Capacities - Part attributes modelling - Shifts modelling	2

Statistical facilities	- Theoretical statistical distributions - Random numbers	2
Experimentation facilities	- Warm up periods - Automatic batch run - Re-start from non empty state	2
Software compatibility	- Integration with word processors - Integration with spreadsheet packages	3
Pedigree	- Availability of references - Spread	3

The proposed hierarchy of criteria for the selection of packages for education in the above table, favours criteria regarding ease of using and learning the package, modelling assistance provided by the packages, efficiency, the incorporation of physical elements typical for manufacturing systems, and visual aspects.

All criteria that support learning and relatively quick and easy model development have higher importance than those that enable the handling of a large quantity of data (eg. software compatibility), and detailed modelling (eg. coding aspects). The main reason for this is the relatively short duration of simulation courses in many cases (Simulation Study Group, 1991). Therefore, it is believed that students should not spend too much time on model building. They should also learn the basics of simulation methodology, such as the use of statistics in simulation, conceptual model development, model validation and verification techniques, design of experiments, and analysis of simulation output.

4.4.3. Selection of a Package for Industry

The selection of a package for use in industry is divided into two groups according to the purpose of modelling. The first group presents a hierarchy of criteria that might be applied for the selection of a package to be used for 'quick and dirty' modelling, whilst the second group establishes a hierarchy of criteria for selection of a package for the detailed/complex modelling of complex real life manufacturing problems.

(i) 'Quick and dirty' modelling

For this type of modelling it is assumed that users have some previous experience in simulation modelling and that they know the basic methodological issues. 'Quick and dirty' modelling of real manufacturing systems means that models should be developed as quickly as possible without too many details in order to provide basic information about the system being modelled. Table 4.19 presents a proposed hierarchy of criteria that may be used for the selection of a package for 'quick and dirty' modelling in industry.

Table 4.19 Hierarchy of criteria for selection of a package in industry for 'quick and dirty' modelling

INDUSTRY - 'QUICK AND DIRTY' MODELLING		
GROUP OF CRITERIA	**CRITERIA**	**LEVEL OF IMPORTANCE**
General features	- Data driven simulator with or without additional programming - Ease of using - Modelling transparency	1
Modelling assistance	- Logic checks - Prompting - On-line help - Automatic editing of data - Use of mouse	1
Efficiency	- Interaction - Time scale for model building - Specification of part flow by a mouse - Automatic model building - Model reusability - Compilation time - Execution time - Reliability	1
Visual aspects	- Animation - Icon library	1

Input/Output	- Menu driven interface - User defined output - Dynamic graphical output	1
Physical elements	- Single machines - Batch machines - Assembly machines - Production machines - Labour - AGVs and trucks - Conveyors	1
General manufacturing modelling features	- Applicability for manufacturing systems - Capacities - Part attributes - Arrival of parts in batches	1
Scheduling features	- Number of scheduling rules provided - Conditional routing - Priority	1
Manufacturing performance	- Throughput - Work in progress - Utilization of production equipment - Makespan - Production sequence summary	1
Testability	- Error messages - Step function - Explode function - Display of events on the screen	1
Financial and technical features	- Price - Hardware requirements - Portability - Life cycle maintenance costs	1
User support	- Documentation - Tutorial - Package maintenance	2
Coding aspects	- Program generator - Built-in functions	2
Experimentation facilities	- Automatic batch run - Speed adjustment - Experimental design capability	2
Statistical facilities	- User defined distributions - Theoretical distributions - Output data analysis	2

Software compatibility	- Integration with spreadsheet packages - Integration with statistical packages	3
Pedigree	- Reputation of supplier - Spread - Success	3

The hierarchy of criteria for the selection of the packages for 'quick and dirty' modelling in industry shows that the most important are criteria regarding modelling assistance for easy development of models, efficiency in the time needed for model development, and standard physical elements and measures of performance typical of manufacturing systems. These criteria are more important than those that facilitate detailed and complicated modelling with a large quantity of data such as criteria concerning coding aspects, software compatibility, and extensive statistical and experimental facilities.

(ii) Detailed/Complex modelling

When a simulation study is carried out to develop a detailed model of complex manufacturing systems, then the most important criteria are those symbolizing the power of the package regarding its robustness, modelling flexibility and efficiency. It is assumed that the users of the software in this group have experience in simulation modelling and a certain level of theoretical knowledge about simulation. Table 4.20 displays a hierarchy of criteria applicable for the selection of a package for detailed/complex modelling in industry.

Table 4.20 Hierarchy of criteria for selection of a package in industry for detailed/complex modelling

INDUSTRY - DETAILED/COMPLEX MODELLING		
GROUP OF CRITERIA	**CRITERIA**	**LEVEL OF IMPORTANCE**
Coding aspects	- Programming flexibility - Link to a lower language - Data storage, retrieval and manipulation facilities - Support of programming concepts - Text/code manipulation - Comprehensiveness of added code	1
Software compatibility	- Integration with DBMS - Integration with CAD software - Integration with expert systems - Integration with statistical packages	1
Efficiency	- Robustness - Level of detail - Number of elements in the model - Reliability - Model reusability - Model chaining - Compilation time - Execution time - Merging of models	1
Visual aspects	- Animation - Virtual screen - Multiple screen layout - Icon editor - Icon library - Switching on/off graphic - No limitation on number of displayed icons	1
Testability	- Quality of error messages - Trace files - Step function - Display of variables - Display of functions - List files - Display of events on the screen	1

Physical elements	- Multi-station machines - Production machines - Assembly machines - Batch machines - Multi-cycle machines - Single machines - Buffers - Labour - AGVs and trucks - Conveyors	1
Scheduling features	- Number of scheduling rules provided - Conditional routing - Predefined part routing - Priority - Preemption - Vehicle scheduling - Scheduling optimization	1
Input/Output	- Special user defined reports - Input data reading from files - Writing reports to files - Multiple inputs - Multiple outputs - Summary reports for multiple runs - Dynamic graphical output	1
Experimentation facilities	- Automatic batch run - Experimental design capability - Accuracy check	1
Modelling assistance	- Logic check - Quality of prompting - Ease of verification - Model and data separation - On-line help - Automatic editing of data - Documentation notes	1
Statistical facilities	- Number of theoretical statistical distributions - User defined distributions - Number of different random number streams - Output data analysis - Confidence intervals	1

General features	- Data driven simulators with additional programming or simulation language - Hierarchical model building - Modelling transparency	1
Manufacturing performance	- Special user defined reports - Schedule related output - Production sequence summary	1
Financial and technical features	- Price - Hardware requirements - Life cycle maintenance costs - Consultancy fees	1
General manufacturing modelling features	- Applicability for manufacturing systems - Part attributes modelling - Equipment breakdowns modelling - Machine setup modelling - Inspection operation modelling - Arrival of parts in batches	2
User support	- Quality of documentation - Training course - Help line - Consultancy - Package maintenance	2
Pedigree	- Success - Spread - Reputation of supplier	2

The above hierarchy of criteria for the selection of a package for detailed modelling in industry has the largest number of criteria that have the highest level of importance in comparison to the other two hierarchies.

The most relevant criteria are those regarding the flexibility of the package supported by coding aspects, the possibility of integration with data base management systems to handle a large quantity of data, efficiency that can speed up such detailed and complex modelling and testability which can ease the time consuming process of model verification.

For this type of modelling it is also important for the package to provide good support in experimentation and in statistical facilities, and the possibility of obtaining special user defined reports. Issues related to scheduling are also important as well the possibility of quickly modelling a variety of physical elements and operations in a manufacturing system.

4.5. SUMMARY

This chapter presents a number of criteria that can be used in the evaluation of software for manufacturing simulation. Although many of the criteria listed can apply to any type of simulation package, they were presented and explained from the perspective of packages for manufacturing simulation. A description of all the criteria is given in Appendix I.

The majority of these criteria were derived from practical experience obtained in using different manufacturing simulators for a case study. The literature also provided evidence of the need for many of the features presented in this chapter, as well as the findings from a survey described in Chapter 6.

After several years of work in the field of simulation modelling, it was realized that there are different requirements for simulation software from different types of users. From this point of view, a possible use of the established evaluation framework has been derived separately for users in education and users in industry.

Different hierarchies of criteria have been proposed expressing the relevance of certain criteria according to the software purpose: education, 'quick and dirty' modelling in industry and detailed/complex modelling in industry. With the proposed 5 levels of importance, none of the groups of criteria for any software purpose appeared to be irrelevant (level of importance 5). The number of groups of criteria that gained the highest level of importance is the greatest for software to be used for detailed modelling in industry.

Hierarchies of criteria established for education and for 'quick and dirty' modelling in industry are somewhat similar because they both favour features of the software that can ease and accelerate model development. On the other hand, the hierarchy of criteria for detailed modelling in industry is substantially different from the first two, supporting features related to flexibility and efficiency of the software. Table 4.21 summarizes the proposed levels of importance of groups of criteria for different software purposes.

Table 4.21 Levels of importance of groups of criteria for different software purposes

GROUP OF CRITERIA	EDUCATION	'QUICK AND DIRTY' MODELLING IN INDUSTRY	DETAILED/ COMPLEX MODELLING IN INDUSTRY
General Features	1	1	1
Visual Aspects	1	1	1
Coding Aspects	2	2	1
Efficiency	1	1	1
Modelling Assistance	1	1	1
Testability	1	1	1
Software Compatibility	3	3	1
Input/Output	1	1	1
Experimentation Facilities	2	2	1
Statistical Facilities	2	2	1
User Support	1	2	2
Financial and Technical Features	1	1	1
Pedigree	3	3	2
General Manufacturing Modelling Features	2	1	2
Physical Elements	1	1	1
Scheduling Features	2	1	1
Manufacturing Performance	1	1	1

It would not be realistic to expect a particular package to satisfy all criteria listed in this chapter. However, which criteria are more important is indicated, according to the software purpose.

This chapter provides a more comprehensive overview of the desirable characteristics of simulation packages, and especially of manufacturing simulators, than those reviews found in the literature. Thus, these guidelines can be used both by users who are looking for a suitable simulator to buy, and by developers of such simulators to improve existing versions of simulators or perhaps to try to develop a new, better manufacturing simulator.

CHAPTER 5. EVALUATION OF MANUFACTURING SIMULATORS

5.1. INTRODUCTION

This chapter deals with the evaluation of several manufacturing simulators, as applied to the development of the models in the case study. These simulators are considered as typical representatives of different types of widely used manufacturing simulators.

The evaluation is not performed in order to discover which is 'the best' simulator, because such a term most likely does not exist in the context of simulation software. The main reason for this is a constant updating of existing software and the release of new software products. Additional factors are the intended software purpose and the personal preferences of simulation software users (Pidd, 1992a). Due to these facts, the evaluation presented in this chapter was primarily performed to demonstrate a possible use of the evaluation framework derived in Chapter 4, and a feasible determination of the suitability of certain types of simulators for particular purposes.

Evaluation has been based mostly on the development of models for the case study, presented in Chapter 3. During this evaluation, the evaluation framework of Chapter 4 has been used in order to see which characteristics a certain simulator possesses, and to what extent it can satisfy the needs of specific types of users. Some information was also derived from an analysis of the literature. However, most of the claims found in the publications were critically reviewed, due to the possible bias present in some papers (previously mentioned in Chapter 2).

Evaluation has been performed mainly from the perspective of the groups of criteria established in Chapter 4, rather then to examine every single criterion for each package. Such an approach has been taken in this research because it is believed that it is better to describe general features of a package, and its possible usability for specific purposes, then to evaluate each simulator in too much detail. In any event simulators are under constant revision which probably makes any evaluation obsolete quite quickly.

The evaluation of manufacturing simulator WITNESS is described in section 5.2. SIMFACTORY II.5 is evaluated in section 5.3. Section 5.4 provides the evaluation XCELL+, whilst ProModelPC is evaluated in section 5.5.

A comparison of the evaluated simulators is made in section 5.6, with an emphasis on their usability either for education or for modelling in industry. A summary of this chapter is presented in section 5.7.

5.2. Evaluation of WITNESS

This section provides a critical evaluation of WITNESS. Positive features of this simulator are addressed in sub-section 5.2.1, whilst its negative features are presented in sub-section 5.2.2.

5.2.1. Positive Features

Several *general features* of this simulator make it adequate for the simulation of manufacturing systems. WITNESS is a data driven, manufacturing oriented simulator, with the facility to add some program code. Its Windows based environment with pull-down menus makes it very user friendly and it is easy to use once it is learnt. Modelling transparency is good.

The *Visual aspects* are quite good, with easy to use icon and screen editors that can produce nice graphical displays of the models, using multiple colours. Icons can be stored in the icon library. These icons can also be manipulated. Full animation is provided, with the movement of elements proportional to the time needed for a change in their state. Panning and zoom function are provided. Graphics can be switched on or off. Icons can be changed during the simulation, when a change of element represented by a particular icon occurs.

With respect to *coding aspects,* WITNESS provides an internal language which enhances modelling flexibility. The user can write code to handle special logical features. The syntax of the code is fairly readable and precise. A number of built-in functions are provided. The user can also write his/her own functions, which can invoke built-in functions. Global variables accessible by all elements in the model can be used.

The *efficiency* of this simulator is mainly expressed by its robustness, achieved by programming flexibility. In addition, it possesses a high level of interactivity and adaptability. Models can be changed at any time, and the status of elements can be inspected. WITNESS enables a model to be saved with its current status, and it is case insensitive. There is no limit to model size apart from hardware limits. Partially developed models can be retrieved and edited.

Modelling assistance is provided by several features. Prompting is provided, but it is biased towards experienced users because it mainly points at what should not be done. Code entered via the text editor is automatically formatted, and the software imposes its own use of upper and lower case letters. An easily accessible on-line help is provided, but the information it gives is somewhat general.

Several useful features that facilitate *testability* are provided. Error messages are supplied. It is possible to obtain a graphical display of the values of functions and variables in addition to animation. When experiments are run in the step mode, every change in model status that happens is written in the interact box. It is also possible to obtain trace files, with all the model changes that occurred during the simulation. The Explode function provides information about the status of model elements, listing all attributes of the parts positioned at these elements. Illegal inputs are rejected, with an appropriate message.

Software compatibility enables integration with spreadsheet packages for output data analysis, and integration with word processors to edit model list files, create input data files or create programs using the WITNESS Command Language.

With regard to the *Input/Output* group of criteria, a variety of reports are automatically provided as well as special user defined reports. Periodic reports written to a file can be also obtained. Dynamic graphical display of histograms and time series is also provided. Data can be entered into the model via a menu driven interface, or they can be read directly from the files.

Experimentation facilities provide automatic batch running of experiments. Speed adjustment is possible as well as the specification of a warm-up period for experimentation. Models can be re-started from a non-empty state.

The quality of *statistical facilities* is good in the sense that a variety of theoretical statistical distributions are provided as well as 100 different random number streams. User defined distributions can be specified. It is possible to perform antithetic sampling.

A high level of *user support* is provided by the supplier. A help-line is available to users, training courses are organized, and user group meetings are held regularly. Documentation and reference cards are supplied, but the quality of documentation could be improved.

Positive aspects of *financial and technical features* are software portability, its availability for standard hardware and for standard operating systems, educational discounts given to universities, and relatively frequent updating of the software.

With regard to the *pedigree* of WITNESS, it is claimed that it is widely used, especially in industry. It was introduced in 1986. References describing characteristics of this simulator and its successful use in simulation projects are available. It was developed from the general purpose language SEE WHY.

Many *general manufacturing modelling features* are supplied such as part attributes modelling, shift modelling, capacities, breakdowns modelling, machine setup modelling, rejects modelling and job lists. Parts can arrive in the model in batches. In addition, it is possible to model buffer delays and a variety of operations such as assembling, disassembling, inspection and fluid composition.

Typical *physical elements* existing in manufacturing systems are pre-defined and incorporated in the simulator. Different types of machines can be explicitly modelled such as single, batch, production, assembly, multi-cycle and multi-station machines. Buffers, labour, conveyors, trucks and vehicles, and continuous processing elements such as tanks and fluids are also provided.

Scheduling features are mostly supported by the programming flexibility of WITNESS. Conditional routing is possible, and a variety of input and output rules are available. Various scheduling strategies can be modelled by programming with the support of input/output rules. Different priorities can be specified for different elements and the preemption of labour can be performed. Vehicle scheduling can also be modelled.

A variety of reports regarding *manufacturing performance* can be obtained such as information on throughput, work in progress, the utilization of production equipment and the scrap level of the parts. In addition, special user-defined reports can be created.

5.2.2. Negative Features

The main shortcoming of WITNESS regarding its *general features* are that, because of its comprehensiveness, it is not easy to learn so that its full potential may be realised, and its special logical features modelled. In addition, it is not possible to create run-time applications.

With regard to *visual aspects*, the icon library supplied is quite small and the icons are too simple. The graphical display of the models is overwritten by windows representing, for example, an interaction box. It is not possible to obtain three-dimensional graphical displays of models.

The main weaknesses of the *coding aspects* are the limited flexibility of the language provided for additional coding, and the restrictions on its use. For example, it is not possible to program actions when a part arrives at a machine. This is possible only when the machine starts operating. Another shortcoming relates to the text editor provided for coding. The maximum length of lines in the editor is 256 characters, which may cause problems when complex features are modelled. In addition, there is no indication of the cursor position within the line, so it is not possible to know when the limit of 256 characters has been reached. Going over the limit is reported only when the code is to be saved. Saving is then not possible, nor is it possible to determine which parts of the lines are surplus.

Efficiency is restricted by the problems with reliability. Namely, the program might get stuck for no apparent reason, and then the computer has to be restarted. Multitasking and model chaining are not provided. There is no automatic saving of models nor the possibility to exit to the operating system within the software. Merging of models is not possible, which is especially inconvenient when large complex models are developed.

Weakness of the *modelling assistance* lie in the limited usefulness of prompting and on-line help, which is to general.

Testability is generally good, but it might be useful if the quality of error messages is improved, because they do not provide advice on how the detected error can be corrected. In addition, a backward clock is not provided and it is not possible to view the workflow path of the parts.

With regard to *software compatibility*, at the moment it is not possible to integrate WITNESS with CAD systems, statistical packages, data base management systems, expert systems, MRP II software and scheduling software.

The shortcomings of the *input/output* features relate to a lack of static graphical displays of simulation results. In addition, there is no automatic rescaling of the y axis in dynamic graphical displays of time series and histograms, and the standard output report written to a file is lengthy and not comprehendible. It is not possible to obtain a summary report of multiple independent experiments.

The main weakness of the *experimentation facilities* is the absence of an experimental design capability and no facility to interrupt experiments run automatically. Setting up an automatic run of experiments is not straightforward.

The main limitation of the *statistical facilities* is the lack of an output data analysis facility. There is a fixed number of random number streams, and the user cannot specify stream seeds. Confidence intervals cannot be obtained, and a facility for distribution fitting is not provided.

The main shortcomings of *user support* relate to the lack of an interactive tutorial which can facilitate learning of the package, and the quality of documentation. Documentation should provide more useful examples of the functions, actions and input/output rules and it should include an explanation of error messages.

With regard to *financial and technical features*, the main obstacle is the high price of the package, and substantial hardware requirements (it requires a minimum of 4MB of memory to operate, and a recommended 8MB of hard disk to install). In addition, a security device is obligatory, which is not very convenient, especially if the software is used for education.

Considering the *general manufacturing modelling features*, it is apparent that an automatic increasing of the buffer capacity is not provided. The explicit modelling of some specific operations such as fixturing and palletization is not straightforward, whilst fluid modelling is quite basic.

Although the major *physical elements* typical for manufacturing systems are provided, some special ones are missing such as pallets with fixtures, pallet shuttles, containers, robots and cranes. Some of those elements that are provided, such as vehicles, are not easy and straightforward to use.

The main limitations of the *scheduling features* are an inability to push/pull a part from specific positions within the element, to push/pull from the element more than one part, and routing restrictions. For example, buffers are passive, which means that they can neither push not pull parts. In addition, there is no departure scheduling for the shipping area, and there is no explicit way of using the batch index. Automatic calculation of optimal scheduling is not provided.

A variety of measures of *manufacturing performance* are provided by the software, or could be obtained with additional programming. Nevertheless, there is no schedule related report such as a Gantt chart, and it is not possible to obtain a production sequence summary report.

5.3. Evaluation of SIMFACTORY II.5

A critical evaluation of SIMFACTORY II.5 is presented in this section. Positive features are analyzed in sub-section 5.3.1, whilst sub-section 5.3.2. addresses negative features of this simulator.

5.3.1. Positive Features

There are several criteria regarding *general features* that make SIMFACTORY II.5 suitable for the manufacturing simulation. This data driven manufacturing simulator includes manufacturing terminology and it is particularly user friendly. It can be quite easily learnt and used, once the basic concepts are understood. Modelling transparency is good. The user can specify time units, length measures and names of entities. It is possible to perform model initialization.

With regard to the *visual aspects*, an icon editor is provided with various facilities for design of icons with multiple colours and manipulation with them. The library of standard icons is provided, which should be good enough for many models. Animation is full with a visual clock. The virtual screen is provided as well as the facility for print screen. Different colours are used to represent the state of elements. Animation can be switched on/off. It is very easy to copy icons, once they are positioned on the screen.

Although a certain level of programming flexibility is supported, *coding aspects* are not a distinctive feature of SIMFACTORY II.5. The Expression Builder which enables the user to write mathematical expressions or simple code fragments to extend the model's logic is provided. It is easy to use, because the user has to choose from available options and select them by mouse. Added code is readable especially to users familiar with the SIMSCRIPT II.5 language.

The *efficiency* is reflected by the criteria such as interaction and adaptability. For example, a special Dynamic Interaction forms are provided where the user can change characteristics of the elements and perceive the effect of these changes to the rest of current simulation run. SIMFACTORY II.5 also enables model status saving and model chaining. Several queuing policies (inventory sorting rules) are provided. It is possible to edit partially developed models, and modify completed models quite easily.

Modelling assistance is revealed by an easily accessible on-line help, the use of mouse and model and data separation. When the Expression Builder is used for coding, the appropriate number of ENDIF commands is added automatically. Text files with an information about the model elements, the workflow, arrival of parts and setup of experiments can be edited separately and imported directly to the model.

Several features facilitate *testability* of this simulator. Error messages are provided when the compilation starts. Display of element's state is given as well as a dynamic display of capacity. Workflow path can be also displayed. Echo and trace files can be obtained. Models can be run using the step function. Rejection of illegal inputs is provided.

Software compatibility is represented by a possibility to import text files created by the word processors or data base packages. In addition, simulation results can be exported to a spreadsheet package or data base for further analysis.

Quality of *input/output* aspects is quite good. A menu driven interface with dialogue boxes is provided. Forms for the data input contain the selection buttons, which are to be selected by mouse. Input data can be read from the text files. A variety of output reports are automatically obtained at the end of simulation experiment. These reports are quite understandable and some of them are presented in the graphical form (pie charts). Snapshot reports are provided during the experimentation. A summary report for multiple runs of independent experiments can be obtained.

Experimentation facilities are also satisfying. Automatic batch running of multiple experiments is provided. Even different models can be run in one batch. Speed adjustment is possible as well as to specify breakpoints for experiments. Perhaps the most distinctive feature are the accuracy check, which reports a level of accuracy of the results, and an automatic determination of run length.

Statistical facilities significantly contribute to the positive features of this simulator. Several theoretical statistical distributions are provided as well as user defined distributions. Although only 10 different random number streams are provided initially, the user can create additional streams by specifying the seeds of random number streams. Antithetic sampling is possible. Output data analysis is provided in the form of mean value, variance and confidence interval for the results obtained in multiple runs. The Datagraph facility, incorporated within SIMFACTORY II.5, enables distribution fitting and performing the Goodness-of-fit tests.

User support is reflected by a training course, user group meetings, package maintenance and documentation. Documentation is very readable.

Positive features of *financial and technical features* are the availability of this simulator for standard hardware and operating systems, portability, ease of installation and educational discounts provided. The supplier provides users with free training and a trial system. In addition, SIMFACTORY II.5 is quite frequently updated, and it does not demand a security device.

With regard to the *pedigree* of this software, it is stated that it is widely used both in industry and educational institutions. It was introduced in 1986. SIMFACTORY II.5 is written in the SIMSCRIPT II.5 simulation language.

The suitability of this simulator for modelling manufacturing systems is reflected through several *general manufacturing modelling features*. It enables modelling of part attributes, breakdowns, rejects, machine setup and teardown, and capacity of manufacturing equipment. It is possible to model arrival of parts in batches. In addition, automatic increasing of buffer capacity is provided as well as shifts modelling. Assembling, disassembling and inspection operations can be easily modelled.

This simulator enables explicit modelling of *physical elements* typical for manufacturing systems. These elements include normal stations (single machines), chamber and batch stations (batch machines), queues (buffers), receivers, conveyors, transporters (AGVs) and labour.

Scheduling features are mainly supported by the in-built scheduling and inventory rules. There are eighteen push and pull rules provided for stations and six inventory sorting rules which specify the order in which products are stored in buffers. Elements within the model can have different priorities which determines the flow of the parts.

Conditional routing is possible by adding the fragments of code. Vehicle scheduling can be also modelled.

A variety of measures of *manufacturing performance* are provided by SIMFACTORY II.5. Information about throughput, work in progress, utilization of equipment and makespan are automatically provided. A distinctive feature of this simulator is the in-built extensive cost analysis facility. Interruption reports are also provided.

5.3.2. Negative Features

The main limitation regarding the *general features* of SIMFACTORY II.5 is a lack of possibility for creation of run-time applications.

With regard to the *visual aspects*, there is no screen editor for the enhancement of models' graphical display. It is only possible to add the text on the screen in addition to icons. It is not possible to change the colour of the elements' status display. There is no possibility to obtain 3-dimensional model display.

There are several shortcomings regarding the *coding aspects*. Although the fragments of code can be added via the Expression Builder, the usability and flexibility of this code is quite limited. User can only select provided commands such as IF, THEN, ELSE and LET. There is no access to source code or link to a lower level language. Added code cannot be manipulated with, and there is no support for the main programming concepts.

Efficiency is restrained by a limited robustness and level of detail that can be modelled by SIMFACTORY II.5. Multitasking is not possible and problems with reliability might occur for no apparent reason. It is not possible to access the operating system within the package. Automatic model building is not provided.

Modelling assistance is lacking prompting which is especially needed when the Expression Builder is used. There is no a text editor as integral part of the simulator, which might be used for editing reports or list files. In addition, there is no possibility for writing documentation notes as the model is developed.

The shortcomings of *testability* are reflected in inadequate error messages which do not provide advice on how to correct the mistake. Display of attributes and variables is not provided as well as a backward clock.

With respect to *software compatibility*, it is not possible to integrate SIMFACTORY II.5 with CAD software, expert systems, MRP II software and scheduling software.

The main limitations of the *input/output* features are a lack of dynamic graphical outputs and user defined outputs.

Although the *experimentation facilities* are generally very good, there is no assistance in an experimental design.

The main shortcoming of the *statistical facilities* is a small number of random number streams provided, so the user has to create additional ones by specifying the seeds of random number streams.

With respect to *user support*, the quality of documentation might be better because it does not provide examples on how more complex logic can be modelled and there are some mistakes regarding the index of terms (some of the terms are not explained on the pages specified). There is no an interactive tutorial for learning of this simulator. Although some help might be obtained from the supplier, there is no official help-line especially established for the users of SIMFACTORY II.5.

Hardware requirements (hard disk with 4MB free space, 4MB of RAM, VGA graphic card) and price of the software might be considered as the shortcoming regarding the *financial and technical features*. In addition, guidelines for an installation described in the user manual do not correspond to the actual process of installation, which does not make the installation straightforward.

With regard to the *general manufacturing modelling features*, it is not possible to model buffer delays, to obtain job list, or to model some specific operations such as fixturing, palletization, fluid composition, evaporation and precipitation.

Though the major *physical elements* typical for manufacturing systems are supplied, some special ones are not included in the SIMFACTORY II.5 such as pallets with fixtures, pallet shuttles, robots, cranes, containers, tanks and fluids.

The shortcomings of the *scheduling features* are reflected in the following features. Conveyors are passive, they cannot use push and pull rules for conditional routing. It is not possible to push/pull part from specific positions within the element, nor to push/pull from the element more than one part. In addition, there is no possibility to use the batch index, it is not possible to use predefined part routing, and there is no departure scheduling for shipping area.

Although many standard measures of *manufacturing performance* are provided, the user can not request special reports. There is no schedule related report such as Gantt chart, there is no explicit way of due dates monitoring, and it is not possible to obtain a summary of production sequence for all part types.

5.4. Evaluation of XCELL+

This section provides a critical evaluation of XCELL+. Positive and negative features of this simulators are analyzed in sub-sections 5.4.1 and 5.4.2 respectively.

5.4.1. Positive Features

The most distinctive *general features* of this data-driven simulator that can be regarded as its advantages are the ease of learning and use. XCELL+ is manufacturing oriented and it incorporates manufacturing terminology. This simulator is primarily designed for non-simulation professionals, and therefore it does not require a substantial level of user's experience or formal education in simulation. Generally, it is user friendly once the basic concepts have been captured through the documentation.

With respect to the *visual aspects*, concurrent animation is provided as well as the virtual screen. The display of the elements' status is supplied. The scale of a graphical display of models can be changed using the zoom function. It is possible to obtain the print of screens.

The *efficiency* of this simulator is primarily expressed by the small time scale for model building. In addition, a high level of interactivity and adaptability is provided and model execution time is relatively short, mainly because the models do not contain a high level of detail. When the model is stored, both the structure and the state of the model is saved. It is possible to perform merging of different models, and to retrieve and edit partially developed models.

Modelling assistance is provided in the form of an easily accessible on-line help, prompting and error messages. The XCELL+ prompts the user when the inputs are required, whilst rejecting illegal specifications such as specifying the wrong type of cell for an operation or entering input values in the wrong format.

Testability is represented by several features. A structural (logic) check is provided, which examines whether Workcentres have Processes; whether Buffers, Processes, Receiving Areas and Shipping Areas have appropriate input and output Links; whether Control Points have incoming and outgoing Paths etc. In addition, a flow analysis is provided which involves calculation of the flows and bottlenecks in the model. Experiments can be run in step mode, and the display of elements' state is provided. Description of events can be shown on the screen. Rejection of illegal inputs is provided as well as the audible alarms which warn on errors or indicate when a certain condition has been achieved.

Software compatibility is reflected in a possibility for integration with word processors and spreadsheet packages. Data related to the arrival of parts and their shipping from the Shipping Areas can be imported from the text files, whilst output data can be exported to spreadsheet packages for further analysis.

Positive features of the *input/output* facilities are a menu driven interface, periodic output of simulation results, standard output reports, reading the input data from the files and writing the reports to the files. Multiple inputs and outputs are also provided, but for each part type separate Receiving and Shipping Areas should be specified. Dynamic plot of buffer contents can be obtained during experimentation.

Experimentation facilities enable speed adjustment when experiments are run in the auto mode. It is possible to suspend display of elements' state which significantly increases the speed of experimentation. The results can be reset after the warm-up period, but this cannot be done automatically.

With regard to *statistical facilities*, this simulator provides few standard theoretical statistical distributions and a general (Ramberg-Schmeiser) distribution which can yield to approximation of few more statistical distributions when appropriate parameters are chosen. The user can specify the seed of random number stream prior to each run.

With respect to *user support*, documentation provided is readable, although the single spaced text does not contribute to its quality. Many examples are provided. Although most of these examples might be considered as useful, some of them give the impression that the simulator is more complex than it really is.

There are several positive features of this simulator regarding the *financial and technical features*. The price of this simulator is relatively not very high, its hardware requirements are quite moderate (1MB of disk space, 640K of RAM, EGA graphic card), and it is easy to install. Educational and quantity discounts are provided. Regarding the *pedigree*, this simulator might be considered as widely used for 'quick and dirty' modelling both in industry and education. Whilst its predecessor XCELL has been written in extended Basic, XCELL+ was developed using the C language. This simulator was released in 1986.

Several *general manufacturing modelling features* are provided by XCELL+. For example, random breakdowns and setup of Workcentres can be modelled as well as scheduled maintenance. Capacity of elements can be specified as well as buffer delays expressed in Minimum Holding Time. There is a possibility to model rejects from the Workcentres. Different types of parts arrivals can be modelled together with an arrival of parts in batches.

A few *physical elements* can be explicitly modelled such as single machines, buffers, vehicles and trucks (Paths). Some other elements such as assembling, production and batch machines or conveyors can also be modelled with some modelling effort, approximations and dummy elements.

With respect to *scheduling features*, several in-built strategies are provided for scheduling Processes and Carriers. A Process-Switching mechanism determines which Process will be performed next on a particular Workcentre. Processes can be triggered by a certain level of stock in upstream or downstream buffers. Different priority can be assigned to Processes. There are several alternatives for departure scheduling of Shipping Areas. Control mechanism for carriers allows modelling of several dispatching rules for loaded and empty Carriers (vehicle scheduling).

Standard measures of *manufacturing performance* are provided such as throughput, levels of work in progress, utilization of Workcentres, Auxiliary Resources and Maintenance Centre and flowtime. An in-built cost analysis is also supplied as well as Gantt charts, which aim to depict the states of all the elements along a horizontal bar chart.

5.4.2. Negative Features

Some of the shortcomings regarding the *general features* of XCELL+ relate to a lack of possibility for creation of run-time applications, and it is not possible to define a time units. The names of entities are system defined, which might be confusing in the case of larger models. In addition, it is not feasible to perform initialization of model parameters prior to experimentation.

The *visual aspects* are restrained by a lack of an icon and screen editors, which eliminates flexibility in design of model's graphical display. There is no 3-dimensional graphics. Only pre-designed graphical symbols are provided which standardizes graphical presentations of models. An argument for this approach might be that it takes less computer memory, it is faster to run and it is easier to use. On the other hand, more realistic graphical presentations of models might be better understood and appreciated by the clients.

The *coding aspects* perhaps contain the majority of the shortcomings of this simulator. XCELL+ is a purely data-driven and there is no programming flexibility. The user has neither the possibility to add code in order to handle specific logic nor to link model to a lower level language. Due to a lack of programming flexibility, several other criteria are not satisfied such as a provision of built-in and user functions, global variables, and support for programming concepts.

The main shortcomings regarding the *efficiency* are a limited robustness and level of detail that can be modelled by this simulator. Multitasking is not provided as well as the model chaining and automatic model building. The number of queuing policies is rather limited. For example, ordered Buffer have only FIFO and LIFO queuing policies.

Although a certain level of *modelling assistance* is provided, some of this features might be improved. On-line help is quite general and of limited use. A complete separation of model and data cannot be achieved, though some of the input data can be imported. Although the possibility to use the mouse is provided, this is somewhat difficult to implement and make it work.

There are several shortcomings regarding the *testability*. Due to a lack of programming flexibility, many logical features such as attributes, variables, and functions do not exist and therefore cannot be displayed and tested. Error messages indicate various types of errors, but do not provide information how these errors can be corrected. There is no possibility to obtain trace files. List files can be printed directly from the program, but they cannot be saved, edited and printed from the word processors.

With respect to *software compatibility*, it is not possible to integrate XCELL+ with statistical packages, CAD software, data base management systems, expert systems, MRP II and scheduling software. However, it is arguable whether this compatibility is needed for such simple and easy to use package, designed for 'quick and dirty' modelling.

A lack of user defined outputs might be considered as the main shortcoming regarding the *input/output*. Although the standard reports are provided, there is no possibility for the user to request special reports, depending on the modelling objectives.

With regard to the *experimentation facilities*, there is no possibility for an automatic running of several different experiments. A warm-up period cannot be specified in advance, and results should be reset manually. There is no capability for an experimental design. This simulator does not provide the check of the accuracy of simulation results nor the automatic determination of run length.

The *statistical facilities* have also several shortcomings. A small number of theoretical statistical distributions is provided explicitly. Some additional distributions can be obtained using a general distribution which is not always straightforward. There is no possibility for antithetic sampling, distribution fitting, Goodness-of-fit testing or output data analysis.

Some of the shortcomings regarding *user support* are a lack of help line and an interactive tutorial. Reference card is not provided as well as the newsletter.

With respect to the *financial and technical features*, this simulator has a limited portability (it is PC based), and security device is needed for its use.

The main limitation regarding the *general manufacturing modelling features* is a lack of possibility to define and use part attributes, which represents the main obstacle for more flexible scheduling and conditional routing. There is no automatic increase of buffer capacity, and there is no possibility for explicit modelling of operations such as fixturing, palletization and fluid composition.

Although several *physical elements* are explicitly provided, some of the elements can be modelled only with a certain level of approximation, or cannot be modelled at all (eg. tanks and fluids).

Limitations of the *scheduling features* are mainly caused by a lack of programming flexibility. Conditional routing is restricted by a lack of part attributes. It is not possible to push/pull parts from specific positions within the element. There are several routing restrictions. For example, input to Buffer cannot come from another Buffer or Receiving Area, and there are limitations on input (there are only X and Y inputs) and output from the Process. Many of these limitations can be overcome by using dummy elements, which can significantly complicate models.

The main shortcomings related to measures of *manufacturing performance* is a lack of possibility to obtain a special user defined reports. A production sequence summary report is not provided. In addition, there is no information about rework and scrap level and there are no interruption reports.

5.5. Evaluation of ProModelPC

A critical evaluation of ProModelPC is provided in this section. Positive features of this simulators are analyzed in sub-section 5.5.1, whilst sub-section 5.5.2 addresses its negative features.

5.5.1. Positive Features

There are several criteria regarding the *general features* that make this simulator appropriate for the simulation of manufacturing systems. ProModelPC is a data driven simulator with a possibility for additional programming in the form of the Turbo Pascal subroutines. It is manufacturing oriented and includes manufacturing terminology. The user can define the names of entities. The basic concepts are relatively easy to learn once the documentation is read and tutorial run. Different time units and measures of length can be used in models.

The *visual aspects* are of an average quality. It is possible to run experiments with animation. Animation is concurrent and full. A library of simple icons is provided together with an icon editor for easy creation of additional icons. It is possible to further enhance the graphical displays of models with additional text, lines or icons on the screen. The colour of already created icons can be changed. Zoom function and panning are provided as well as the print screen facility, and a facility for easy copying of icons. Icons can be changed during simulation as parts changes their type and name.

Positive features of *coding aspects* primarily refer to programming flexibility achieved by a possibility to link ProModelPC with the Turbo Pascal subroutines. In addition, a relatively simple logic of models can be modelled by internal facilities such as IF-THEN commands, built-in and user functions, global variables. It is possible to write comments during model development which is particulary important for Routing module. Text/code manipulation is possible, due to a possibility to develop models directly from the text editor.

There are several *efficiency* related positive features of ProModelPC. Its robustness is achieved by programming flexibility. Automatic model building can be optionally chosen. Interactivity is provided during model development and experimentation.

Positive features of *modelling assistance* are modularity, model and data separation, an easily accessible on-line help and use of the mouse. In addition, a text editor is provided within this simulator as well as an automatic editing of data.

Although several facilities regarding *testability* are provided, models are generally not easy to debug. However, error messages are provided, though they are of very limited usability. Display of variables is also provided as well as a display of events on the screen. Trace and list files can be obtained. Experiments can be run in step mode.

Software compatibility is achieved by a possibility to integrate ProModelPC with spreadsheet packages, statistical packages and word processors.

There are several positive features regarding the *input/output*. A menu driven interface with pull-down menus is provided. Multiple inputs and outputs can be modelled. General output reports are automatically provided as well as static graphical reports in the from of pie charts and bar graphs. Output reports are quite understandable. It is possible to obtain periodic output of reports. Results can be available at any moment of simulation. It is possible to read input data from files, and to obtain reports written in the file. It is also possible to write reports directly to printer. Snapshot reports are provided as well a summary report for multiple runs.

With regard to the *experimentation facilities*, several features are provided. Automatic batch run is possible. ProModelPC Interface enables alteration of various model parameters, and running a series of 'what-if' scenarios without having to work within the model itself. It is possible to specify breakpoints of simulation and to chose an adequate speed of simulation.

Positive features of the *statistical facilities* are expressed by facilities such as in-built theoretical statistical distributions, possibility to use user-defined distributions, different random number streams and distribution fitting. In addition, output data analysis is provided in form of the ProModelPC Interface Multiple Replication Summary, which generates a variety of summary reports for multiple experiments both in the textual and graphical form.

The main positive features of *user support* is a documentation that is quite easy to read, an interactive tutorial which might be useful for the first phases of learning this simulator, and training course organized by supplier.

With respect to the *financial and technical features* of this simulator, it is claimed that its price is relatively moderate comparing, for example, to the price of WITNESS. It is easy to install with reasonable minimum hardware requirements (640K RAM, 4.5MB of hard disk and EGA colour graphics adapter and monitor). Educational discount is provided.

With regard to the *pedigree*, many users of ProModelPC can be found in industry as well at universities. This simulator was released in 1986.

Many *general manufacturing modelling features* are provided such as modelling of shifts, breakdowns, setup of routing locations, rejects modelling, part attributes modelling and arrival of parts in batches. Several typical manufacturing operations can be modelled such as assembling, disassembling, containerization palletization and inspection operation.

With respect to the *physical elements*, the following elements are explicitly provided: AGVs and trucks, different types of conveyors and cranes. Elements such as labour, tools and fixtures can be modelled as resources, whilst buffers and various types of machines can be modelled as routing locations using the appropriate logic.

Scheduling features are mainly facilitated by a number of built-in or user defined scheduling rules. Conditional routing is possible as well as to assign different priorities to model elements. It is also possible to specify the quantity of parts to be moved between elements. Vehicle scheduling is provided in addition to modelling of vehicle acceleration and deceleration. Finally, scheduling optimization facility is provided which executes simulations of all possible combinations of production schedules (for less than 10 part types) and finds the order release sequence with the shortest throughput time and highest throughput.

A variety of measures of *manufacturing performance* are provided, such as throughput, work in progress, utilization of production equipment and makespan. In addition, the user can obtain a production sequence summary, which summarizes sequences of operations performed on different part types.

5.5.2. Negative Features

The main shortcomings of ProModelPC's *general features* are a lack of possibility to create run time applications, and a relatively small length of entity names. In addition, some features are not very straightforward to learn and use such as modelling of conveyor system or multiple logical (IF-THEN) conditions.

Some of the shortcomings regarding the *visual aspects* are as follows. It is not possible to merge icon files nor to manipulate with created icons. It might be more convenient if the graphics is developed concurrently rather than after, or before model development. There is no graphical indication of element's status. There is a limitation on number of displayed icons (12 icons in student version, and 36 icons in a commercial version). It is not possible to develop icons with multiple colours nor to design 3-dimensional graphics.

The main weakness of the *coding aspects* is very limited flexibility of internally provided logical constructs. Because of this, probably in many cases models should be linked to the Turbo Pascal subroutines to handle more complex logic. Names of the functions, variables and attributes are system defined, which might be confusing in case of complex models. There is a strict limitation on the length of commands in routing module, which makes specification of complex conditions very difficult.

There are several shortcomings regarding the *efficiency*. There is no automatic saving, and it is not possible to save model status. Adaptability is limited, because it is not possible to change model parameters during experimentation, and continue simulation. If changes are to be made, simulation has to be started from the beginning. There is no model chaining nor it is possible to access the operating system within this simulator. Model execution time is quite long when the graphics is used. ProModelPC is case sensitive, which is also inconvenient. Problems with reliability occasionally occur for no apparent reason, and the only thing to do is to reset a computer. Finally, it is not possible to edit partially developed models, and it is not easy to edit completed models. Models that are not complete cannot be retrieved for further development, so models are to be developed again from scratch.

With regard to the *modelling assistance*, the main problem is a lack of adequate prompting during model development. Although a certain type of prompting is provided during an automatic model building, there is no such prompting that would help constructing the logic of model. On-line help is too general, and there is no facility for writing documentation notes apart from the possibility to write comments within the modules. When model's logic is developed using the text editor, the order of added commands is sometimes changed, ie. model retrieved is different from those developed and saved using the text editor.

One of the main characteristic regarding the *testability*, is that the debugging of models is not easy. The quality of error messages is very poor. They are provided prior to experimentation and not at the model entry. These messages indicate occurrence of errors, but they do not give the information what is wrong, where, and how an error can be corrected. There is no display nor the access to part attributes. Neither a display of element's state is provided nor a display of workflow path. Illegal inputs are not rejected except when, for example, theoretical statistical distribution is to be used when only constants or user defined distributions can be used. Explode function is not provided and it is not possible to move simulation clock backwards.

Shortcomings regarding *software compatibility* are expressed by inability for integration with CAD software, data base management systems, expert systems, MRP II software and scheduling software.

Some of the weaknesses regarding the *input/output* relate to a lack of dialogue boxes, selection buttons and dynamic graphical output. In addition, the user is not given the choice of selecting the name of the file into which output data is saved. Therefore, the data from a previous run is over-written. The only way to save output data is to exit from ProModelPC, and transfer output file into another directory.

With respect to the *experimentation facilities*, it is not possible to re-start models from non empty state and there is no experimental design capability. Although it is possible to estimate the accuracy of results from the statistical analysis of output, there is no explicit facility for the accuracy check. Automatic determination of run length is not provided.

The main limitations of *statistical facilities* are a relatively small number of random number seeds (10), which might be insufficient for large models, and a lack of facility for antithetic sampling.

With regard to *user support*, it is stated that although the documentation is easy to read, it does not provide enough examples for complex modelling constructs. The same applies to the tutorial which should include more modelling examples instead of general information about the ProModelPC and simulation modelling.

The main limitation concerning the *financial and technical features* is a limited portability, because ProModelPC can be used only on IBM XT, AT, PS/2 and compatibles (it would be more convenient to run large models on workstations). Another problem is a security device needed for the use of ProModelPC, which is especially inconvenient when this simulator is to be used for education.

Some of the shortcomings regarding the *general manufacturing modelling features* are as follows. There is no explicit maintenance modelling apart from a possibility to model it as a downtime. Automatic increasing of buffers capacity is not provided nor the job lists for labour. There is no possibility to model any operations related to fluid processing.

Although a variety of *physical elements* can be modelled with an appropriate logic, it might be more convenient if physical elements such different types of machines were provided. In addition, there are no elements regarding the continuous processing such as tanks and fluids.

Weaknesses of the *scheduling features* are expressed in a relatively small number of in-built scheduling rules. For example, there are 13 in-built rules for routing of parts and only 2 rules for vehicle scheduling. There is no preemption possibility, it is not possible to retrieve and use predefined part routing, and there is no departure scheduling for shipping area.

With regard to the measures of *manufacturing performance*, there are no explicit facilities for due dates monitoring and manufacturing costs analysis. In addition, information about rework and scrap level is not provided as well as interruption reports.

5.6. COMPARISON OF THE EVALUATED SIMULATORS

This section provides a comparison of the evaluated simulators. Information presented here is derived from the evaluation of these simulators presented in Section 5.2 - 5.5 as well as from the overall impressions and experience gained through learning and using these simulators. A proposed rating of the evaluated simulators according to their performance in terms of the groups of evaluation criteria is presented in sub-section 5.6.1. Sub-section 5.6.2 addresses the suitability of the evaluated simulators for particular purposes.

5.6.1. Rating of the Evaluated Simulators

In order to compare the evaluated simulators, a rating of these simulators has been established. This rating is based on an analysis of the simulators being evaluated. As such, it should be considered as a relative measure of quality of these simulators from the perspective of groups of criteria, rather than as an absolute value.

Table 5.1 shows a proposed rating for the simulators being evaluated, in terms of the general quality of features within particular groups of criteria. The rating interval used in this assessment is similar to the one proposed by Ekere and Hannam (1989). The general quality of simulators with respect to particular groups of criteria is rated from 1 to 10, where 1 represents very poor quality or absence of the features within particular groups of criteria, whilst grade 10 represents excellent quality. Accordingly, we propose that 5 is taken to be a 'nominal acceptance level', or NAL for short. The grades for a certain group of criteria that are above the NAL indicate that a package is performing adequately, whereas those below signify the opposite. Whilst the NAL is clearly subjective, it does provide a level against which the relative performance of a package can be measured and reflected on. Since evaluation cannot be entirely objective, this qualitative measure of performance, the NAL, does provide a relative measure. However, clearly any particular grade is merely a 'qualitative' number, and the rules of arithmetic can only be applied with caution and with caveats, if at all.

Table 5.1 Comparison of evaluated simulators in terms of groups of criteria

SIMULATORS / GROUPS OF CRITERIA	WITNESS	SIMFACTORY II.5	XCELL+	ProModel PC
General Features	8	8	7	7
Visual Aspects	8	7	5	6
Coding Aspects	7	5	1	6
Efficiency	8	7	6	7
Modelling Assistance	8	7	7	6
Testability	8	7	6	5
Software Compatibility	6	7	6	7
Input/Output	8	7	6	7
Experimentation Facilities	7	8	6	8
Statistical Facilities	7	8	5	7
User Support	8	8	7	7
Financial and Technical Features	4	6	7	8
Pedigree	9	8	8	8
General Manufacturing Modelling Features	8	8	6	7
Physical Elements	8	8	6	7
Scheduling Features	8	7	5	7
Manufacturing Performance	8	7	6	7

The above table shows that all simulators are rated quite high regarding *general features*. They are all data driven and manufacturing oriented. WITNESS and SIMFACTORY II.5 are considered be slightly more user friendly than the other two simulators because of several features such as quality of graphics, assistance provided in modelling, user support provided etc. On the other hand, XCELL+ is the easiest to learn and use because of its simplicity, whilst ProModelPC balances ease of learning and use with user friendliness and comprehensiveness.

Visual aspects are rated highest for WITNESS, which satisfies the majority of criteria within this group. SIMFACTORY II.5 follows with quality of graphics, which is also above the NAL. The next in the sequence is ProModelPC, and finally XCELL+ which uses symbolic graphics.

With respect to *coding aspects*, WITNESS and ProModelPC are both rated above the NAL, with WITNESS being graded higher than ProModelPC. However, none of them have achieved a very high rating because of the limited flexibility of the internal languages provided. The quality of SIMFACTORY II.5 regarding this group of criteria is even lower, at the NAL level, whilst XCELL+ does not allow for any programming at all.

The *efficiency* related rating of the simulators also shows good quality. WITNESS is rated the highest, mainly because of its relatively high robustness and interactivity. Next in line are SIMFACTORY II.5 and ProModelPC, with SIMFACTORY II.5 being better in features such as adaptability and interactivity, whilst ProModelPC is better regarding robustness. Finally, XCELL+ is lacking robustness, but it has a short time scale for model building.

Modelling assistance is slightly better ranked for WITNESS than for the other simulators. However, SIMFACTORY II.5 is graded quite well because of features such as model and data separation and the automatic editing of data. XCELL+ is graded quite high due to its prompting and rejection of invalid values. The last simulator in the sequence is ProModelPC. Although, it possesses several features regarding modelling assistance, some of them are of little use.

With regard to *testability*, WITNESS again outperforms all the other simulators. It is rated quite high because it has many features that facilitate model verification. Next are SIMFACTORY II.5 and XCELL+ respectively. The lowest rated is ProModelPC, because testability is perhaps the weakest feature of this simulator. The main reason for this is the poor quality of error messages, which do not even provide information about where an error has occurred.

The quality of features with regard to *software compatibility* is above the NAL, but not very high. Whilst all simulators under consideration enable integration with word processors and spreadsheet packages, SIMFACTORY II.5 and ProModelPC are slightly better ranked, because they can be linked with data bases and statistical packages respectively. At the moment, none of them can be integrated with CAD software or expert systems.

Concerning the *input/output* features, WITNESS has achieved the highest performance mainly because of the variety of standard and special user-defined reports, and its facilities for user friendly input of data. Next in the sequence are SIMFACTORY II.5 and ProModelPC, providing, in addition to standard reports, facilities such as summary reports for multiple runs or snapshot reports. Finally, XCELL+ is last in the sequence, mainly because of its lack of user defined reports and summary reports for multiple runs.

SIMFACTORY II.5 and ProModelPC are the best ranked regarding *experimentation facilities*, providing features such as facilities for multiple runs, accuracy checks and the automatic determination of run length (SIMFACTORY II.5) or a facility for the automatic testing of 'what if' scenarios (ProModelPC). Experimentation facilities for WITNESS are slightly worse, mainly because the setting up of automatic experimentation is not straightforward. Finally, XCELL+ is rated just above the average, because it cannot automatically run multiple experiments. None of the simulators has a facility for experimental design.

It is judged that SIMFACTORY II.5 has the best *statistical facilities* in comparison to other evaluated simulators. It not only provides features such as a number of theoretical statistical distributions and antithetic sampling, it also enables distribution fitting and Goodness-of-fit tests. WITNESS and ProModelPC follow, where WITNESS is lacking, for example, facilities for distribution fitting and output analysis, whilst ProModelPC is lacking a large number of random number streams and antithetic sampling. Finally, XCELL+ is rated at the NAL, because of its small number of theoretical statistical distributions, and lack of antithetic sampling and distribution fitting.

With regard to *user support*, WITNESS and SIMFACTORY II.5 are rated the highest. The suppliers of both simulators provide a high level of support in the form of user group meetings, help-lines etc. The next in sequence are XCELL+ and ProModelPc, with above the NAL levels but not so extensive as is the case for the other two simulators.

ProModelPC is ranked as the best regarding *financial and technical features*. Its price depends on the number of operations purchased, and even so it is the cheapest simulator (in comparison with the other simulators evaluated), with moderate hardware requirements. This was the main reason for a such high scoring, although it might be argued that it has a limited portability. The next simulator is XCELL+ with similar characteristics, but it is slightly more expensive. SIMFACTORY II.5 follows with a significantly higher price, but with high portability, and free software trials. In addition, this is the only simulator among those evaluated that does not require a security device. WITNESS is in the last position regarding this group of criteria, because its price is the highest and its hardware requirements are high.

All simulators are rated highly regarding their *pedigree*, because they are all quite well known and widely used. They are all of similar age, as they were all released on the market around 1986. Information about these simulators appear in various sources of literature. However, WITNESS is ranked slightly better then other simulators, due to its SEE-WHY origin (SEE-WHY introduced visual interactive systems).

Concerning the number and quality of *general manufacturing modelling features*, WITNESS and SIMFACTORY II.5 are rated the highest. Both these simulators enable the modelling of a variety of features typical of manufacturing systems. Then follows ProModelPC, and finally XCELL+, whose main shortcoming is an inability to model part attributes.

The same gradation applies regarding the *physical elements*. Both WITNESS and SIMFACTORY II.5 explicitly provide a variety of physical elements typical of manufacturing systems, such as various types of machines and materials handling systems. Different physical elements are modelled by an appropriate routing logic when ProModelPC is used. A similar approach applies to XCELL+ which requires, for example, the use of dummy elements if a certain type of machine is to be modelled.

Concerning *scheduling features*, WITNESS was given the highest grade, mainly because one can model a variety of scheduling strategies using both the in-built input/output rules and additional programming. SIMFACTORY II.5 and ProModelPC follow with similar characteristics, although the modelling of scheduling is less flexible.

Finally, it was estimated that the scheduling features provided by XCELL+ are of an average quality, mainly because of its restricted flexibility to model a variety of scheduling strategies.

A similar gradation applies to the group of criteria regarding *manufacturing performance*. Although all simulators provide automatic collection of statistics, there is a difference in the number, quality and form of reports. An additional factor that was considered important is the facility to obtain special user-defined reports. Regarding these criteria, WITNESS was rated at the highest level, following by SIMFACTORY II.5 and ProModelPC, and finally by XCELL+.

An additional analysis of the rating of the evaluated simulators in provided in Table 5.2, and Figures 5.1 and 5.2. Table 5.2 shows deviations from maximum scores obtained by a simulator, specified for each group of criteria. Therefore, the closer a value of deviation is to zero, the better. Figure 5.1 shows the deviations from maximum scores within each group of criteria. For each simulator, the total number of groups of criteria that have a certain level of deviation is counted. Cumulative values of these deviations are shown in Figure 5.2.

Table 5.2: Deviations from maximum scores specified for the groups of criteria

SIMULATORS / GROUPS OF CRITERIA	WITNESS	SIMFACTORY II.5	XCELL+	ProModelPC
General Features	0	0	-1	-1
Visual Aspects	0	-1	-3	-2
Coding Aspects	0	-2	-6	-1
Efficiency	0	-1	-2	-1
Modelling Assistance	0	-1	-1	-2
Testability	0	-1	-1	-3
Software Compatibility	-1	0	-1	0
Input/Output	0	-1	-2	-1
Experimentation Facilities	-1	0	-2	0

Statistical Facilities	-1	0	-3	-1
User Support	0	0	-1	-1
Financial and Technical Features	-4	-2	-1	0
Pedigree	0	-1	-1	-1
General Manufacturing Modelling Features	0	0	-2	-1
Physical Elements	0	0	-2	-1
Scheduling Features	0	-1	-3	-1
Manufacturing Performance	0	-1	-2	-1

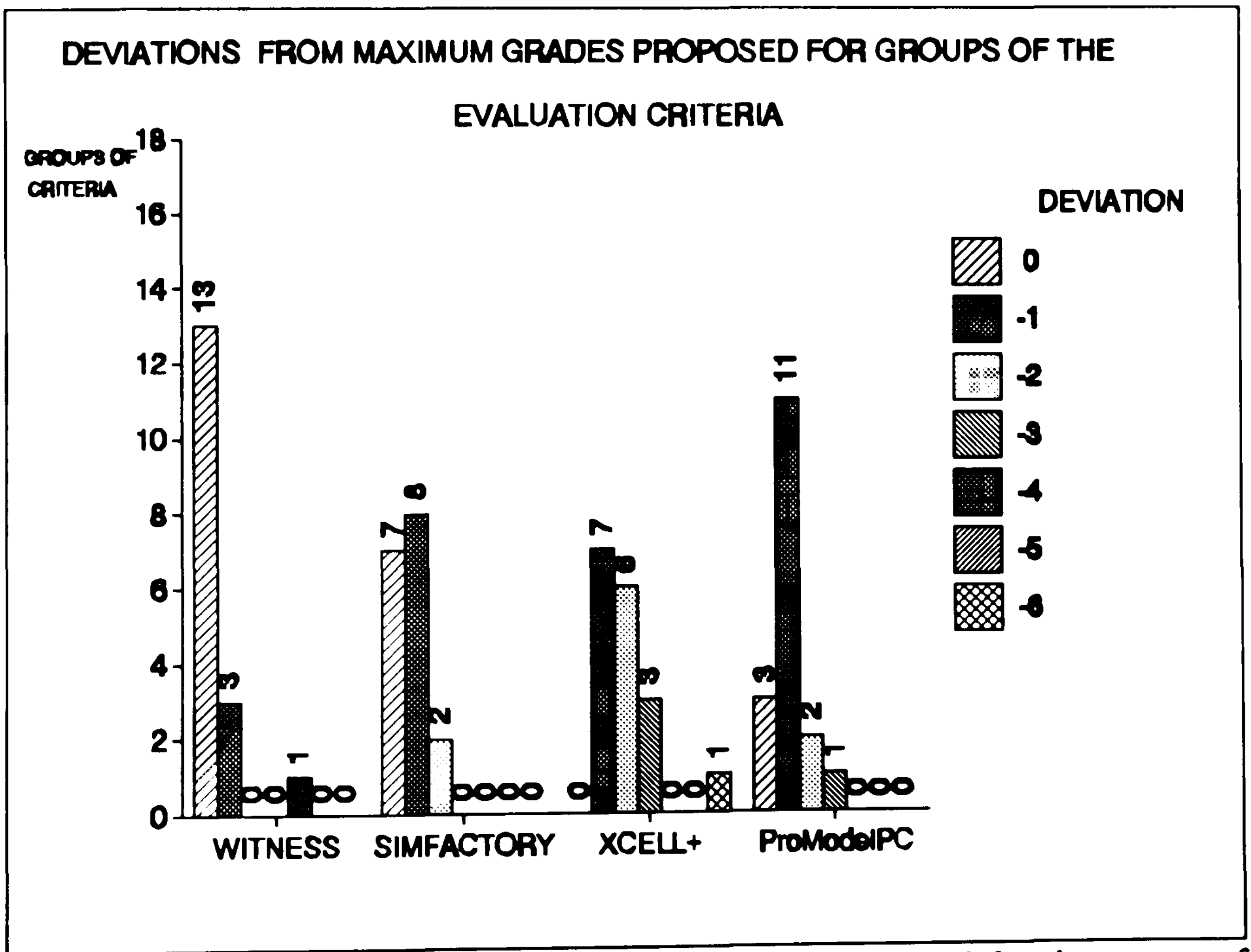


Figure 5.1 Deviation from maximum values of rates proposed for the groups of criteria

Figure 5.1 shows, for example, that WITNESS scored its maximum value for 13 groups of criteria. SIMFACTORY II.5 has a balanced number of the highest and second highest grades (7 and 8 groups of criteria respectively). XCELL+ did not achieve a maximum score for any group of criteria, whilst ProModelPC was given the most grades which deviated from the maximum values by only one score (11 groups of criteria).

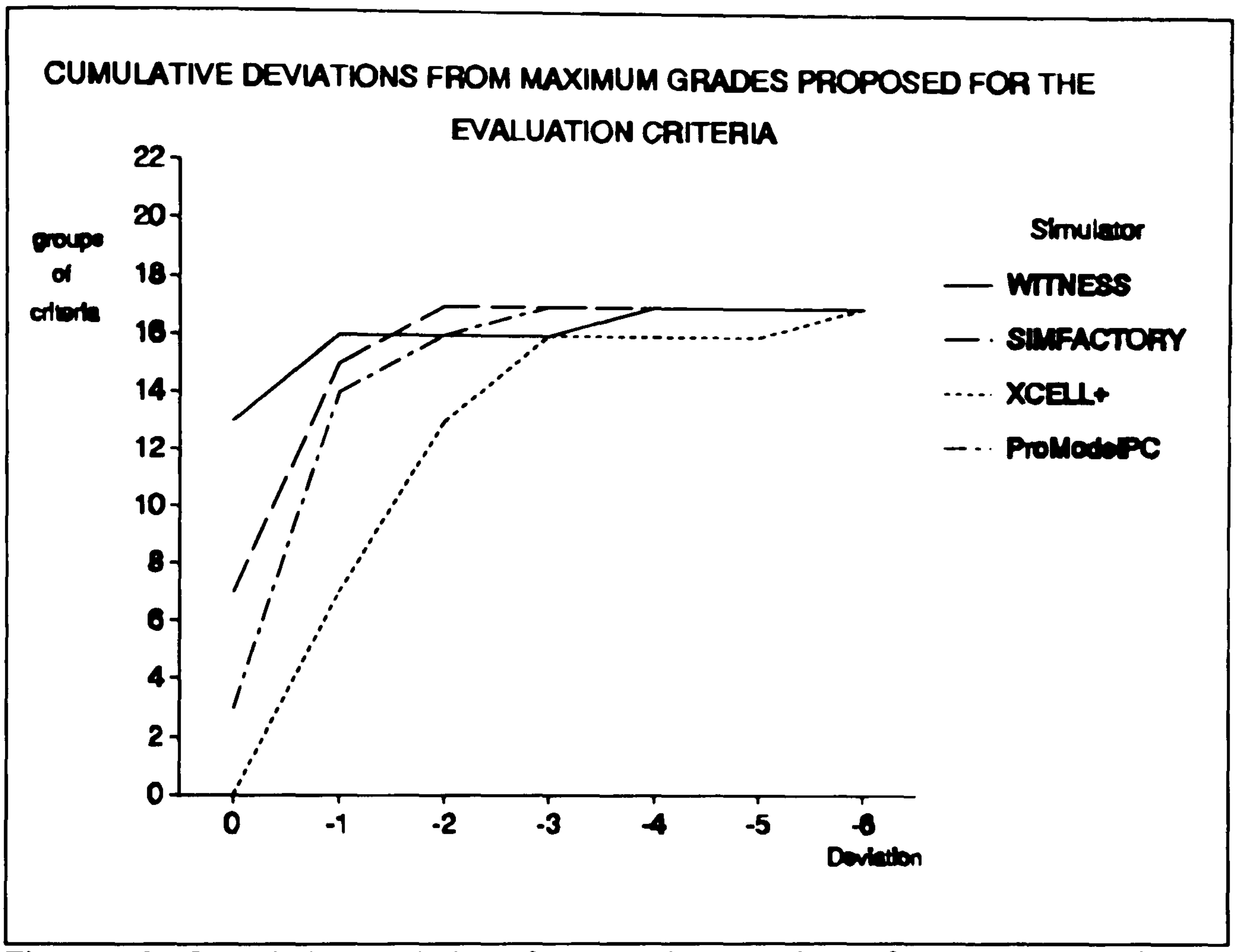


Figure 5.2 Cumulative deviations from maximum values of rates proposed for the groups of criteria

Cumulative deviations from maximum grades shown in Figure 5.2 provide a basis for further analysis. With regard to these deviations, the higher the line is at its starting-point the better, because at this point there is no deviation from maximum grades. As the lines shown represent a cumulative values, they obviously end up at the same level which represents a total number of groups of criteria (17).

Nevertheless, significant differences between these cumulative lines are apparent. At the first two levels of deviation (0 and -1) there is the same order of simulators' performance: WITNESS scored the best, then SIMFACTORY II.5 followed by ProModelPC and XCELL+. At the second level of deviation (-2), the order is changed: SIMFACTORY II.5 has achieved its maximum deviation, ProModelPC has reached the level of deviation of WITNESS, and then follows XCELL+. At the subsequent deviation level (-3), ProModelPC ascends to maximum deviation, whilst deviations of WITNESS and XCELL+ become equal. At the following level (-4) WITNESS finally reaches its maximum deviation, whilst XCELL+ achieves this at the last level (-6).

The above comments on cumulative deviations from maximum grades within each group of criteria further support the claim that although some simulators might have a better overall performance than the others, they do not perform equally well for all groups of criteria.

5.6.2. Suitability of the Evaluated Simulators for Particular Purposes

The results of the comparison of the evaluated simulators revealed several facts. Although some simulators scored higher than the others, for example WITNESS v. XCELL+, there is no simulator that satisfies all criteria, and shows good performance in all features. Usually, features of the simulators such as robustness and comprehensiveness require more learning and an increase in model development time, demanding at the same time higher costs of purchasing.

Consequently, there is no simulator which is equally good for all the purposes of education, 'quick and dirty' modelling in industry, or complex and detailed modelling in industry. As was shown in Chapter 4, the level of importance of certain software features is different for different purposes. In this context, on the basis of the evaluation of the simulators, and on the basis of experience obtained in using these simulators, a suggested suitability of simulators for particular purposes is shown in Table 5.3.

Table 5.3. The suitability of evaluated simulators for particular purposes

EDUCATION	'QUICK & DIRTY' - INDUSTRY	DETAILED/COMPLEX - INDUSTRY AND RESEARCH
1. XCELL+	1. SIMFACTORY II.5	1. WITNESS
2. SIMFACTORY II.5	2. XCELL+	2. ProModelPC
3.+ WITNESS	3. ProModelPC	3. SIMFACTORY II.5

3.+ ProModelPC	4. WITNESS	4. XCELL+

Table 5.3 shows that XCELL+ can be considered as the most appropriate for education. The main reason for this is its simplicity, ease of learning and use, and short model development time. The second best simulator for education is SIMFACTORY II.5 which can be fairly easily learnt and used, providing at the same time good overall features. At the third level are the more comprehensive and difficult to learn WITNESS and ProModelPC, which require more experienced users.

With regard to 'quick and dirty' modelling, it is appraised that SIMFACTORY II.5 is the most suitable simulator. In addition to its relatively easy learning and use, it has quite straightforward modelling of many features typical of manufacturing systems. The second position is assigned to XCELL+, which is even easier to learn and use, but is more inflexible. Although both WITNESS and ProModelPC can also be used for 'quick and dirty' modelling in industry, they are in the last position because it takes quite long to learn and use them properly.

For detailed/complex modelling in industry and research, it is estimated that the most suitable simulator is probably WITNESS. This simulator is quite comprehensive, robust and flexible, as much as simulators can be. At the same time it is user friendly and easy to use once it is learnt. The second most suitable simulator is ProModelPC mainly because of its programming flexibility and the possibility of linking to a lower language, although its models are not easy to debug. Then follows SIMFACTORY II.5, which despite many general good features, is quite limited in flexibility and robustness. Finally XCELL+ might be considered as the least suitable for complex and detailed modelling due to its simplicity and inflexibility.

5.7. SUMMARY

This chapter provides an evaluation of several manufacturing simulators under consideration. A delineation of the basic characteristics of each simulator as well as a description of their models' development is provided. A critical evaluation including positive and negative features of each simulator (derived from the perspective of groups of criteria presented in Chapter 4) is given in the appropriate Appendices. During the evaluation not every single criteria within each group was examined, because the aim was to generally perceive basic features of each simulator. Specific features are probably going to change and be added to with new releases of simulators under consideration.

A comparison of evaluated simulators is provided. The general quality of each group of criteria was ranked for each simulator. This revealed that although all simulators belong to the same type of simulation software, there is a variety of differences between them. In addition, none of the simulators satisfies all criteria, and none is equally good for all purposes. Although some simulators are more comprehensive and flexible than others, a simulator that can fit any manufacturing problem does not exist. At the same time those simulators that are more robust and adaptable are usually more expensive and difficult to learn and use properly. This confirms the statement that "the less work required of the user the more must be done by the package itself, which increases its complexity, size, cost and execution times" (Carrie, 1988).

The fact that a selection of simulation software is a matter of compromise between many factors is substantiated by this research. One of the most important factors that determines which software is more suitable than others is its intended purposes. Other factors to consider are financial constraints and subjective factors such as individual preference and experience in using simulation software.

CHAPTER 6. MANUFACTURING SIMULATORS: WHAT IS NEEDED AND HOW TO CHOOSE

6.1. INTRODUCTION

This chapter provides a further development and completion of the knowledge acquired during this research. It considers manufacturing simulators from the users' perspectives and analyzes those features that are needed, how these simulators can be improved, and how to choose an adequate simulator.

The main intention is to produce ideas that can be of a practical use for users, vendors and developers of manufacturing simulators and simulation software in general. A survey has been carried out to find out users' opinions about simulation software, problems they experience using this software, and their requirements for the enhancement of simulation software. A methodology for software selection is derived on the basis of the experience gained during this research and from studying the literature. Finally, improvements to manufacturing simulators are proposed with regard to their purposes, which should result in easier and more effective modelling.

A survey conducted is described in section 6.2. The survey, plus results from previous chapters, provide the basis in section 6.3 for a proposed methodology and guidelines for the selection of manufacturing simulators. Also based on the survey results and findings from previous chapters, improvement proposals for manufacturing simulators are given in section 6.4. Section 6.5 contains a summary of this chapter.

6.2. A SURVEY

This section presents the results of a survey on the use of simulation software in manufacturing environments. The survey of a number of simulation specialists in industry and universities across Europe was carried out to discover whether users are satisfied with the simulation software they use, and how this software might be further improved.

More details about the purpose of this survey are provided in sub-section 6.2.1. Sub-section 6.2.2 contains some information about the survey sample. Results of the survey are presented in sub-section 6.2.3, whilst the findings of the survey are discussed in sub-section 6.2.4.

6.2.1. Purpose of the Survey

The main purpose of the survey was to investigate users' requirements of simulation software, especially software used for manufacturing simulation, and to seek opinions about ways of improving current simulation software tools to better satisfy their needs. This survey was conducted at the later stages of the research presented in this thesis, in order to improve, expand and confirm research findings. It was believed that information about simulation software provided by other users could contribute to achieving the objectives of this research.

The questionnaire distributed to the participants in the survey consists of nine questions dealing with the type of simulation software used (1); the specification of particular packages used (WITNESS, SIMFACTORY II.5, SIMAN/CINEMA, ProModelPC, XCELL+, INSTRATA or other) (2); the purpose of using simulation (3); general opinions about each software used (4); and the types of systems being modelled (5). Other questions include an estimation of how successful the simulation studies carried out were from the point of view of the software used (6). In particular, users were asked whether substantial approximations had to be made due to limitations of the software, or whether all desirable features of the systems under consideration could be modelled. The participants were also asked to list the main weaknesses and limitations of the software used (7), as well as the most important positive features (8). Finally, they were asked to specify the most important features that should be included in existing simulation packages, and that are to the best of their knowledge not yet provided (9). Appendix J includes a copy of the questionnaire distributed in this survey.

The majority of the questions regarding opinions about the software and possible ways of improving it (questions 4,6,7,8 and 9) were open-ended. It is believed that this approach avoids the possibility of putting suggestions into the minds of the participants, and hence gives better and unprejudiced responses.

6.2.2. Survey Sample

The survey sample includes a number of regular simulation users both in educational institutions and industry around Europe. Some of the participants from Great Britain include academics from the University of North London, Lancaster University, the University of Birmingham, Loughborough University, the University of Strathclyde, the University of Salford, the University of Sheffield, Newcastle University, the University of Durham, the University of Plymouth etc. Academic participants from other countries include those from Netherlands, Germany and Denmark.

Participants from industry include simulation users from British Aerospace, British Airways, Lucas Engineering & Systems Ltd, the Rover Group, BOC Ltd, BASS plc, the Johnson Matthey Technology Centre etc.

The survey sample was not selected by any formal statistical method. The participants were known to be, or were believed to be, regular users of simulation, and hence were selected deliberately for this reason. It was intended to obtain a sample of users experienced mainly in the use of simulators (referred to as simulation packages in the questionnaire) rather than languages. The response rate was moderate, 30% out of 120 distributed questionnaires (36 questionnaires in total). In addition, the ratio of responses from universities and from industry was about 70% and 30% respectively, although an approximately equal number of questionnaires was distributed to each group of users.

It was intended to distribute the majority of questionnaires with the help of software vendors. As not all of them were equally cooperative, the responses of users of some packages was greater then was the case for some other software products. Nevertheless, the majority of responses were obtained from academics using on average several simulation packages, which enhances the generality of results obtained for this group. Open-ended questions have tried to provoke more general answers, based on overall experiences in using simulation software.

Not only was the response significantly higher from university users, on average each response from a university provided more information than a response from a user in industry. All these facts might raise questions concerning the statistical significance of the obtained results. However, this is the fate of surveys of this type, and it is believed that the deliberate selection of the survey participants, all of whom have experience in simulation, in fact enhances the importance and usefulness of the results.

6.2.3. Results of the Survey

The responses of the survey are classified in two groups, distinguishing users at universities and users in industry. The main reason for this was to discover whether and how software purpose influences the requirements for simulation software.

(i) Responses from users at universities

With regard to the type of software used, 51.7% of the users at universities use only simulators, 44.8% use both simulators and simulation languages, and 3.5% use only simulation languages. Analysis of the specification of simulation software tools used reveals that more than half (51.7%) of the users use only one software tool, but the other half use more than one software tool, up to six different software packages. Table 6.1 summarizes results obtained regarding the number of simulation packages used.

Table 6.1 The results obtained with regard to the number of simulation packages used at universities.

NUMBER OF SIMULATION PACKAGES USED	PERCENTAGE OF USERS (%)
1	51.7
2	13.8
3	6.9
4	17.2
5	3.5
6	6.9
	100

With respect to the simulation purpose, 20.7% of participants use simulation only for modelling real systems, 10.3% use simulation only for education, whilst the majority of 69% use simulation both for modelling real systems and education.

Common elements from the responses concerning general opinions about the software used are summarized in Table 6.2, together with the percentage of users that have specified a certain software feature. Several features can have the same independent percentage responses. For example, features 'easy to learn' and 'biased to manufacturing problems', both have 13.8% responses in Table 6.2.

Table 6.2 A summary of users' general opinion about the software (universities)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Too limited for complex problems	24.1
- Easy to use	20.7
- Good graphics	17.2
- Easy to learn - Biased to manufacturing problems	13.8
- Slow - User friendly - Poor statistical support	10.3
- Inadequate experimentation facilities - Difficult to validate models	6.9

With regard to the systems being modelled, 31% of users model only manufacturing systems, 44.9% are involved in modelling both manufacturing and other types of system, whilst 24.1% model only other types of systems.

When asked about the success of modelling, 27.6% of participants declared that they have been able to model desirable features of the systems being modelled, 37% have managed to model most of the features, whilst 34.5% had problems in modelling due to software limitations and inflexibility.

Table 6.3 summarizes responses concerning the main limitations and weaknesses of the software used, whilst Table 6.4 summarizes responses regarding the most important positive features of the software used.

Table 6.3 A summary of users' opinion about the main limitations of the software (universities)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Restricted flexibility	31.0
- Validation difficulties	17.2
- Slow - Lack of facility for output analysis	13.8
- Difficult to use - Difficult to learn - Lack of facility for experimental design - Poor statistics	10.3
- Lack of database linkages - Limits to the size of models	6.9
- Expensive	3.4

Table 6.4 A summary of users' opinion about the most important positive features of the software (universities)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Graphics (animation)	34.5
- Ease of use	20.7
- Ease of learning - Automatic report generation	13.8
- User support - User interface	10.3
- Flexibility - Documentation - Good statistical analysis - Speed of modelling	6.9
- Interface with other software - Support for UNIX platforms - Incorporated cost analysis - Easy check of 'what-if' questions - Cheap	3.4

Finally, a summary of the features that academic users would like incorporated in simulation software that could improve the software they use is presented in Table 6.5. The most common feature specified, 'better software compatibility', is further sub-divided with regard to compatibility with specific types of software, according to the corresponding percentage responses.

Table 6.5 A summary of users' opinion about the features that should be included in simulation software (universities)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Better software compatibility	24.1
Link to databases	17.2
Link to spreadsheets	10.3
Link to CAD software	3.4
Link to statistical packages	3.4
Link to MRP scheduling software	3.4
- Facility for output analysis	17.2
- More flexibility - Help in experimental design	13.8
- Better and more intelligent on line-help - Better experimentation facilities - Support of standard programming concepts	10.3
- Elimination of memory limitations - Better documentation - Easy model editing	6.9
- Ability to create run-time applications - Automatic save - More prompt to save - Hierarchical model building - Low cost of software - Easy design of on-line reports - Availability on standard hardware and software systems	3.4

(ii) Responses from users in industry

With regard to the type of software used, 72.7% of users in industry use only simulators, 18.2% use both simulators and spreadsheet software, and 9.1% use only simulation languages. Examination of the number of simulation software tools used shows that all users use only one simulation software product (100%).

Considering the simulation purpose, 90.9% of participants use simulation only for modelling real systems, whilst 9.1% use simulation both for modelling real systems and education, and none of them use simulation only for education.

Analysis of the responses concerning general opinions about the software used is summarized in Table 6.6, together with the percentage of users that have specified a certain software feature.

Table 6.6 A summary of users' general opinion about the software (industry)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Generally very good - Interactive - Graphics - Slow to run - Easy to use but only when applied to standard systems	72.7
- Reasonably easy to learn - Difficult to use for non standard systems - Biased to manufacturing problems	18.2
- Quick - Easy to use - Lack of good support for fluid processing	9.1

With regard to the systems being modelled, 45.5% of users model only manufacturing systems, 36.4% model both manufacturing and other types of system, whilst 18.1% of users are involved in modelling any other types of systems.

Concerning the success of modelling, 27.3% of participants report that they have been able to model desirable features of the systems, 54.5% have managed to model the majority of the features, whilst 18.2% had problems in modelling because of software limitations and inflexibility.

Tables 6.7 and 6.8, summarize the responses concerning the main limitations and weaknesses of the software used, and the responses regarding the most important positive features of the software used, respectively.

Table 6.7 A summary of users' opinion about the main limitations of the software (industry)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Limited flexibility for non standard systems	36.4
- Too slow - Manufacturing bias and terminology problem	27.3
- Inadequate graphics - Expensive - Lack of a support for fluid processing - Lack of support for object oriented concepts - Big models are not understandable	9.1

Table 6.8 A summary of users' opinion about the most important positive features of the software (industry)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Graphics	36.4
- Ease of use - Interactivity	27.3
- Speed to build models - Being menu driven	9.1

Table 6.9 presents a summary of the features that users would like incorporated in the simulation software, and which to their knowledge does not yet exist in the software they use.

Table 6.9 A summary of users' opinion about the features that should be included in simulation software (industry)

SOFTWARE FEATURES	PERCENTAGE OF USERS (%)
- Dedicated systems to more specific applications - Higher execution speed	18.1
- CAD links - Improved editing facilities - Removal of unnecessary constraints - Enhancement of fluid processing facilities - Automatic generation of entity cycle diagrams	9.1

6.2.4. Findings

The above presented results of the survey show that there are both similarities and differences in the responses obtained from the two different groups of users. Concerning the type of software being used, users that use only simulation languages are in a minority for both groups of users. The percentage of users that use both simulators and simulation languages is quite even for academic users. The explanation for this might be that almost half (48.3%) of these users use more than one simulation software tool (some of them are even using six different simulation packages), combining education, research and real life projects (Hlupić and Paul, 1993f).

On the other hand, users in industry are much more oriented to using simulators. It is believed that the main reasons for this are a deliberate sampling of users of simulators, and the fact that all the users from industry (100%) who participated in the survey use only one software tool for simulation. In addition, industrial companies usually have to pay the full price of the package, whilst the majority of software vendors offer educational discounts to universities.

Regarding the simulation purpose, it is interesting to note that the majority of users at universities (69%) use simulation both for education and modelling real systems, which indicates that many of the academic participants in the survey are involved in research and work on real life projects. Those that are involved only in modelling real systems are probably those doing only research and not teaching. On the other hand, the percentage of academics that are involved only in education (at least concerning simulation) is relatively low (10.3%), which supports the point concerning the diversity of activities performed in an academic environment.

As expected, a vast majority of users in industry use simulation for modelling real systems, a small proportion of them are involved both in modelling real systems and education, and none of them are involved only in education.

Figure 6.1 shows a comparison of responses obtained for different groups of users regarding the type of software used, whilst Figure 6.2 shows the results obtained regarding the purpose of simulation.

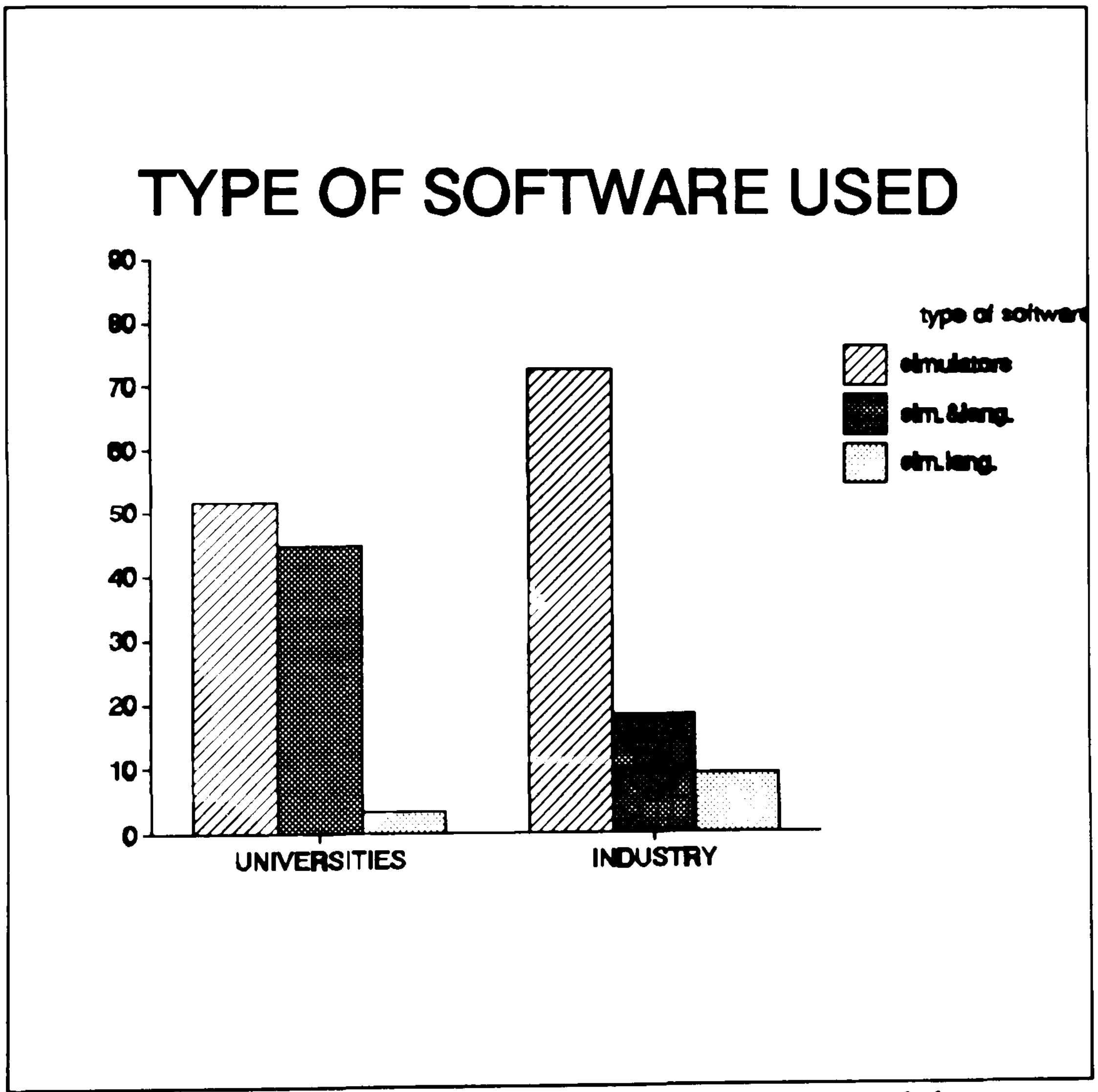


Figure 6.1 Comparison of the type of software used by survey participants

Analysis of the open-ended questions regarding general opinions about the software used, (positive, negative and desirable software features) reveals that users in universities have listed the features that could be expected from users in industry. Many of these features actually correspond to those listed by users in industrial companies. The main reason for this may be the involvement of the majority of academics in modelling real systems in addition to teaching.

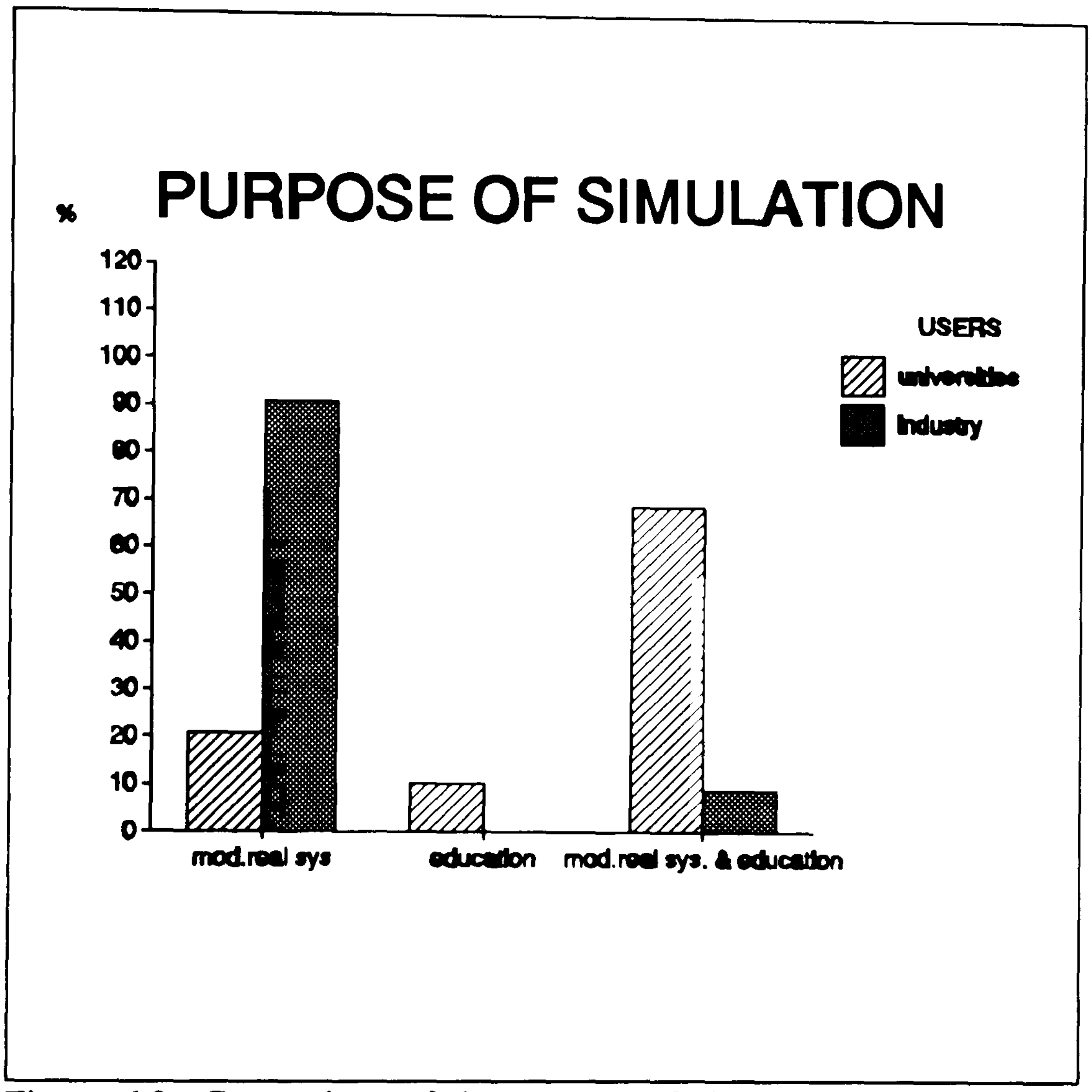


Figure 6.2 Comparison of the purpose of simulation performed by survey participants

Concerning general opinions about the software used, the main objection is that this software is too limited for complex problems (academics) and non-standard problems (users in industry). The majority of users in industry generally have positive opinions about the software they use, favouring the interactivity and graphical features of simulation software, but are not satisfied with the running speed. Ease of use is more approved by users at universities, whilst both groups agree that the software they are using is biased towards manufacturing problems.

Analysis of the main weaknesses listed exposes the main limitation for both types of users as the limited flexibility of the software being used. Academic users are more aware of validation difficulties, a lack of facilities for output analysis and experimental design, whilst both groups agreed that the software is too slow. None of the users in industry considers the price of the software to be a problem, and similarly, only a small percentage of users in education consider simulation software to be expensive. The reason for this might be the fact that the price is not a problem for those who are already using the software.

Regarding the most important positive features of the software being used, it is notable that a majority of participants in both groups have specified graphics as the most beneficial software feature (Hlupić and Paul, 1993g). The second best feature for both groups is ease of use. Academics are aware of ease of learning, automatic report generation and good user interfaces. On the other hand, not too many of them consider flexibility, statistical facilities, documentation, modelling speed or software compatibility to be either of good quality or distinctive advantages of the software they use. Modelling speed is also listed by very few users in industry.

An examination of the features that users would like to be incorporated in simulation software shows that better software compatibility is the most important feature for the majority of academics. Within this feature, a linkage to databases appears to be the most needed, and then follows a linkage to spreadsheet software, and then a linkage to other types of software. Further important features specified by this group of users include a facility for output analysis, more flexibility and experimental design. Some of these users have requested features such as an improvement of documentation and easier model editing, whilst not many of them have demanded features such as the ability to create run-time applications, automatic save, hierarchical model building or lower costs of the software. Users in industry want an improvement in the execution speed and more systems dedicated to specific applications. Some of them require features such as CAD links, improved editing facilities, or removal of unnecessary constraints.

A general analysis of all the results obtained shows that simulation software currently being used by all participants in this survey is predominantly easy to use, visual, interactive, but too limited for complex and non-standard problems, too slow and biased to manufacturing problems. In addition, there are a variety of features that users have requested that refer to better software compatibility, more flexibility and more systems dedicated for specific applications (which is actually contradictory), a provision of facilities for output analysis and experimental design, and better modelling assistance (eg. easier editing and better on-line help).

The obtained results further indicate that users prefer using data driven simulators instead of doing bespoke programming. However, the majority of them would like these simulators to be more flexible and improved, with additional features that would make modelling easier and faster.

6.3. A METHODOLOGY FOR SELECTING A MANUFACTURING SIMULATOR

This section provides a methodology for selecting manufacturing simulators. This methodology has been derived from all findings gained during this research. Since the main subject of research is manufacturing simulators, the methodology for software selection has been established and described from the perspective of this type of simulation software. Nevertheless, the guidelines presented can be also used for selection of other types of simulation software (general or special purpose). In this case, a part of the evaluation framework (derived in Chapter 4) which is specific to manufacturing oriented simulation software will have to be abandoned.

Consideration of the terms 'method' and 'methodology' in the context of software selection is presented in sub-section 6.3.1, whilst sub-section 6.3.2 provides a proposed methodology for selecting manufacturing simulators.

6.3.1. Method and Methodology

There are numerous definitions of the terms 'method' and 'methodology'. For example, a definition provided in Cornford (1992) defines a 'method' as "a description of a specific technique in some symbolic language such that it can be communicated, taught or become an aspect of standard practice". According to the Oxford Advanced Learner's Dictionary (1989), "method is a way of doing something" or "orderly arrangement".

Methodology consists of stages (sets of tasks for generating intermediate results) and tasks (items of work within stages, with a defined deliverable) (Cornford, 1992). Another definition states that methodology is "set of methods used in doing something"

(Oxford Advanced Learner's Dictionary, 1989). To summarize, a methodology provides "a very useful distinction between what is to be done next, who is to do it, and how" (Avgerou and Cornford, 1993). In the context of this research, a proposed methodology for selecting simulation software represents a structured set of stages and tasks that have to be carried out in order to select adequate simulation software.

6.3.2. Proposed Methodology for Selecting a Manufacturing Simulator

On the basis of all findings gained during this research, a structured approach to simulation software selection was derived and is presented in this sub-section. The main stages and elements of the proposed methodology are presented in Figure 6.3. A more detailed explication of tasks to be performed within each stage, a description of the main elements to be considered within each stage, as well as a specification of intermediate results are subsequently provided.

Once a need for purchasing simulation software has been established, several factors have to be initially considered. These factors include the intended simulation purpose, the existing constraints within the company, the main types of models to be simulated, and information regarding the modellers and potential users.

With regard to the intended simulation purpose, it should be decided whether the simulation software is going to be used for education, 'quick and dirty' modelling in industry or for complex/detailed modelling in industry, and/or research. In the case that software is to be used for several different purposes, the most demanding purpose should be chosen as the basis for software evaluation. For example, if software is to be used both for education and research, then software features that are essential for research should be requested. Once the intended software purpose is defined, the appropriate hierarchy of evaluation criteria (see Chapter 4) can be chosen.

It is likely that some organizational constraints would be imposed. Financial constraints might include hardware available for use of simulation software, and the budget available for software purchasing, installation and maintenance costs, purchasing additional hardware, training of personnel etc. Another constraint is time available for software evaluation, selection and implementation.

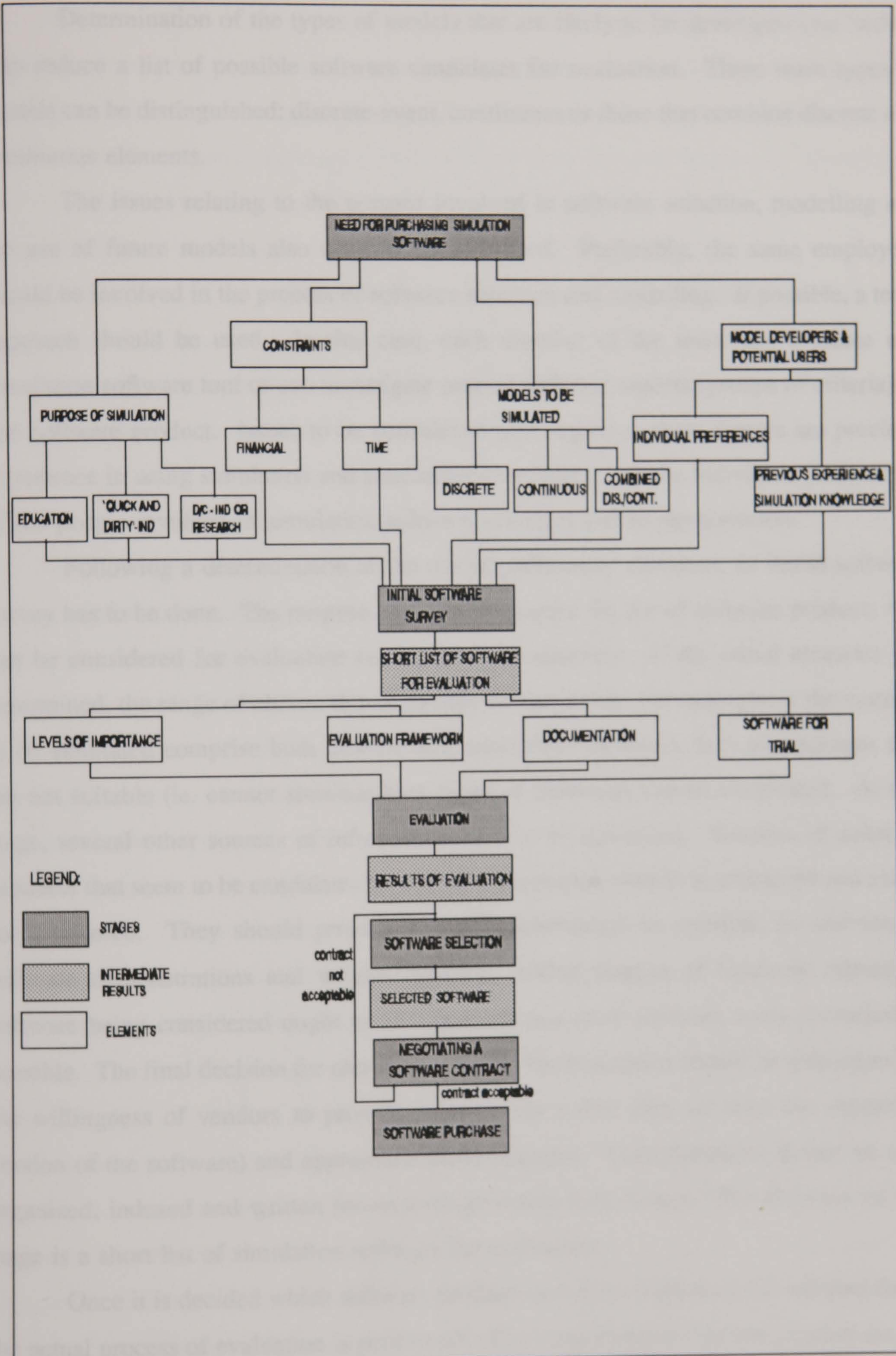


Figure 6.3 The methodology for software selection

Determination of the types of models that are likely to be developed can further help reduce a list of possible software candidates for evaluation. Three main types of models can be distinguished: discrete-event, continuous or those that combine discrete and continuous elements.

The issues relating to the persons involved in software selection, modelling and the use of future models also need to be addressed. Preferably, the same employees should be involved in the process of software selection and modelling. If possible, a team approach should be used. In this case, each member of the team can evaluate one simulation software tool or can investigate several different aspects (groups of criteria) of one software product. Issues to be considered with regard to these factors are previous experience in using simulation and simulation knowledge, and the individual preferences of the people involved in simulation software selection and implementation.

Following a determination of the above preliminary elements, an initial software survey has to be done. The purpose of this is to shorten the list of software products that can be considered for evaluation and subsequent selection. If the initial elements are determined, the range of choice should already be narrowed. For example, if the systems to be simulated comprise both discrete and continuous elements, then all packages that are not suitable (ie. cannot simulate both types of elements) can be eliminated. At this stage, several other sources of information have to be consulted. Vendors of software products that seem to be candidates for software evaluation should be contacted and asked for assistance. They should provide as much information as possible, in addition to software demonstrations and written material. Other sources of literature related to software being considered ought to be examined and other software users contacted, if possible. The final decision for choosing software for evaluation should be influenced by the willingness of vendors to provide software for a free trial (at least the simplified version of the software) and appropriate documentation. Documentation should be well organized, indexed and written for an average non-technical user. The outcome of this stage is a short list of simulation software for evaluation.

Once it is decided which software products are to be evaluated and selected from, the actual process of evaluation is performed. The main elements for this process are the evaluation framework with the appropriate hierarchy of criteria (derived in Chapter 4), the software to be evaluated, and documentation. It is advisable to first examine the most important software features according to purpose, and after that investigate additional features within groups of criteria, according to available time and preferences. The importance of contacting other users of a particular software is emphasized in this phase. Software vendors should agree to provide some names for user reference. It might be useful to visit several users involved in similar types of business or production, and ask them about their experience and opinion about the software, documentation, discovered bugs, vendor support etc. The least valuable are users from companies that have only recently purchased software, and have not used it extensively.

It is desirable to develop a preliminary model that is typical for the intended software purpose and type of systems to be modelled. This practical work is valuable for evaluation, because it gives an impression about the software and actually tests software facilities (it is not impossible that certain features work on paper but not in practice). At a certain stage of evaluation, after some notes are made and it is clear which features are important and which additional information is needed, it might be useful to meet a vendor representative. This ought to enable a discussion of present and forthcoming features of the particular software with somebody who should have an adequate level of technical expertise.

All the above mentioned actions related to software evaluation should at the end produce credible evaluation results. On the basis of these results, it should be decided which software seems to be the most appropriate and which are suitable alternatives (if there are any). In an ideal situation, it should be possible to purchase more than one software product for particular purpose. In this case, software tools should be chosen on the basis of evaluation results.

Once the software is selected, the next step is to negotiate a software contract acceptable by both parties. In general, the contract should specify what products and services are provided, where and when they may be used, how the licence may be transferred to other parties, and how long the product may be used (Gray, 1987). Further issues to be addressed include the number of licences, price of each licence, quantity discounts, educational discounts, cost of training, consulting fees, terms of payment, the ownership and licence terms, support, maintenance, software revisions, documentation, penalties for non-conformance or non-delivery, vendor's response in case of bugs etc. The basic rule is to precisely specify dates and obligations to avoid any future misunderstandings.

If an acceptable contract is acquired, software can be purchased and implemented. In the case where a suitable agreement cannot be achieved, nor an adequate level of support secured, the subsequent best alternative should be chosen on the basis of evaluation results.

It is believed that the above proposal for a simulation software selection methodology is more comprehensive than those few found in the literature (see Chapter 2). It is also believed that this methodology together with an evaluation framework derived in Chapter 4 can be of practical use for any industrial company or educational institution planning to purchase simulation software. Although the guidelines provided are especially derived for a selection of software for manufacturing simulation, they can be also used when other types of simulation software are considered.

There are obviously many diversities in real manufacturing environments and companies and some of the resources or elements identified here might not be available for software evaluation and selection. However, the methodology proposed here presumes circumstances which should be not impossible to achieve and make software selection more structured and hence successful (Hlupić and Paul, 1993h).

6.4. IMPROVEMENT PROPOSALS FOR MANUFACTURING SIMULATORS

This section addresses possible ways for improving manufacturing simulators. Information derived here is based on the overall experience obtained during this research and on the results of the survey presented in section 6.2.

The proposals for improvement relate both to the features that should be implemented and the existing features that should be enhanced in manufacturing simulators. These features were specified in the context of groups of criteria derived in Chapter 4, and from the perspective of software purpose. For each software purpose, the most important groups of criteria that need improvement are indicated, and within each group specific software features that have to be implemented and/or improved are specified. Although the improvement proposals are derived mainly for manufacturing simulators, many of these proposals could be also applied to other types of simulation software.

Despite all similarities between manufacturing simulators, they are intrinsically different and unique. Therefore, it cannot be expected that every improvement proposal can be applied to any software product. These proposals reflect general requirements and deficiencies, and for each individual simulator it should be determined which features should be and can be improved.

Since it was possible to analyze only a limited number of simulators in this research, and as the existing simulators are constantly being revised, it was not possible to make a distinction between the features that do not yet exist and those features that need improving. Nevertheless, it is believed that the information provided can be useful for software vendors and developers, who should realize how they can further improve their software products and better satisfy user requirements.

Improvement proposals for manufacturing simulators used for education are presented in sub-section 6.4.1. Proposals for improvement of simulators used for 'quick and dirty' modelling in industry are addressed in sub-section 6.4.2, whilst sub-section 6.4.3 provides recommendations for the improvements of simulators used for complex and detailed modelling in industry and research.

6.4.1. Simulators for Education

This sub-section proposes features to be implemented and/or improved in manufacturing simulators used for education. These features mainly relate to easier learning and more effective using of simulators. For example, more intelligent and less general on-line help should provide guidance at each step of model development. Error messages should not only detect and precisely locate errors, they should also indicate how an error can be corrected. Interactive tutorials should be provided for learning, including numerous examples of modelling specific features with step-by-step guidance. Documentation should be more complete, properly indexed, and should include many examples.

Several groups of criteria were not listed here, because it is assumed that most of the existing simulators perform adequately regarding these features. For example, it was assessed that the visual aspects are generally acceptable. The same applies to coding aspects, as flexibility is not so crucial a feature of simulators used for education. Similarly, software compatibility is usually sufficient as many simulators can integrate with word processors and spreadsheet packages. A complete list of the proposed improvements to manufacturing simulators used for education is provided in Table 6.10.

Table 6.10 Proposals for improvement to manufacturing simulators used for education

GROUPS OF CRITERIA	FEATURES TO BE INCLUDED/IMPROVED
Modelling assistance	- Better prompting - More intelligent and less general on-line help - Automatic editing of data
Testability	- More comprehensive logic checks - Better quality of error messages - Models should be generally easier to debug - It should be possible to display and easily access all logical elements
User support	- Better documentation - Interactive tutorial for learning - More useful demo models
Statistical facilities	- Facilities for output data analysis
Experimentation facilities	- Experimental design capability
Efficiency	- Better reliability (less bugs)
Financial and technical features	- No security device - Less expensive software - Versions of software for network - Free software trials - Frequent and comprehensive updates - Provided maintenance
General features	- Easier to learn and use - More user friendliness - Conceptual model generator
Scheduling features	- More in-built scheduling rules - No routing restrictions

6.4.2. Simulators for 'Quick and Dirty' Modelling in Industry

Features proposed within this sub-section support fast and effective model development, verification and running. For example, characteristics such as better model reusability or effective multitasking, improved prompting and on-line help, more in-built scheduling rules and physical elements can accelerate model development. Better quality of error messages and a backward clock (which could enable the running of the model backwards until a certain error emerges) could decrease the time needed for verification. Experimental design capability and a facility for output data analysis could advance the experimentation.

Groups of criteria that are omitted from the list of improvement proposals are excluded for similar reasons to simulators used for education. The reasons relate to a relatively smaller importance of features regarding robustness, flexibility and the handling of a large amount of data for simulators used for 'quick and dirty' modelling in industry. Table 6.11 provides the improvement proposals for manufacturing simulators used for 'quick and dirty' modelling in industry.

Table 6.11 Proposals for improvement to manufacturing simulators used for 'quick and dirty' modelling in industry

GROUPS OF CRITERIA	FEATURES TO BE INCLUDED/IMPROVED
Efficiency	- Faster model compilation and execution - Better reliability (less bugs) - Better model reusability - Operative multitasking
Modelling assistance	- Better prompting - More intelligent and less general on-line help - Automatic editing of data
Testability	- Comprehensive logic checks - Better quality of error messages - Backward clock - Understandable display of workflow path - Easier debugging
Statistical facilities	- Facility for output data analysis - Variety of in-built statistical distributions
Experimentation facilities	- Facility for experimental design - Easier experimentation
Scheduling features	- More in-built scheduling rules - No routing restrictions - Scheduling optimization

Physical elements	- In addition to different types of machines, a variety of materials handling and storage facilities should be explicitly provided
User support	- Better documentation - Interactive tutorials for learning and/or training courses - Easily accessible help-line
General manufacturing modelling features	- Easier explicit modelling of specific operations such as fixturing, palletization, containerization etc.
Financial and technical features	- Less expensive software - Free software trials - No security device - Frequent and comprehensive updates - Low maintenance costs
General features	- More user friendliness - Easier to use
Manufacturing performance	- Easier specification of user defined reports
Input/Output	- Summary reports for multiple runs - More and better dynamic graphical outputs

6.4.3. Simulators for Complex/Detailed Modelling in Industry and Research

This sub-section provides improvement proposals for simulators used for complex/detailed modelling in industry and research. The list of recommended improvements is the most comprehensive for this software purpose. This indicates that the present limitations of manufacturing simulators significantly affect the users involved in this type of modelling.

Several software features that especially need improvement can be distinguished. These features include more flexibility (achievable by the improvement of characteristics relating to coding aspects and efficiency), better software compatibility (there is a need for integration with DBMS, CAD software, statistical packages etc.), and more assistance in the design of experiments, output data analysis and debugging. There are also several other aspects which might be enhanced and make manufacturing simulators more productive, as listed in Table 6.12.

Table 6.12 Proposals for improvement to manufacturing simulators used for complex/detailed modelling in industry and research

GROUPS OF CRITERIA	FEATURES TO BE INCLUDED/IMPROVED
Efficiency	- More robustness (flexibility) - Better reliability (less bugs) - Faster model compilation and execution - It should be possible to model more details - No restriction on number of elements in model apart from memory limitations - Better model reusability - Automatic saving - Effective multitasking - Variety of in-built queuing policies
Software compatibility	- Integration with DBMS - Integration with CAD software - Integration with statistical packages - Integration with MRP II software - Integration with scheduling software - Integration with expert systems
Coding aspects	- More programming flexibility - More comprehensive internal languages - Easier link to lower level languages - Better support of programming concepts - More in-built functions, system variables and attributes etc. - Easier manipulation with code
Modelling assistance	- Better prompting - More intelligent and specific on-line help - Automatic editing of data - Facility for writing documentation notes concurrently with model development
Testability	- Better and more comprehensive logic checks - Backward clock - More useful error messages on data entry - Display and access to all logical elements - Easier debugging
Statistical facilities	- Facility for output data analysis - Variety of in-built statistical distributions and random number streams
Experimentation facilities	- More help in experimental design - Better experimentation facilities - Automatic determination of run length

Financial and technical features	- Less expensive software - Free software trials - No security device - Quantity discounts - Frequent and comprehensive updates - Low maintenance costs
User support	- Better documentation - Appropriate training course - Easily accessible help-line - Consultancy provided by supplier
Scheduling features	- More in-built flexible scheduling rules - No routing restrictions - Ability to push/pull parts from the specific positions within elements - Variety of in-built vehicle scheduling strategies - Scheduling optimization
Manufacturing performance	- More scheduling related outputs - Better and easier due dates monitoring - Easier specification of user-defined outputs
Input/Output	- Better quality of output reports - Summary report for multiple runs
General manufacturing modelling features	- Easier explicit modelling of specific operations such as fixturing, palletization, fluid composition etc.
General features	- Hierarchical model building - Run-time applications

6.6. SUMMARY

This chapter contributes the final findings of this research. It provides an insight into users' requirements for simulation software, based on the survey conducted. This survey has included a number of simulation professionals in academic and industrial environments. A general conclusion to be drawn from the survey findings is that simulation software users regard these software products as easy to use and they appreciate their visual interactive features. On the other hand, simulators are too limited for complex and non-standard problems, and too slow and biased towards manufacturing problems. Features to be improved (as users have requested) mainly relate to better software compatibility, more flexibility and a provision of facilities for output analysis and experimental design.

The next part of this chapter provided a methodology for the selection of a manufacturing simulator. This consists of pragmatic guidelines related to the actions to be taken and factors to be considered during the process of evaluation and selection of software for manufacturing simulation.

Finally, proposals for improving manufacturing simulators are given, according to the software purpose. The main groups of criteria together with specific features within each group that need improving are specified for each software purpose. Table 6.13 provides a summary of groups of criteria that need improvement, specified for different software purposes.

Table 6.13 A summary of groups of criteria that need improvement specified for different software purposes

EDUCATION	'Q & D' - INDUSTRY	C/D - INDUSTRY AND RESEARCH
Modelling assistance	Efficiency	Efficiency
Testability	Modelling assistance	Software compatibility
User support	Testability	Coding aspects
Statistical facilities	Statistical facilities	Modelling assistance
Experiment. facilities	Experiment. facilities	Testability
Efficiency	Scheduling features	Statistical facilities
Financial and technical features	Physical elements	Experiment. facilities
General features	User support	Financial and technical features
Scheduling features	General manufacturing modelling features	User support

________	Financial and technical features	Scheduling features
________	General features	Manufacturing performance
________	Manufacturing performance	Input/Output
________	Input/output	General manufacturing modelling features
________	________	General features

It is evident that the list of features that need improvement is the longest for simulators used for complex/detailed modelling in industry and research, and it is the shortest for simulators used for education. This indicates which types of simulators especially need further development. This development should concentrate on achieving more flexibility and software compatibility.

There are also improvement proposals that are similar for all software purposes. For example, more reliable software is needed with less bugs as well as more assistance in modelling, debugging, experimentation and output analysis. Better documentation should be provided, security devices eliminated, and software prices should be lowered (the support for the claim relating to price can be found in Pidd (1989), Davis and Williams (1993) and Christy and Watson (1983)).

The ideas presented in this chapter can be of practical use to users, and simulation software suppliers and developers. Users might apply the guidelines for software selection together with the evaluation framework derived in Chapter 4. This structured approach should provide a more efficient and cost effective procedure for simulation software selection. On the other hand, software vendors and developers might utilize the information gained from the users' survey and from the proposed improvements to plan further software developments. As a consequence of this, simulation should become more effective and widespread.

CHAPTER 7. SUMMARY AND CONCLUSIONS

This chapter provides a recapitulation of the results and findings of this thesis. It draws out the major conclusions of this thesis and discusses their relevance to simulation modelling software approaches to manufacturing problems. Finally, it addresses the possibilities for future research.

Section 7.1 provides a summary of this thesis. Conclusions based on the findings of the thesis are drawn in section 7.2, whilst section 7.3 presents the lines for future research.

7.1. SUMMARY

This thesis investigates simulation modelling software approaches to manufacturing problems. In particular, it addresses the issues related to evaluation, selection, and possible way of improving manufacturing simulators. The final aim of this research is to discover how simulation software users' requirements can be better fulfilled. As a results of this, simulation modelling should become easier and more accepted.

Chapter 1 introduces the basic issues related to research presented in this thesis. It provides essential information regarding simulation modelling, advanced manufacturing systems, simulation software, and the case study carried out. Finally, it establishes the objectives of this thesis.

Chapter 2 deals with background research material, used as a basis for the research presented in this thesis. It presents a number of research studies regarding the use of simulation in advanced manufacturing environments, and especially studies related to simulation software evaluation and selection. A critical analysis of the presented research studies is provided. On the basis of this analysis, conclusions are drawn. These conclusions further justify the objectives of this research.

Chapter 3 presents the case study carried out at BICC-VERO Electronics, Eastleigh. The study relates to simulation modelling of an automated system for the electrostatic powder coating of metal components. The main purpose of this study was to analyze several widely used manufacturing simulators by modelling a real manufacturing system. A conceptual model of the powder coating system was developed using an activity cycle diagram. A comprehensive computer model was developed using the WITNESS manufacturing simulator. This model was used for experimentation in order to discover how the throughput of the system can be improved. Other, less detailed models (the SIMFACTORY II.5, XCELL+ and ProModelPC models) were also developed, mainly for the purpose of software analysis.

Chapter 4 establishes a comprehensive simulation software evaluation framework. This framework comprises more than 310 criteria, which are especially derived for the evaluation of simulation packages used in manufacturing environments. The evaluation criteria are grouped according to their nature. A possible use of the evaluation criteria has been discussed separately for users at universities and users in industry. Different hierarchies of the criteria were derived expressing the relevance of certain criteria for particular software purposes: education, 'quick and dirty' modelling in industry, and complex/detailed modelling in industry.

Chapter 5 presents an evaluation of the manufacturing simulators used for the modelling in the case study. It was not intended to discover 'the best' simulator, because every software evaluation quickly becomes obsolete due to continuous software revisions. The main purpose of this evaluation was to demonstrate the use of the evaluation framework derived in Chapter 4, and to determine the suitability of these simulators for particular purposes. The general quality of each group of criteria was ranked for each simulator. On the basis of this ranking, the evaluated simulators were compared, and their suitability for particular purposes determined.

Chapter 6 provides a final contribution to this thesis. It considers manufacturing simulators from the users' perspective and analyzes which features are needed, how these software products can be improved, and how the user can select an adequate simulator. The main findings of the survey carried out to discover users' views about simulation software are presented. A methodology for software selection is derived on the basis of all the previous research findings. Finally, improvements to manufacturing simulators are proposed according to their purposes. It is believed that these research findings are of practical use both for simulation software users and vendors, and could result in more effective simulation software selection and utilization.

7.2. CONCLUSIONS

This thesis has investigated the issues related to evaluation, selection and improvement of simulation software used in manufacturing environments. Several data driven manufacturing simulators have been analyzed on the basis of the case study carried out at BICC-VERO Electronics in Eastleigh. These software products were investigated from the users' perspective. Because of this, a 'black box' approach was utilized, which means that the main emphasis of the research was on software features accessible to users, rather than on the internal structure of these simulators.

In this context, the ultimate aim of this research was to derive results that can be of practical use both for the users of simulation software, and for software vendors and developers. As such, these results can be considered as the main contribution to research in this area. Simulation software users should benefit from the comprehensive evaluation framework and software selection methodology derived in this thesis. Since "the evaluation criteria change with each report" (Chikofsky *et al*, 1992), it is believed that such a comprehensive evaluation framework can be used as the basis for standardization of simulation software evaluation.

The need for standardization of simulation software selection is also evident. "Many organizations choose a tool or tool set without establishing formal evaluation criteria or thoroughly examining tools. Instead, they frequently base their decision on highly visible attributes such as documentation or look and feel, rather than on quality and support of a specific method. It's now time to find ways to consistently, objectively evaluate a tool's utility and appropriateness" (Chikofsky *et al*, 1992). The software selection methodology derived in this thesis is a research contribution towards this standardization. The final outcome of this approach should be a more efficient and cost effective selection of adequate simulation software products.

Finally, simulation software vendors and developers should find useful both the results of the users' survey and proposals for improving manufacturing simulators derived in Chapter 6. The results of the survey have provided users' opinions about simulation software, with an emphasis on possibilities for better fulfilment of users' requirements. This should be particularly useful for software developers, who should realize that there are still many problems associated with simulation software. The results also show that the claim that a particular software can quickly and easily develop a simulation model of any complexity is not valid.

Improvement proposals for manufacturing simulators indicate which areas of further software developments are especially important for particular software purposes. The improvement proposals are general, which means that it has to be determined which proposals are the most applicable for each particular simulator.

The research conducted in this thesis has lead to the derivation of many findings. These findings are summarized below:

(i) An analysis of the background research material in conjunction with practical experience obtained through the case study and previous research in simulation, reveals that despite all advances in hardware and simulation software, the time taken to develop and test simulation models of reasonable complexity restricts the use and applicability of simulation.

(ii) Most publications on the use of simulation in manufacturing environments reflect the views of users whose expertise in one software product is difficult to relate to publications that demonstrate expertise in another software product.

(iii) Many reports on simulation software packages are written by its vendors, and as such lack criticism. On the other hand, independent simulation users usually describe only the successful parts of simulation studies.

(iv) None of the studies analyzed provides an extensive list of simulation software evaluation criteria. This finding supports the usability and credibility of the evaluation framework derived in this research.

(v) Evaluations and comparisons of WITNESS, SIMFACTORY II.5, XCELL+ and ProModelPC have been given in some previous studies, none of which have been as comprehensive as the one provided in this thesis.

(vi) None of the studies concentrate on a software selection methodology, which further justifies the need for a methodology as derived in this research.

(vii) The experience obtained from the case study carried out revealed that regardless of the current advances in data-driven simulators, they are still not capable of handling the full complexity of advanced manufacturing systems.

(viii) When data-driven manufacturing simulators were used for modelling the powder coating system, several software limitations were discovered and problems experienced. This supports the need for further improvements to these software products. It also confirms that simulation languages are nowhere near obsolete.

(ix) Even the most flexible simulators could not properly model some features of the case study system, such as commencing the setup before parts arrive at the machine.

(x) On the other hand, modelling time was significantly reduced by using a simulator, compared to the time that would have been needed in the case of bespoke programming using a simulation language.

(xi) With regard to the applicability of activity cycle diagrams for conceptual modelling of complex manufacturing systems, it is evident that this method, despite its advantages, quickly becomes too limited as the complexity of the systems rise.

(xii) The criteria included in the evaluation framework are classified in two main groups: general criteria, and criteria specific to the evaluation of manufacturing oriented simulation software. General criteria were mapped to phases of the simulation process, which showed that criteria representing certain features of simulation software can significantly influence each phase of this process.

(xiii) Establishing a hierarchy of evaluation criteria revealed that different criteria have different levels of importance for different software purposes.

(xiv) With the proposed 5 levels of importance, none of the groups of criteria for any software purpose appeared to be irrelevant.

(xv) The number of the groups of criteria that gained the highest level of importance is the greatest for software to be used for complex/detailed modelling in industry.

(xvi) When a package to be used for education is evaluated, then the most important criteria are those that support learning and relatively quick and easy model development rather than those that enable the handling of a large quantity of data and detailed modelling.

(xvii) Similarly, for a package to be used for 'quick and dirty' modelling in industry the most important criteria relate to modelling assistance for easy model development, and saving modelling time.

(xviii) On the other hand, the most relevant criteria for a package to be used for complex/detailed modelling in industry and/or for research are those relating to flexibility, software compatibility, efficiency, testability, and experimentation and statistical facilities.

(xix) Evaluation and comparison of WITNESS, SIMFACTORY II.5, XCELL+ and ProModelPC revealed that although some of these simulators had a better overall performance than the others, they did not perform equally well for all groups of evaluation criteria.

(xx) These evaluation results further indicated that there is no simulator which is equally good for all purposes.

(xxi) Analysis of the suitability of evaluated simulators for particular purposes revealed that XCELL+ can be considered as the most appropriate for education because of its simplicity, ease of learning and use, and short modelling time.

(xxii) It has also been estimated that SIMFACTORY II.5 could be the most suitable for 'quick and dirty' modelling in industry due to its ease of use, and straightforward modelling of many features typical of manufacturing systems.

(xxiii) WITNESS was regarded as the most appropriate simulator for complex/detailed modelling in industry and research, because of its comprehensiveness and relative flexibility.

(xxiv) A general conclusion to be drawn from the results of the survey carried out is that simulation software users consider these software products as easy to use, with good visual interactive features.

(xxv) At the same time, users find simulation software too limited for complex and non-standard problems, too slow, and biased towards manufacturing problems.

(xxvi) Most of the users have requested the following software features to be improved: software compatibility, better flexibility, and support for simulation output analysis and experimental design.

(xxvii)A derived methodology for simulation software selection should result, when appropriately applied, in more structured and successful selection of this type of software.

(xxviii)With regard to the proposals for improvement of manufacturing simulators according to their purpose, it has been discovered that the list of features that need improvement is longest for simulators used for complex/detailed modelling in industry and research, whilst it is shortest for simulators used in education.

(xxix) Several improvement proposals are similar for all software purposes, such as: better software reliability; more assistance in modelling, debugging, experimentation and output analysis; better documentation; elimination of security devices; and lower software prices.

(xxx) A general finding about current simulators, based on the overall simulation experience, on the analysis of the literature, on the case study, and on the survey conducted, reveals that simulators should be more flexible, compatible with other types of software and more reliable.

(xxxi) Furthermore, there is a lack of match of the software to the problems they are supposed to be able to handle. Current simulators are not capable of handling a variety of problems that occur in real manufacturing systems, and they will never be able to fit all problems in the manufacturing domain.

(xxxii)The results of this research, in particular the case study and the survey findings, confirmed the pre-assumptions made for the simulation process, simulation software, and software evaluation in general.

(xxxiii)A final conclusion to be drawn from this thesis is that further improvement and research in simulation software is needed, in order to reduce the time spent on any simulation study, and to increase the usefulness, and effectiveness of simulation in general.

Of these findings, the main contributions to knowledge of this thesis are the determination of a structured set of evaluation criteria for selecting simulation software for a manufacturing applications, combined with a methodology for undertaking the selection. Proposals for improving current software approaches are also made.

7.3. FUTURE WORK

All research can be further expanded, and the research presented in thesis is no exception. There are several possibilities for future research, such as:

(i) Using other case studies in real manufacturing environments, which should include the simulation modelling of a variety of different advanced manufacturing systems. This should enable the testing of simulation software on a wide range of problems and logical features.

(ii) Analysis of other simulation packages, and especially other data-driven manufacturing simulators such as: AutoMod II (a simulation package with a comprehensive set of material handling modules, which is supplied by AutoSimulations Inc.); INSTRATA (a manufacturing simulator based on the GENETIK simulation environment, supplied by the Insight Logistic); and ARENA (a recently released manufacturing simulator based on SIMAN/CINEMA, and supplied by the CIMULATION CENTRE).

(iii) Testing of other software products could probably result in a further expansion of the evaluation framework derived in this thesis.

(iv) Findings produced in this thesis should be further tested in practice. The first step should comprise visits to various industrial companies, which do not use simulation, but plan to do so. In that case, a particular company will have to choose an adequate simulation software product. This should provide an opportunity to test both the evaluation framework and the simulation software selection methodology derived in this thesis.

(v) In addition, Table 5.1, which shows the rating and comparison of simulators evaluated in this research, can be further validated by structural interviews. These interviews should involve other users who have practical experience in using these simulators.

(vi) Another survey should be carried out in order to investigate how industrial companies select simulation software. The results of this survey, together with the results of this research regarding software selection, should provide a basis for an effective standardization of this selection process.

(vii) Visits to industrial companies which already use simulation to interview simulation software users in order to supplement the survey findings.

(viii) Further collaboration with software vendors and developers to reveal which improvement proposals could be implemented in particular software products.

These are some of the possibilities for further research. It can be expected that other research areas and opportunities will emerge in the near future.

APPENDICES

Appendix A. The Methodology of Activity Cycle Diagrams

A.1 ACD Methodology

An activity cycle diagram is a graphical method used for the development of conceptual simulation model. It is particulary useful for systems with a strong queuing structure. The method facilitates the modelling of the interaction among entities. An entity represents a component of the system that has to be modelled, which retains its identity throughout the time it spends in the simulation model. Every entity in the model can either be in an active state, usually engaged with other entities in an activity; or it can be in an idle state - a queue, where the entity is waiting to participate in an activity.

The following conventions have to be considered in drawing activity cycle diagrams:

(i) Each type of entity has a life cycle.
(ii) The cycle consists of activities and queues.
(iii) Activities and queues alternate in the cycle.
(iv) The cycle is closed.
(v) Activities are represented by rectangles, whilst queues are depicted by circles, as shown in Figure A.1.

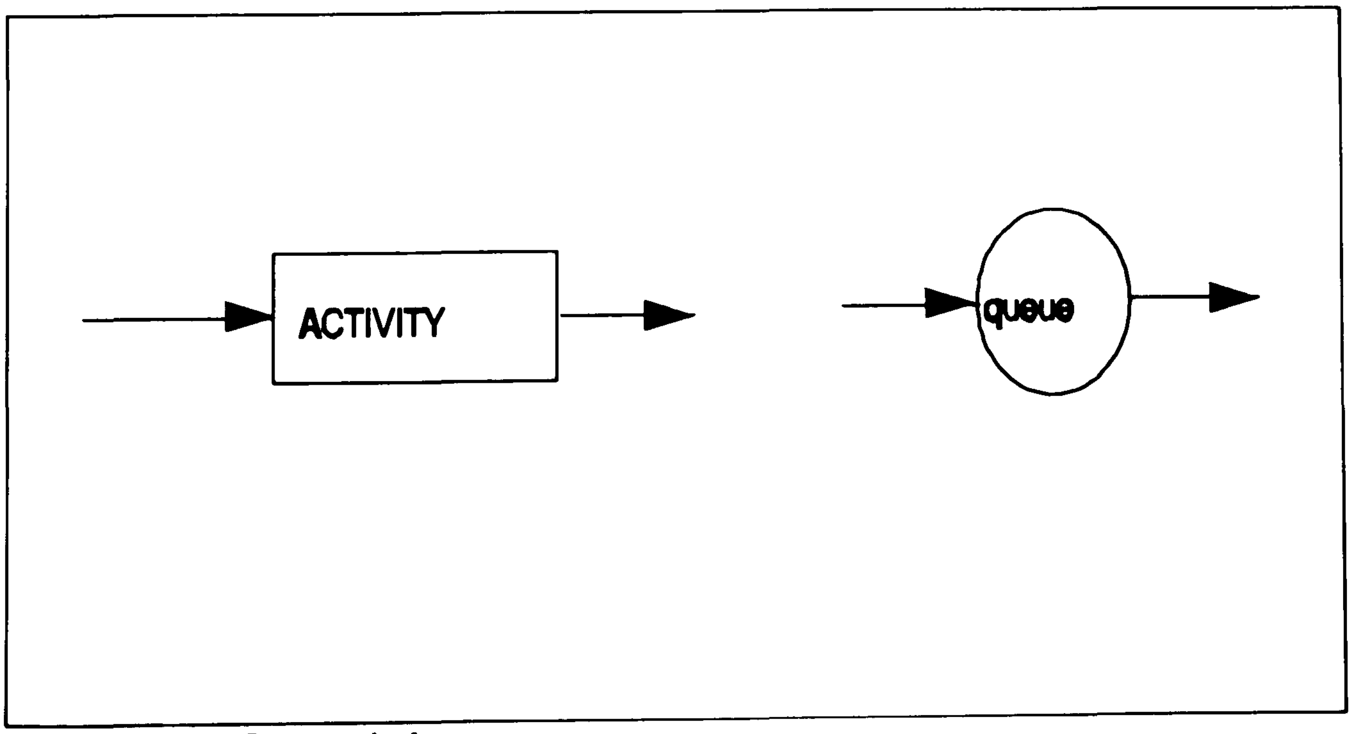


Figure A.1 ACD symbols

A.2 Stages of models development

The process of simulation model development using activity cycle diagrams is carried out through the following stages:

(i) The first stage

First, the relevant entities of the model have to be chosen as well as the appropriate queues and activities. The next step is drawing the entity life cycles as a closed loop of alternate queues and activities.

(ii) The second stage

At this stage, the activity cycle diagram is constructed by linking the life cycles of the entities engaged in the model in one connected diagram. It is also necessary to specify whether the entities are temporary or permanent and how they enter or leave the model. Whilst permanent entities stay in the model throughout the duration of simulation, temporary entities stay in the model only for part of the time of the simulation.

(iii) The third stage

This stage relates to the determination of priorities where entities can be involved in more than one activity, and to an indication of whether entities possess attributes that determine their movements through the model.

Activity cycle diagrams were chosen in this study for simulation model development for the following reasons:

(i) A small number of different graphical symbols enables a simple presentation of systems.

(ii) Manufacturing systems are systems that include numerous serving and queuing places (eg. parts waiting to be loaded on conveyor or painted on the booth), and the activity cycle diagram method is specially suitable for that kind of systems.

(iii) Conceptual models developed using this method can be used as a basis for subsequent development of the computer models with any software package.

Appendix B. Technical Description of WITNESS Model

The main characteristic of the WITNESS model is its substantial level of detail. Many features which were assumed to have an influence on the performance of the system being modelled were included in the model.

The model developed is obviously only an approximation of the real system. Some of its characteristics were omitted due to the software constraints, such as cleaning of the booths when the painting colour is to be changed and selection of batches from the buffer according to the values of attributes. There are also many features that were not considered as relevant for the system's efficiency such as the control system for conveyor and spraying guns, powder recycle unit, extraction filters, equipment for pretreatment and the sources of energy.

The final version of the WITNESS model obtained during the process of gradual model development consists of a number of physical and logical elements. These elements are described below, whilst a summary of the types and quantities of the elements used in the model is provided in Table B.1.

Table B.1 A summary of the WITNESS model elements

ELEMENT	QUANTITY
PHYSICAL ELEMENTS:	
Machines	18
Conveyors	34
Parts	4
Labour	8
Buffers	2
LOGICAL ELEMENTS:	
Functions	25
Attributes	10
Variables	86
Shifts	20
Timeseries	4

User-defined distributions	2
Files	5

B.1 Physical Elements

(i) Machines

Machines are used in the model to represent physical equipment, specific operations performed by labour, and were also used for dummy activities used due to the software requirements. Whilst the quantity of machines representing production equipment corresponds to the number in the real system, the number of machines used to represent operations was chosen according to the amount of labour of a certain type available for specific operation.

Spraying booths are modelled as single machines, because they spray only one flight bar at a time. The manual finish of specific batches coated automatically is performed on separate single machines using labour, to enable conditional routing of batches after automatic coating (either to a station for a manual finish or to a conveyor).

Jigging of the parts and their loading on flight bars is modelled by three assembling machines, which take (using labour) a number of parts according to the attribute representing the number of parts per flight bar for a particular batch, and produce one part which represents jigged parts to be loaded on the flight bar. On the other hand, the unloading activity is modelled by two production machines which take each part from the conveyor after the last processing stage, and produce as many parts as are loaded on the flight bar.

The masking operation is modelled using five batch machines which means that each batch to be masked is equally divided into five groups, and each worker has to mask one part of the batch. Preemption rules are used here, thus although some workers may be busy with masking, they should leave the masking stations and participate in loading or unloading when required.

Finally, there are some dummy machines, used to handle the logic related to masking. For example, when a batch to be masked is moved to the front of the buffer, dummy machines are used to pull this batch from the buffer and to push it to the masking station in the case that all of them are idle. If not, this batch will be pushed to the back of the buffer, because different batches should not be mixed up.

(ii) Conveyors

The system being modelled consists of a closed overhead conveyor chain. Separate conveyors were defined for various sections in the chain. Whilst the length of separate conveyors varies according to the number of flight bars in the sections of the conveyor chain which these conveyors represent, the speed of all conveyors is the same as well as the breakdown pattern.

The approach of modelling separate conveyors was taken for several reasons. First of all, WITNESS does not allow branching of conveyors. They can be defined with a fixed size and represented by a linear display. In addition, there are several points in the chain where a batch can be routed in different directions, depending on part attributes. Such conditional routing was possible to model only by using separate conveyors. In that case, the routing condition was programmed as an output rule for one conveyor, which determines where the batch will be pushed next. The final reason for using a relatively large number of smaller conveyors to model the conveyor chain was to obtain a more realistic graphical display of the model. This has been done in order to improve communication with managers and production engineers and their awareness of the model and simulation in general.

(iii) Parts

Several different part types have been used in the model. One type relates to the raw materials which represents parts to be coated arriving in the buffer (storage area). After coating, parts are changed to 'painted', and after baking they are changed to 'finished', which means that the process of coating is completely finished and the parts represent final products. A special type of parts is used to represent gaps between different batches, with zero parts per flight bar.

(iv) Labour

Different types of labour were modelled, according to the type of work in which they can participate. One worker is dedicated only to painting. Several other participate in jigging and loading of parts. Some of the workers are assigned to unloading of parts from flight bars after the last processing stage. Finally, 'floating' workers can participate in any activity, depending on the current situation in the system. All types of labour can participate in the masking activity, system maintenance or equipment repair in the case of breakdowns.

(v) Buffers

Only two buffers have been used in the model. One to represent a storage area where parts wait to be loaded on a conveyor, and another to temporarily store parts that have been masked until a complete batch is masked and pushed to the storage area to wait for loading.

B.2 Logical Elements

(i) Functions

A variety of functions were written in order to handle the special logical features of the system. For example, for each attribute a separate function was written in order to assign the same values of the attributes to all parts within one batch. Otherwise, each part within the batch would have different values of the same attribute, or the values of attributes could be fixed, which means that all batches will have exactly the same values of attributes.

Some other examples of the use of functions are as follows. Functions were used to determine the capacity and cycle times of loading and unloading stations, and to decide when a new batch can be pulled for loading. Functions were written to decide when parts can be loaded on the first conveyor, and when and how many gaps should be placed on this conveyor.

Several functions were used to handle the logic relating to the capacity of masking stations, routing of batches that require masking to different destinations in order to avoid merging of different batches.

(ii) Attributes

A number of attributes have been defined for parts in order to describe different characteristics of different part types. The main feature of the system being modelled is a large variety (more than two thousand) of different part types with different characteristics. Attributes were also needed in order to enable conditional routing of batches.

Attributes defined in the model relate to the batch number, batch size, colour, batch priority, number of parts per flight bar, masking requirements, the number of the booth on which the batch is to be painted, the total number of flight bars required for a particular batch and the manual finish of complex parts coated automatically.

(iii) Variables

Many variables have been used in the model. They were either used as global variables to monitor the changes in the system (eg. the number of colour changes), or as local variables used in functions. For each function, separate variables have been defined in order to avoid logical errors. The display of some variables was very useful for model verification. For example, the number of flight bars to be loaded by a particular batch was displayed on the screen and it was easy to check whether the model behaved correctly. Some of the attribute values have been assigned to variables displayed, which was especially useful for testing conditional routing.

Another example of the use of variables is a control of the random number streams. The model comprises more than twenty random variables. Each random number stream was assigned to a different variable, and the values of these variables were initialised at the beginning of each experiment.

(iv) Shifts

A relatively complex shift pattern had to be modelled in order to include lunch breaks at different time for different groups of labour, and to include weekly, monthly and yearly maintenance during which the system does not operate. This has resulted in a number of different sub-shifts and main shifts, which were modelled hierarchically to represent daily, weekly, monthly, and half-yearly performance of the system.

(v) Timeseries

Several timeseries have been defined in order to display the performance of the system dynamically. These timeseries, very useful for model verification, showed the total throughput of the system during simulation, labour utilization and utilization of painting booths.

(vi) User-defined distributions

User defined distributions have been used for a small percentage of special part types in order to provide values for the batch size and the number of parts per flight bar. These parts represent small screws, where five hundred of them are jigged together and loaded on one flight bar. This special type of part was defined because it was not possible to fit such extreme values properly into theoretical distributions together with other values.

(vii) Files

Several files were specified into which specific reports were written. The standard written report provided by WITNESS is lengthy and not very understandable. Therefore, special files were used to provide information about the throughput, labour utilization, and machine utilization which was particularly useful when several experiments were run automatically.

The graphical display of the model is presented in two parts, because it was not possible to show the entire model on one screen. Figures B.1 and B.2 show the display of an empty state of the model, whilst figures B.3 and B.4 show the display of the model during experimentation.

Figure B.1 The first part of the WITNESS model before experimentation

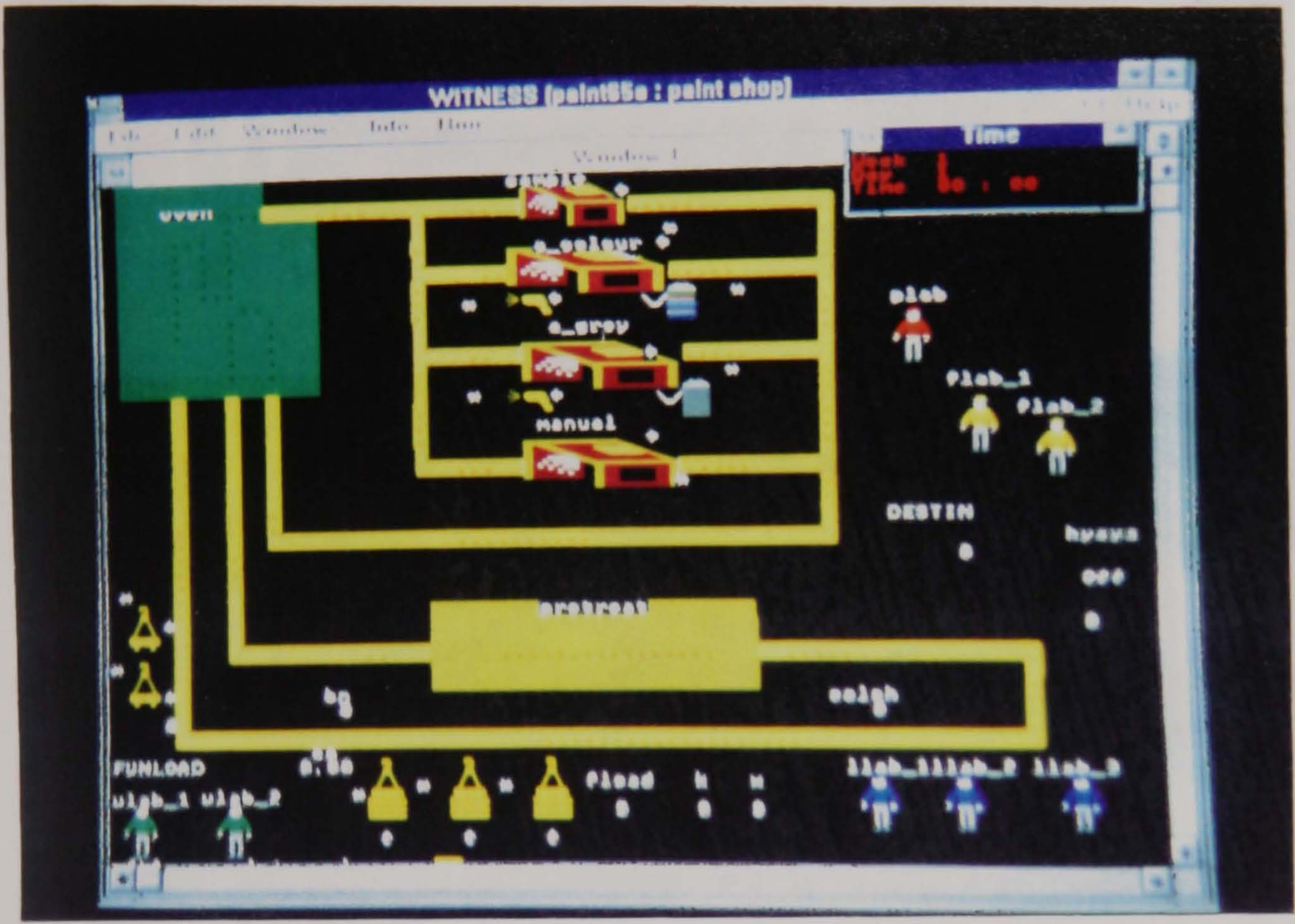


Figure B.2 The second part of the WITNESS model before experimentation

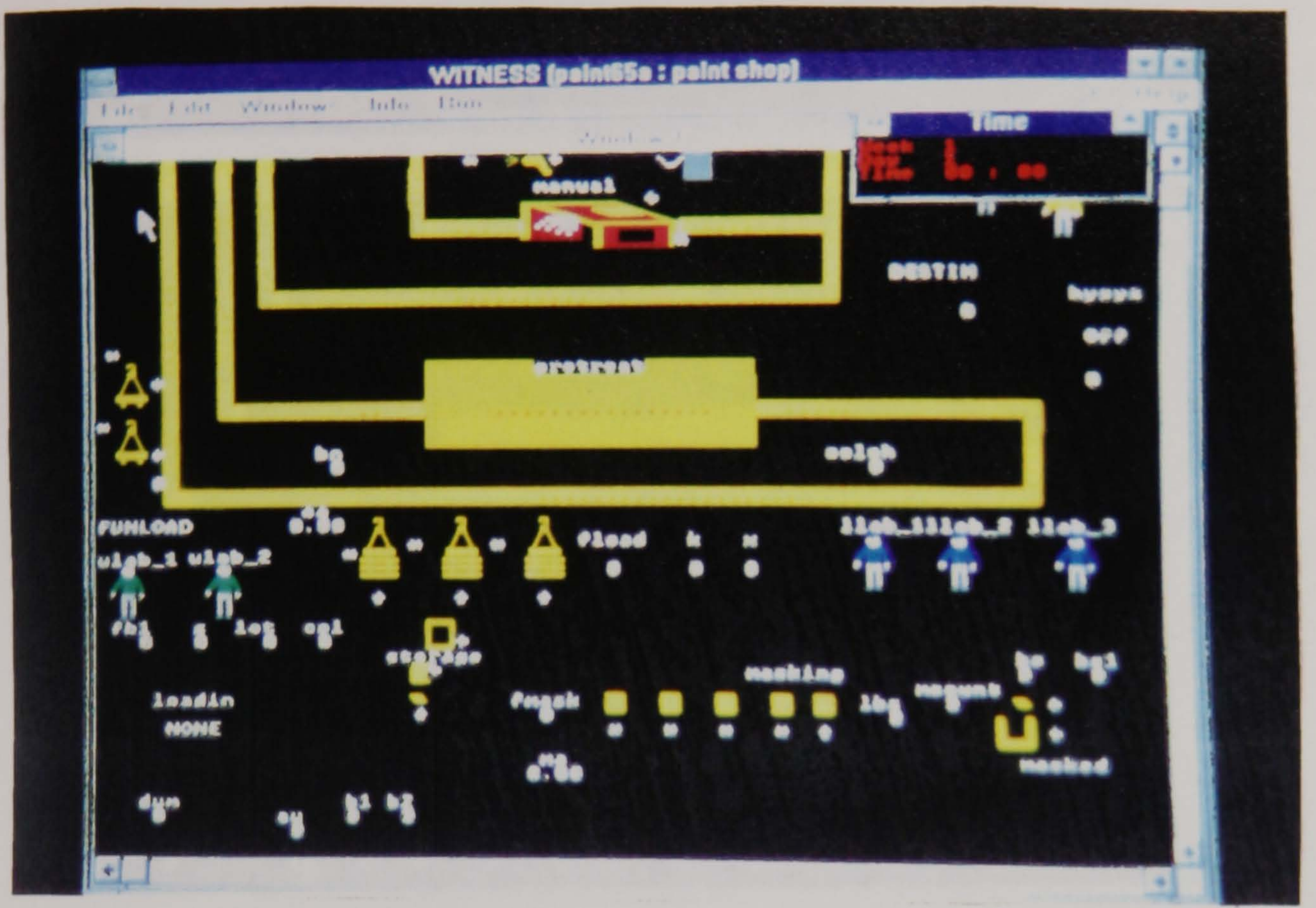


Figure B.3 The first part of the WITNESS model during experimentation

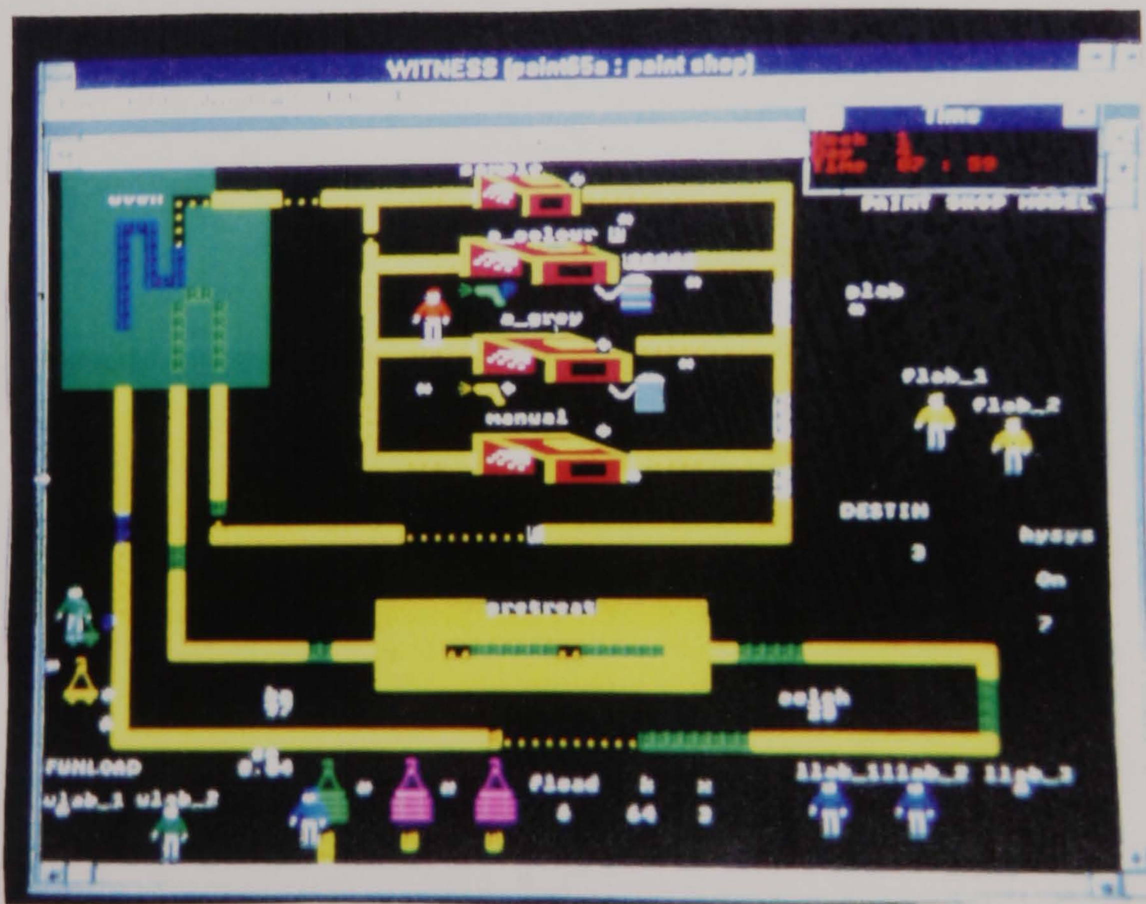
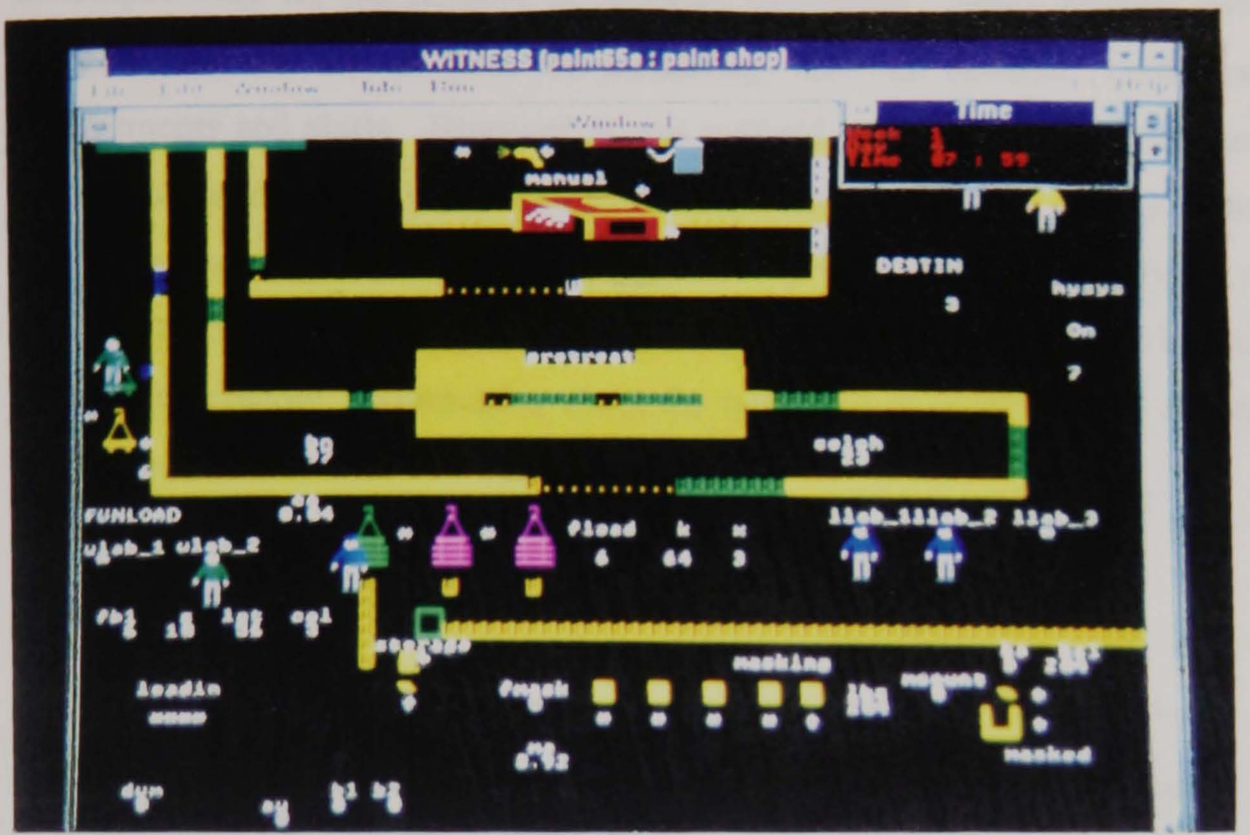


Figure B.4 The second part of the WITNESS model during experimentation



Appendix C. Technical Description of SIMFACTORY II.5 Model

The SIMFACTORY II.5 model does not contain too many details, because it was developed only for the purpose of software evaluation. Many features have not been included due to the purpose of modelling such as machine breakdowns, labour requirements and shifts. However, it was realized that some specific features could not have been modelled properly due to the software limitations. For example, setup on machines (painting booths) can be triggered only when input parts arrive at machine.

The SIMFACTORY II.5 model comprises several physical and logical elements, as described below. Information about the types and quantities of the elements used in the model is summarized in Table C.1.

Table C.1 A summary of the SIMFACTORY II.5 model elements

ELEMENT	QUANTITY
PHYSICAL ELEMENTS:	
Stations	9
Queues	6
Conveyors	14
Receiving Area	1
LOGICAL ELEMENTS:	
Attributes	2
Expressions	5

C.1 Physical Elements

(i) Stations

Several stations have been included in the SIMFACTORY II.5 model. In addition to loading and unloading stations, four painting booths were modelled as stations. Pretreatment, drying and baking stations were also defined, without being used in the model's logic. These processing areas were modelled as conveyors, but they were defined only for graphical purposes. This is because there is no screen editor provided by SIMFACTORY II.5, so only defined elements can be graphically displayed by icons.

(ii) Queues

Queues were used as dummy elements at particular sections of the conveyor chain. As SIMFACTORY II.5 does not permit the transferring of parts from one conveyor to another directly, nor is it possible to incorporate any decision logic into conveyors, queues were positioned at locations on conveyor system where conveyors are directed towards different booths, or they merge after painting.

(iii) Conveyors

A number of conveyors were defined in order to model various sections of the conveyor chain. Whilst different conveyors have different lengths, they all have the same speed. The number of separate conveyors defined for this model is smaller than the number of conveyors defined for the WITNESS model. The reason for this is a possibility to bend sections of one conveyor, so the entire part of a conveyor chain that precedes the first branching point was modelled by one conveyor.

(iv) Receiving Area

One Receiving Area was specified, where parts entered into the system.

C.2 Logical Elements

(i) Attributes

Two attributes were defined for the purpose of conditional routing: an attribute for the batch size and an attribute for the colour in which the part is to be painted.

(ii) Expressions

Several expressions were defined for the purpose of attributes assignment and for conditional routing.

Figure C.1 shows the display of the model before experimentation, whilst a graphical display of the model during experimentation is shown on Figure C.2.

Figure C.1 SIMFACTORY II.5 model before experimentation

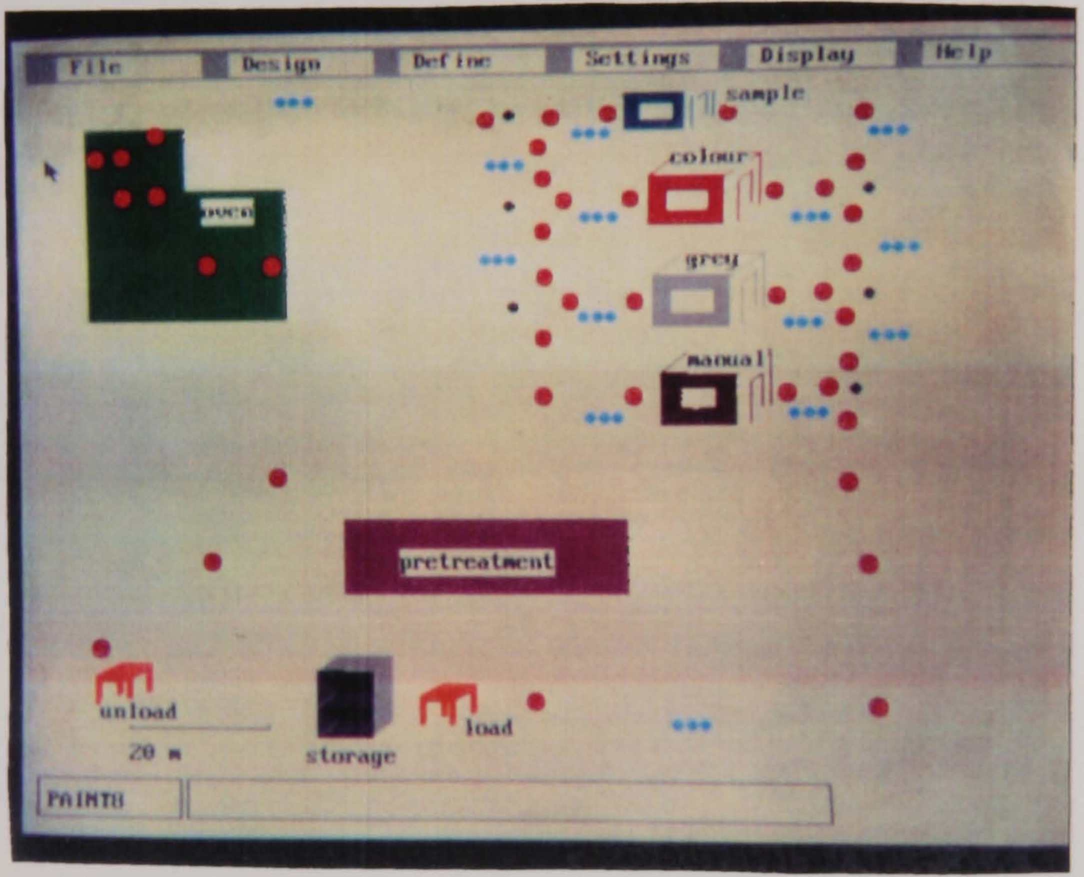
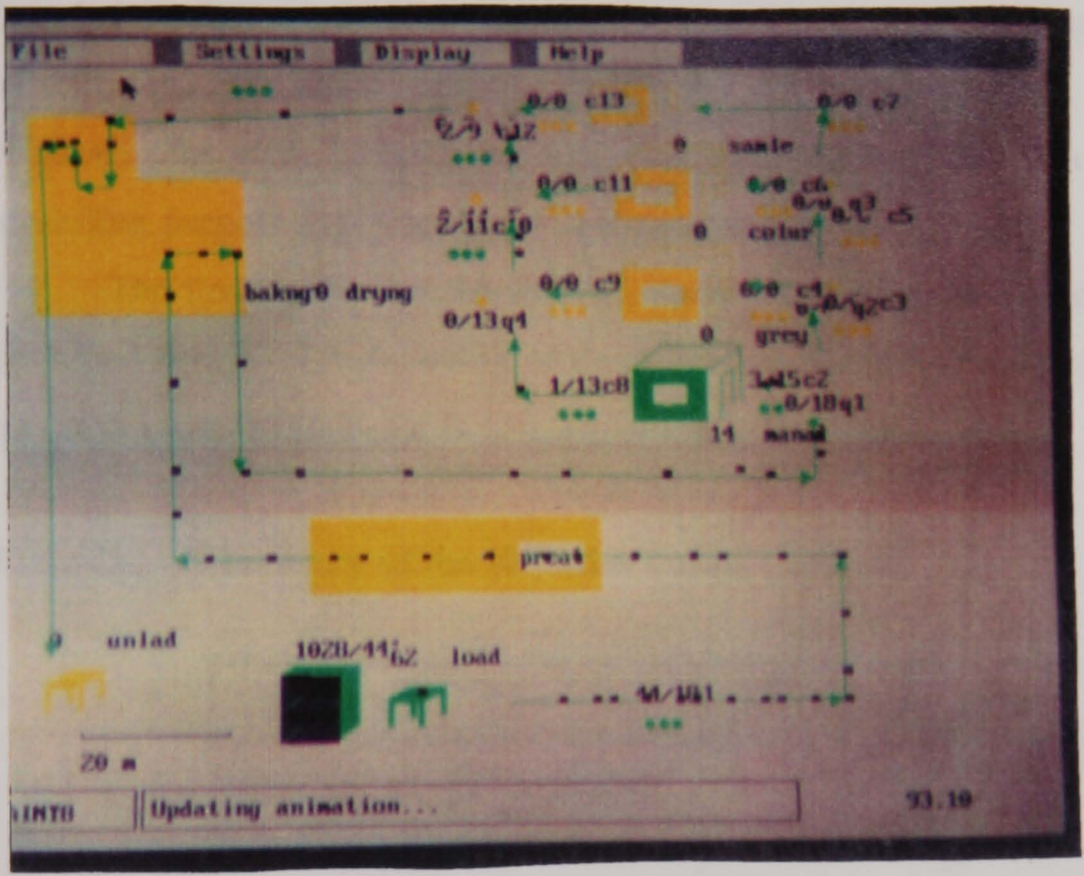


Figure C.2 SIMFACTORY II.5 model during experimentation



Appendix D. Technical Description of XCELL+ Model

The model developed using XCELL+ is 'quick and dirty'. It therefore does not contain many details, and it was developed rapidly mainly for the purpose of software evaluation. Some of the features were omitted due to the software characteristics (eg. part attributes, masking, machine setup, jigging of parts and conditional routing), whilst some other characteristics were excluded because they were not considered to be relevant for the modelling purpose (eg. machine breakdowns and shifts).

The final version of the XCELL+ model comprises several physical elements as described below. The summary of the types and quantities of the elements used in the model is provided in Table D.1.

Table D.1 A summary of the XCELL+ model elements

ELEMENT	QUANTITY
PHYSICAL ELEMENTS:	
WorkCentres	10
Buffers (Conveyor type)	8
Receiving Area	1
Shipping Area	1
LOGICAL ELEMENT:	
Process	1

D.1 Physical Elements

(i) WorkCentres

In addition to the WorkCentre used for painting of parts, there are several WorkCentres defined for logical purposes. Namely, between separate sections of conveyor which are modelled as buffers of conveyor type, there is one WorkCentre. The first reason for this is a routing restriction, which forbids parts to be moved from buffer to buffer (or from Receiving Area to buffer and vice versa). The second reason is a requirement that parts (parts actually represent loaded flight bars with seven parts, which is an average number obtained from the data collected in the real system) enter buffers one by one, not several at the same time. Therefore, the cycle time for every WorkCenter between sections of conveyor is 0.81 minutes which is a cycle of one flight bar. This ensures that the time between the arrival of every part (flight bar) to buffers (conveyors) is fixed.

The first WorkCentre that follows a Receiving Area, has increased cycle time for the average time that it takes to load one flight bar and for the average time that is lost because of gaps between batches. Similarly, the last WorkCentre that precedes the Shipping Area has a cycle time that equals the average time needed for unloading the parts from the flight bars.

(ii) Buffers (Conveyor type)

Sections of conveyor chain were modelled as buffers of conveyor type. This enabled the capacity of each buffer to be specified, which represents the number of flight bars comprised in a particular conveyor (section of conveyor chain). In addition to capacity, a Minimum Holding Time is specified, which represents the time each part has to spend in the buffer before it can leave it. In the model of the powder coating system, this time actually represented the time it takes each flight bar to travel from the beginning to the end of a particular section of the conveyor chain. This time was calculated by multiplying the capacity of the buffer by the cycle time of one flight bar (0.81 minutes).

(iii) Receiving Area

One Receiving Area was specified, where parts were generated and entered into the system.

(iv) Shipping Area

One shipping area was used, where parts were shipped from the system to the outside world after being unloaded from the flight bars at the last WorkCentre.

D.2 Logical Element

(i) Processes

According to a convention adopted by XCELL+, processes have the same name as parts that are processed in these processes. In this model, only one part type was defined, and therefore there was only one process specified for all WorkCentres.

Figure D.1 shows the display of the model before experimentation, whilst figure D.2 shows the display of the model during experimentation.

Figure D.1 XCELL+ model before experimentation

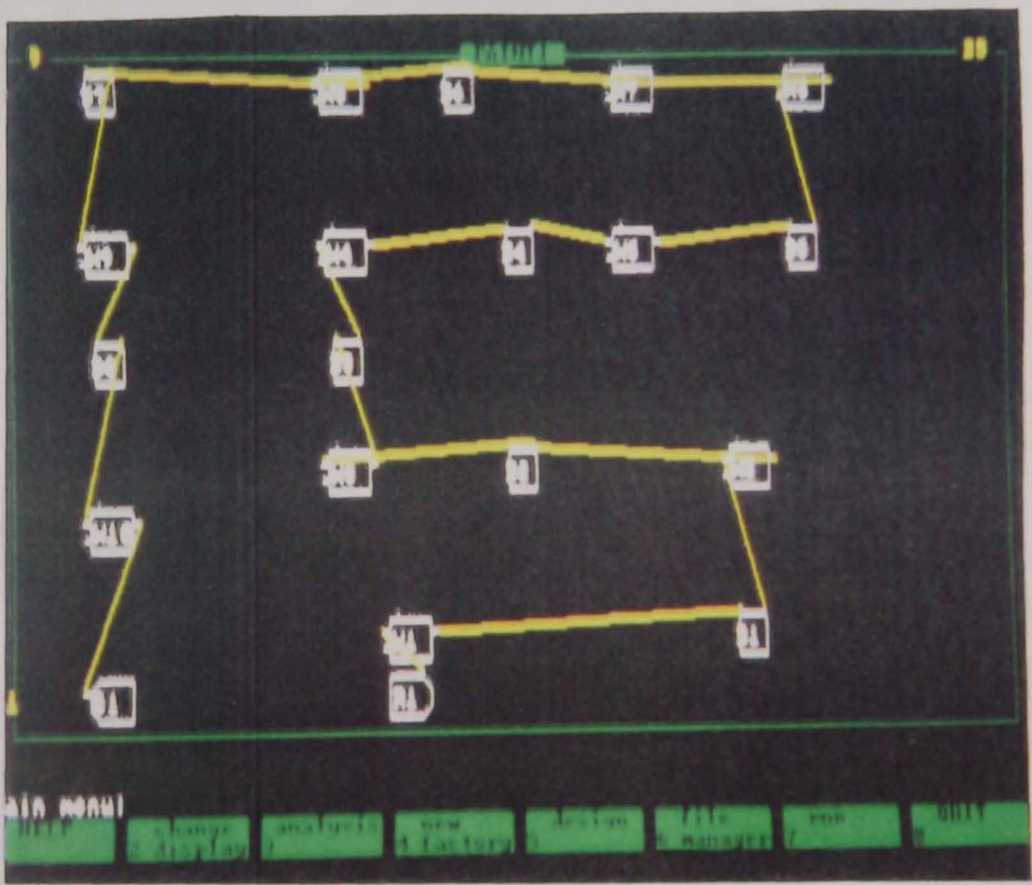
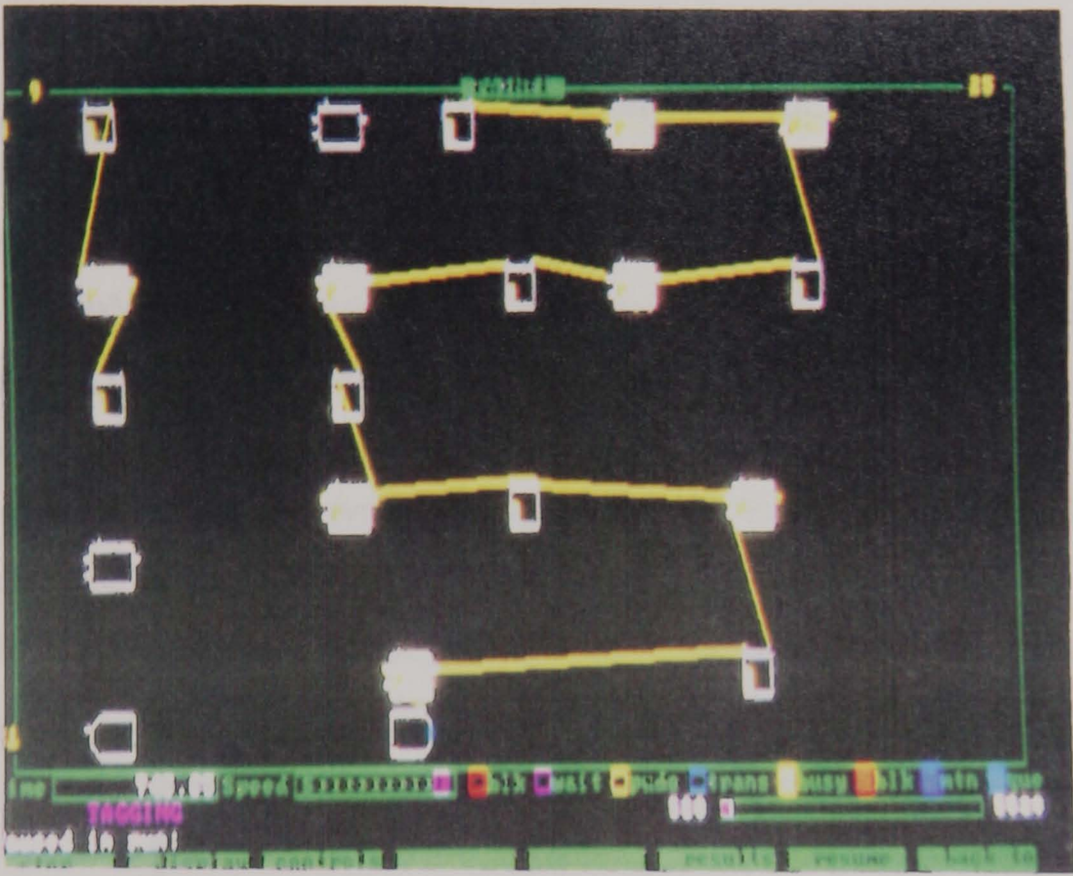


Figure D.2 XCELL+ model during experimentation



Appendix E: Technical Description of ProModelPC Model

A ProModelPC model was also developed only for the purpose of software evaluation. It contains a similar level of detail as the SIMFACTORY II.5 model, because of the features of these two simulators. For example, as both simulators enable modelling of part attributes, the main attributes such as the batch size or colour were defined and used for conditional routing. None of the models include details such as machine breakdowns, labour requirements and shifts.

ProModelPC comprises several physical elements, although they have not been separately defined and modelled as was the case for the other models. In this model, physical elements were specified within the location column in the routing section. Information about the types and quantities of the elements used in this model is summarized in Table E.1.

Table E.1 A summary of the ProModelPC model elements

ELEMENT	QUANTITY
PHYSICAL ELEMENTS:	
Routing locations	7
Conveyors	5
LOGICAL ELEMENTS:	
Dummy routing locations	2
User-defined distribution	1
Attributes	3

E.1 Physical Elements

(i) Routing locations

Routing locations relate to physical locations used for processing such as loading and unloading stations or painting booths. Pretreatment, drying and baking stations were not defined, because these processing areas were modelled as conveyors. As ProModelPC enables adding the graphical elements on the screen, these areas were additionally indicated on the screen, after the model has been developed.

(ii) Conveyors

Since ProModelPC enables merging and branching of conveyors, this model contains a relatively small number of conveyors in comparison to other models. As the entire logic of the model is defined within the routing section, conveyors are used only as the means of transport, and as such do not include any logic. They are merely used to model an elapse of time whilst parts are being transported between different locations.

E.2 Logical Elements

(i) Dummy routing locations

Several dummy routing locations were defined for logical purposes. For example, a dummy location has been used for conditional routing on the basis of attribute values.

(ii) User-defined distribution

User defined distribution has been used to provide values for the attribute related to colour in which a part is to be painted.

(iii) Attributes

Few attributes were defined in this model: attribute for the batch size, attribute for the number of parts per flight bar, and attribute for colour.

Figure E.1 shows the display of the model before experimentation, whilst a graphical display of the model during experimentation is shown on Figure E.2.

Figure E.1 ProModelPC model before experimentation

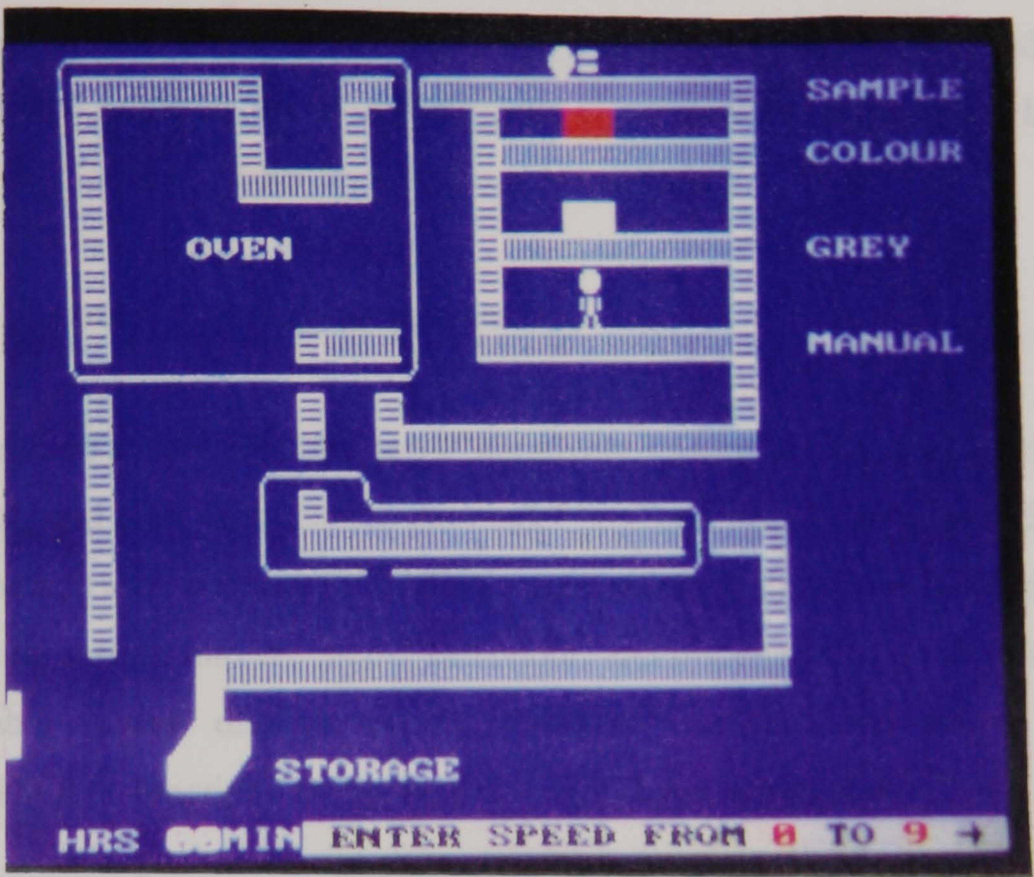
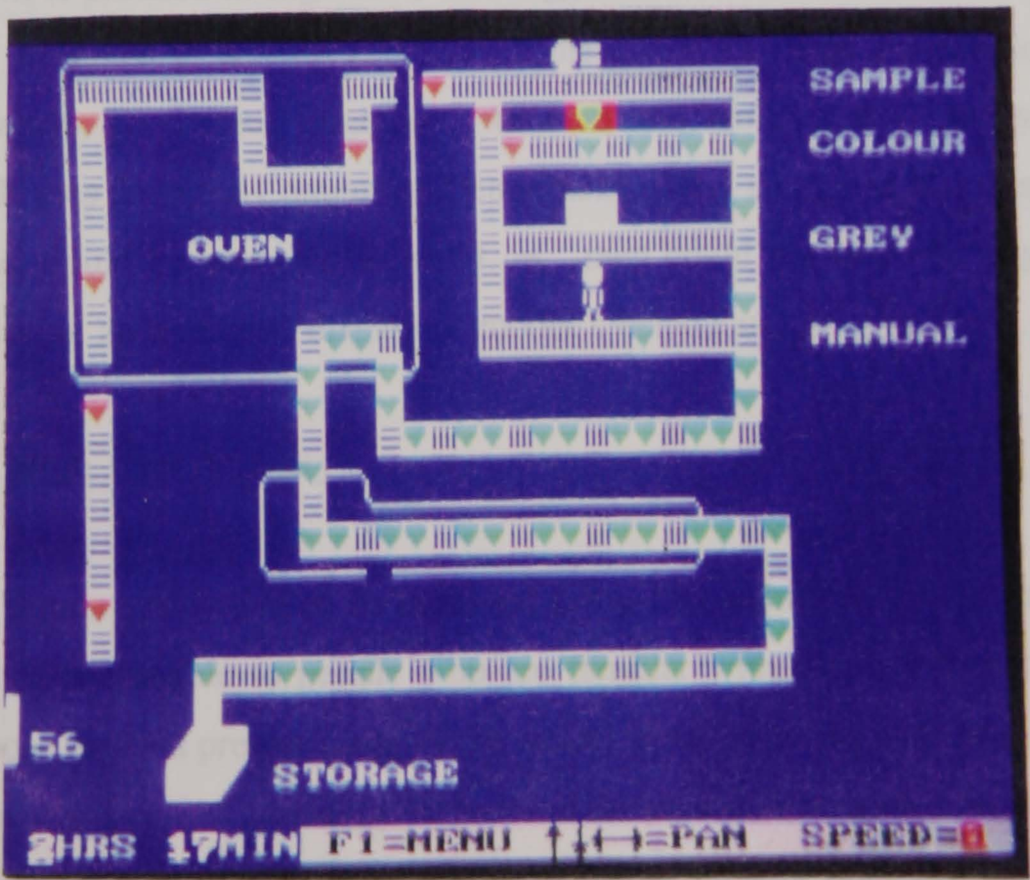


Figure E.2 ProModelPC model during experimentation



Appendix F. Verification Techniques Applied

Several different methods were used for computer model testing. These methods were applied to the WITNESS model, which was experimented with. Model verification was performed applying the formal methods of verification as described by Law and Kelton (1991). These methods are described in sections F.1 - F.8.

F.1 Gradual model development

Simulation model was developed iteratively. More details were gradually added to the model, and after each change the model has been tested using several facilities provided by the software package WITNESS. Animation of simulation was very useful, enabling a visual insight into the model behaviour, and discovering logical mistakes. Displays of variable values, displays of the status of elements, and access to part attributes (EXPLODE function) also provided a significant testing aid.

This gradual model development resulted in about 70 different versions of the model. Each version incorporated additional details, and was thoroughly tested until the final version of the model was obtained with the results corresponding to the real data.

Many of these models developed through the iterative process described were shown to production engineers, who suggested what had to be modified. The final version was presented to a wide audience in the company. The presentation was attended by managers, production engineers, foreman and workers from the shop floor, and the employees from the Computing Department. They all agreed that this final version of the model could be accepted as valid, ie. they could not discover any major mistake by observing an animated display of model's behaviour.

F.2 High Face Validity

The primary objective to be achieved using this technique was to develop a model that seemed reasonable to people who are familiar with the system under study. For that purpose, several presentations of the model were organized in the company, as mentioned in Section F.1. The participants of the presentation agreed that the model developed did not appear to be wrong, and accepted it as an adequate representation of the system being studied.

F.3 Running the model under simplifying assumptions

The model was run under simplifying assumptions, in order to test its behaviour. For example, only two part attributes were introduced, and then deterministic values assigned to these attributes. These attributes referred to the batch size and colour. Only two colours were used and two values for the batch size, in order to test conditions specified for routing of batches to different booths, and to model placing the gaps on the conveyor after each batch. It was assumed that only one part was hung on the flight bars. With such simple and deterministic values, it was possible to determine fairly quickly via the animated display of the model whether its specific logical parts were correct.

F.4 Animation of simulation

Although this method overlaps with the above mentioned verification methods, it is listed separately due to its significance. During the iterative process of model development, animation was constantly used to reveal the impact of any modification incorporated in the model.

A graphical display of the system layout was created using the Icon and Screen editors. An animated display of the model's behaviour, together with the display of variables representing part attributes and counters showed the dynamic changes which happened in the system during experimentation. Another useful feature of the software was a step function. This function enables running the model one step at the time, and writing a message on the screen which event has occurred.

In addition to the model animation, time series and histograms have also been designed. This provides a dynamic graphical display of model output during experimentation, such as information about the current throughput, labour utilization and utilization of painting booths. A dynamic display of simulation gave an additional proof that there were no significant errors in the model.

F.5 Use of a simulator

The use of a simulator significantly reduces the required number of lines of written code.

Verification techniques applied

Although it is usually not possible to avoid programming when a complex real manufacturing system is modelled, in most cases it is better to use a data driven simulator that enables additional code to be added for modelling specific features than to program everything from scratch.

Such an approach was adopted in this research. A data driven simulator was used for the specification of basic elements and characteristics of the model, whilst special logical features were programmed and added into the model. Although a significant programming effort had to be invested due to the model complexity, this approach resulted in a smaller amount of code to be debugged than would be the case if a simulation language was used for model development. The code has been developed mainly in the form of functions and routing conditions, which provided modularity that additionally eased debugging.

F.6 Distribution fitting

Data collected on the shop floor was statistically analyzed and used in the model. Values recorded for the batch size and the number of parts per flight bar, which have a significant influence on product mix, were fitted into theoretical statistical distributions as described in section 3.4.

F.7 Sensitivity analysis

Sensitivity analysis has been carried out in order to test to what extent the simulation output is sensitive to the changes of input. The main characteristic of the system under study is a large variety of product types. Input variables relating to the batch size and the number of parts per flight bar do represent a significant source of such variability. These variables were therefore chosen as two factors in the complete 2^2 experimental design, whilst the throughput was selected as a response variable because the efficiency of the system depends on the value of this variable.

Two values of each chosen factor have been used in the experimental design: one value relates to the fitted distribution, whilst another value relates to the sample mean (mean values obtained for the collected data). The levels of factors are presented in Table

F.1.

Table F.1 Selected levels of factors in 2^2 factorial design

FACTOR	LEVELS	
	+ (FITTED DISTRIBUTION)	- (SAMPLE MEAN)
BATCH SIZE	lognormal (80.8575,118.38)	80
NUMBER OF PARTS PER FLIGHT BAR)	lognormal (8.5,11)	7

Such a design of experiments determined four combinations of factors, presented in Table F.2:

Table F.2 Combinations of factors in experimental design

COMBINATION	FACTOR 1 BATCH SIZE	FACTOR 2 NUMBER OF PARTS PER FLIGHT BAR	RESPONSE AVERAGE DAILY THROUGHPUT
1.	+	+	R_1
2.	+	-	R_2
3.	-	+	R_3
4.	-	-	R_4

On the basis of the values of responses, the main and interaction effects have been calculated. The main effect of factor 1 for the first response was obtained as:

$$e_1 = ((R_1-R_3)+(R_2-R_4))/2 \qquad (1)$$

The main effect of factor 2 for the first response was obtained as:

$$e_2 = ((R_1-R_2)+(R_3-R_4))/2 \qquad (2)$$

The interaction effect, which measures the dependency of the effect of the first factor on the level of the second factor was obtained as:

$$e_{12} = ((R_1-R_3)-(R_2-R_4))/2 \quad (3)$$

The simulation conditions have been the same for both models. A sample size of three independent simulation experiments was chosen for each combination of factors. The measures of system performance considered were the mean values of the results obtained by each replication. Each replication simulated the performance of the system in a period of 4 weeks with a warm-up period of 4 days. Running three replications of experiments for every combination of two factors resulted in 12 experiments.

The values of an average daily throughput have been used to calculate the effects of changing the levels of the factors on system response. Table F.3 shows the average values (obtained from three independent runs) of daily throughput obtained for each combination, whilst the main and interaction effects obtained are presented in Table F.4.

Table F.3 Average daily throughput obtained for 4 combinations of factors

COMBINATION	RESPONSE AVERAGE DAILY THROUGHPU T	DIFFERENCE FROM REAL DATA
1.	1429	-1.06%
2.	2193	55%
3.	1420	0.42%
4.	2009	42%

Table F.4 Main and interaction effects for selected combinations of factor's levels

	R_1
e_1	96.5

e_2	-676.5
e_{12}	-87.5

Results obtained for the main and interaction effects show that the average daily throughput has been increased by 96.5 parts because a mean sample value (obtained from the data collected on the shop floor) for the batch size has been used instead of fitted theoretical distribution. The effect obtained for the second factor shows that the model is much more sensitive to the use of mean sample value for the number of parts per flight bar instead of fitted distribution. This resulted in a decrease of an average daily throughput by 676.5 parts. The interaction effect shows that using a mean sample value for the batch size depending on the values used for the number of parts per flight bar, resulted in a decrease of average daily throughput by 87.5 parts.

The results presented above reveal that product mix has a significant impact on throughput. The main factor that determines product mixture is the number of parts per flight bar. This could be expected due to a large variety of part types, in a range from one part per two flight bars to five hundred parts per one flight bar.

Another interesting point is comparison of the average values of the performance measures obtained for four combinations of factors with real data. This comparison shows that when deterministic mean value for the number of parts per flight bar was used instead of stochastic fitted distribution, the average daily throughput obtained from the model differs from the real data by 55%. This further supported decision for using theoretical fitted distribution for final experimentation instead of deterministic value. Difference obtained for the batch size did not provide evidence for the benefits of using stochastic distribution instead of deterministic value. However, the table presented in the next section shows that when the final version of the model which used stochastic distribution for both factors was run for a longer period, the difference between model output and real data was further decreased to 0.3%.

F.8 Comparison of simulation output with real data

This was the most important and definite test of model validity. Using this method, results obtained from the model were compared to the real data collected on the shop floor. The values compared relate to the average daily throughput ie. the number of components coated per day.

Table F.5 shows a comparison of the results obtained for the model from three independent runs with different random number seeds in the period of 40 days with a warm-up period of 7 days with the average data collected in the factory. The average value of real data has been derived as a mean value of the daily throughput in March 1992 when data was collected for the modelling purpose, and the daily throughput in July 1991 when a group of production engineers carried out a research on the paint shop performance.

Table F.5 Comparison of model results and real data

	MODEL RESULTS	REAL DATA (March 1992)	REAL DATA (July 1991)	AVERAGE REAL DATA	DIFFER. (model-average real data)
AVERAGE DAILY THROUGHPUT	**1419**	1447	1381	**1414**	0.3%

The above results show that the difference between an average model output and average real data is only 0.3%, which was a final proof that simulation model could be accepted as an appropriate approximation of real system and used for further experimentation in order to test various production alternatives.

Appendix G. Examples of Simulation Results

In addition to the results presented in Chapter 3, some further results are presented in this appendix, which were obtained during the experimentation with WITNESS models. Figure G.1 presents the average daily throughput obtained for the first model in three independent runs, with a run length of 70 days, and without warm-up period.

The results shown on other three figures were obtained in three independent runs, with a run length of 40 days, with a warm-up period of 7 days. Figure G.2 presents the average daily throughput obtained for six different models. The machine utilization obtained for the first model which simulated present situation is presented in Figure G.3, whilst the labour utilization obtained for the same model is presented in Figure G.4.

Figure G.1 shows that the average daily throughput obtained for the first model significantly varied during the simulated time. The reason for this is a large variety of different part types. A certain product mix determines the size of the parts, the number of parts that can be hung on the flight bar, which subsequently determine the throughput in certain period.

When such a type of system is simulated, it is not possible to explicitly determine the steady-state of the system, because it will never be reached due to the product mixture, as far as the throughput is concerned. Therefore, a warm-up period for the subsequent experiments was determined approximately in order to ensure that there is no shortage of the parts in the raw material storage area.

The results regarding the daily throughput obtained for six models demonstrate trend similar to the results relating to total throughput, which could be expected. It is apparent that the highest increase in daily throughput was obtained for the second model (where the number of parts per flight bar was doubled), whilst for other four models an increase was slight but not insignificant.

With regard to machine utilization in the first model, it is evident that one booth is much more utilized that other. This problem was foreseen during the data analysis, because it was realised that the light grey colour to which one automatic booth was dedicated has been used for only 14.3% of the parts. On the other hand, mid grey colour has been used for 48.8% of the parts and this colour was painted together with all other colours on another automatic booth. Therefore, that booth used for painting all other colours except the light grey, was much more utilized than all other booths. The managers and production engineers were not aware of this fact before

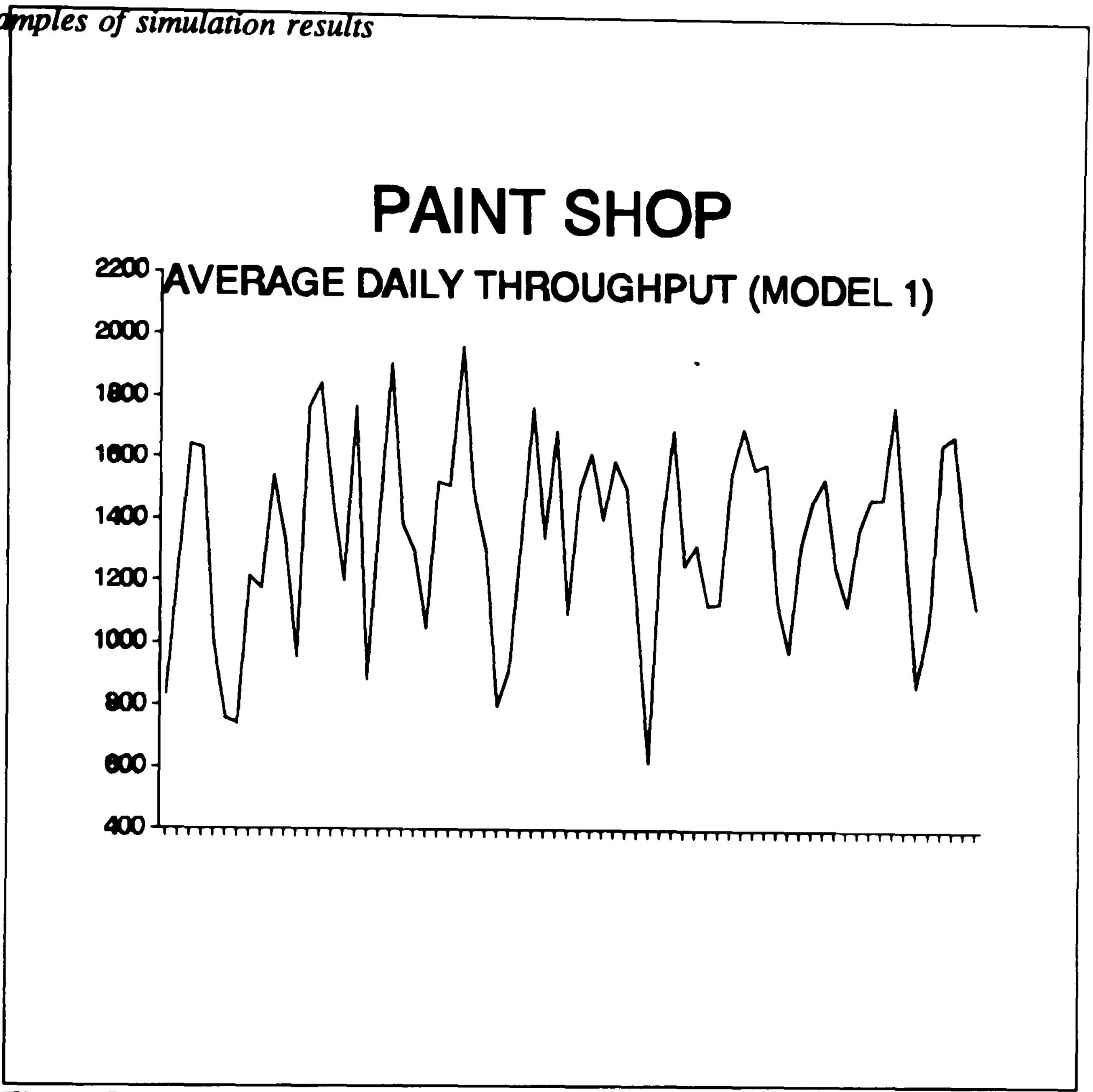


Figure G.1 The average daily throughput obtained for the first model

this study, and afterwards they have taken an action to change this mistake.

The least utilized booth is the booth used for painting of sample batches (up to two parts in batch), which was also expected, because a sample paint is done very rarely. The total utilization of all booths is relatively low. The reason for this is the fact that only one booth can be used at a time, due to the system configuration.

The results obtained for labour utilization in the first model show that labour is not a critical resource in the system, although the results are slightly underestimated. Namely, the labour activities relating to setup of the booths and system maintenance are not included in the model due to the software characteristics (during the maintenance system does not operate, and this was included in the shift pattern). Since the results obtained for labour utilization relate to the period when the system was operating, final results were not significantly influenced by omission of maintenance activities.

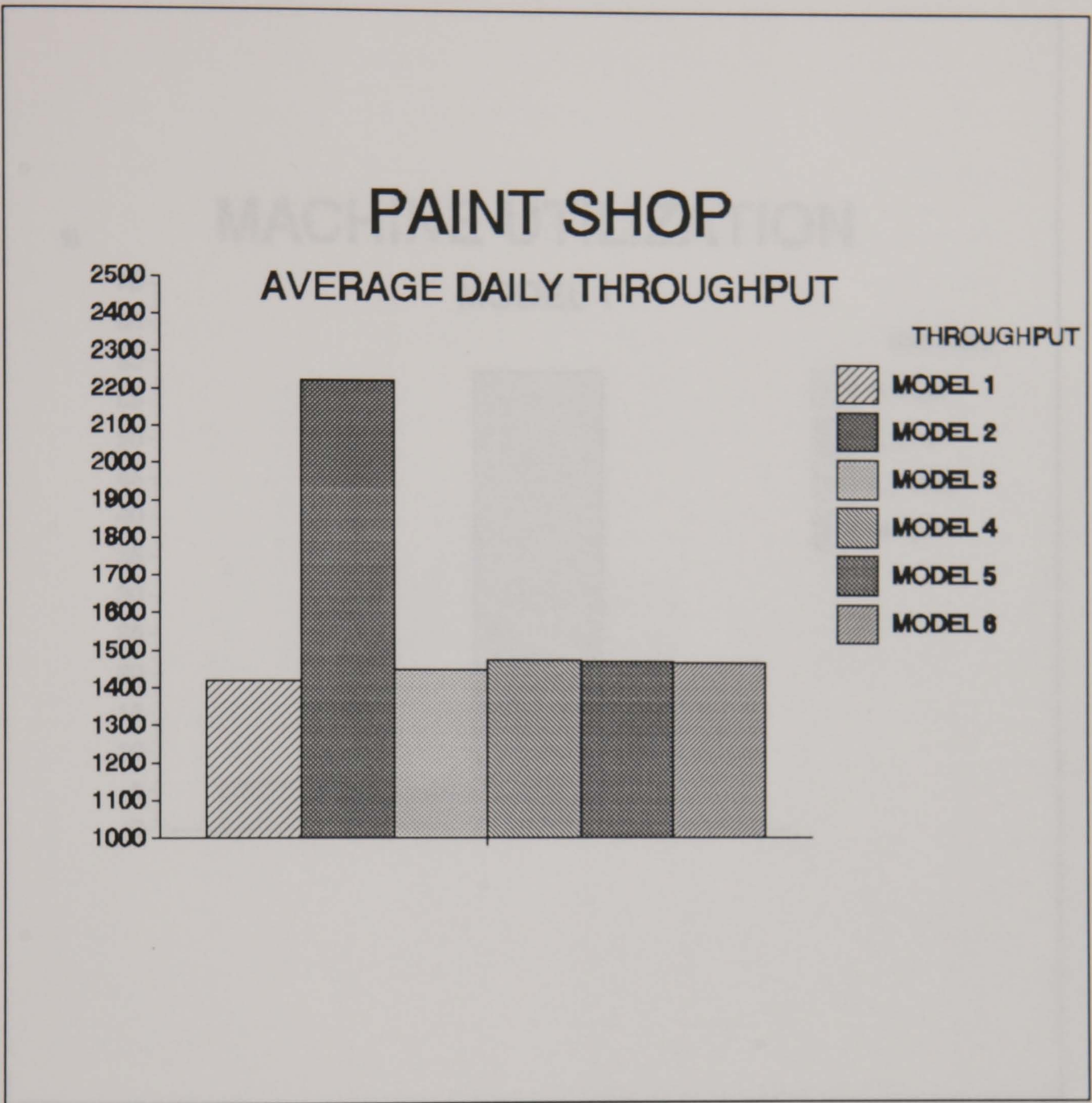


Figure G.2 Average daily throughput obtained for six models

On the other hand, an exclusion of the setup activity can have some impact on final results. An approximate calculation of the average time spent on setup (according to the data related to the average colour change) revealed that 10% of utilization can be added to each type of labour assuming that all of them have equally participated in this activity.

Even by an increase of the results by 10%, none of the workers would have been utilized more than 65%, which supports the claim that labour is not a resource in shortage. It is rather advisable to find a way of better labour utilization, perhaps by its rearrangement and modification of the shift pattern to extend the operating hours of the system.

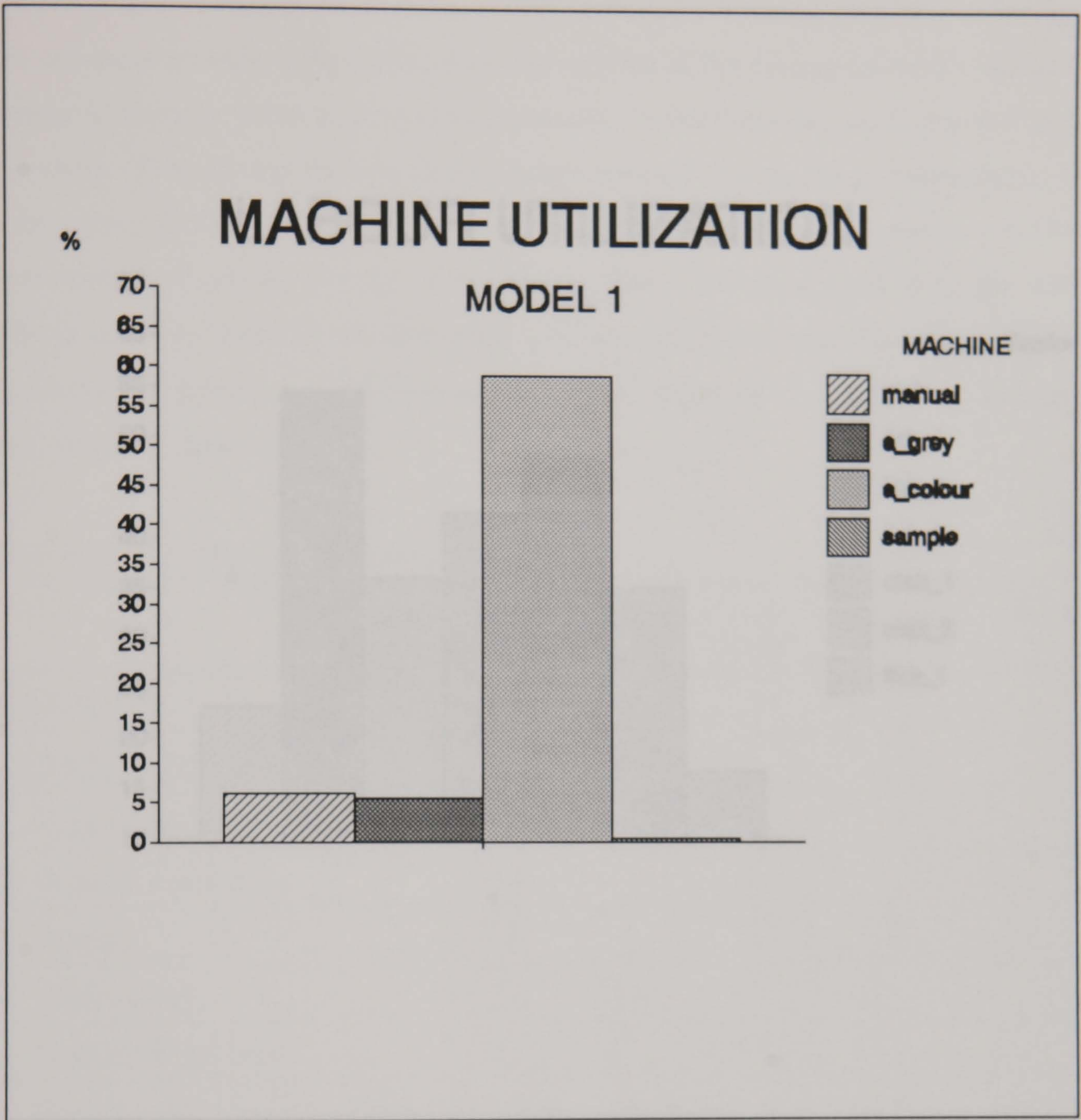


Figure G.3 Machine utilization obtained for the first model

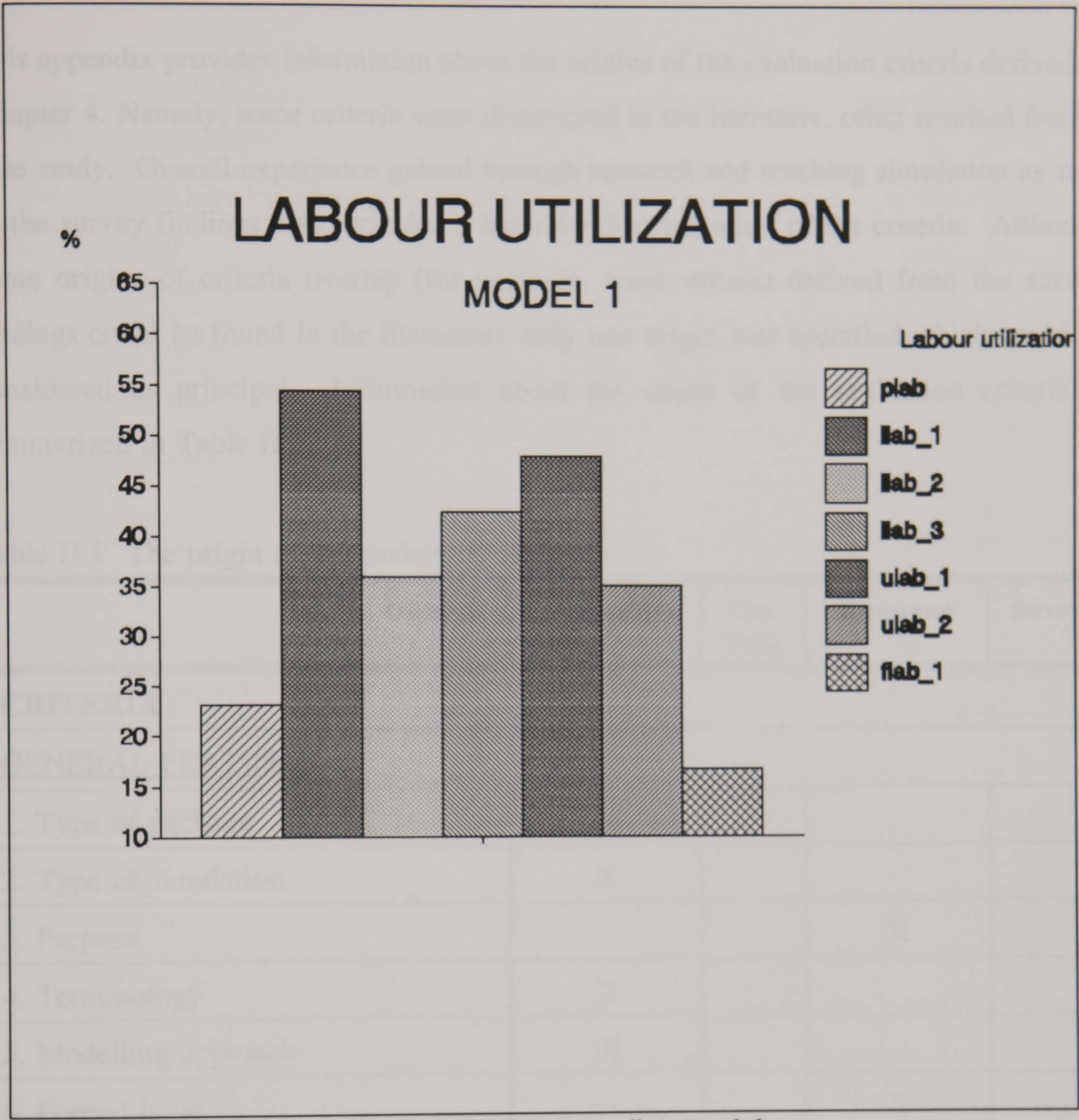


Figure G.4 Labour utilization obtained for the first model

Appendix H. The Origins of the Evaluation Criteria

This appendix provides information about the origins of the evaluation criteria derived in Chapter 4. Namely, some criteria were discovered in the literature, other resulted from a case study. Overall experience gained through research and teaching simulation as well as the survey findings also provided a basis for determination of the criteria. Although some origins of criteria overlap (for example, some criteria derived from the survey findings could be found in the literature), only one origin was specified which could be considered as principal. Information about the origin of the evaluation criteria is summarized in Table H.1.

Table H.1 The origin of the evaluation criteria

ORIGINS	Literature	Case Study	Experience	Survey
CRITERIA				
GENERAL FEATURES				
1. Type of package	X			
2. Type of simulation	X			
3. Purpose			X	
4. Terminology	X			
5. Modelling approach	X			
6. Formal logic			X	
7. Representativeness of models	X			
8. Ease of conceptualization of simulation logic	X			
9. Modelling transparency	X			
10. Hierarchical model building				X
11. Run-time applications				X
12. Conceptual model generator				X
13. The length of entity name				X
14. Entity name		X		
15. Experience required for software use			X	

ORIGINS	Literature	Case Study	Experience	Survey
16. Formal education in simulation required for software use			X	
17. User friendliness			X	
18. Ease of learning			X	
19. Ease of using			X	
20. Initialization		X		
21. Specification of time units	X			
22. Specification of length measures		X		
VISUAL ASPECTS				
1. Animation			X	
2. Type of animation			X	
3. Timing of animation	X			
4. Type of graphical display	X			
5. 3-D graphic			X	
6. Integrity of graphics			X	
7. Animation layout development		X		
8. Multiple screen layout		X		
9. Animation with visual clock			X	
10. Icon editor			X	
11. Screen editor		X		
12. Ease of icon development			X	
13. Ease of using screen editor		X		
14. Type of icons	X			
15. Icon library		X		
16. Merging icon files			X	
17. Resizing of icons		X		
18. Rotating of icons		X		
19. Changing the colour of the icons		X		
20. Zoom function			X	

ORIGINS	Literature	Case Study	Experience	Survey
21. Panning			X	
22. Switching on/off the graphics			X	
23. Switching between screens		X		
24. Switching between character and icon graphic	X			
25. Print screen facility			X	
26. Virtual screen			X	
27. Indication of elements' status			X	
28. Changing the colour of the elements' status display		X		
29. Limitation on number of displayed icons		X		
30. Number of icons stored in icon library		X		
31. Change of icons during simulation		X		
32. Icons with multiple colours		X		
33. Easy copying of icons			X	
<u>CODING ASPECTS</u>				
1. Programming flexibility			X	
2. Program generator			X	
3. Access to source code			X	
4. Readability of source code			X	
5. Readability of added code			X	
6. Self-documentation of added code	X			
7. Precision of added code	X			
8. Comprehensiveness of added code			X	
9. Link to a lower language		X		
10. Data storage, retrieval and manipulation facilities	X			

ORIGINS	Literature	Case Study	Experience	Survey
11. Quality of data storage, retrieval and manipulation facilities	X			
12. Built-in function		X		
13. User functions		X		
14. Global variables			X	
15. Names of functions, variables and attributes		X		
16. Writing comments for logical elements		X		
17. Type of time variable	X			
18. Type of translation			X	
19. Text/code manipulation		X		
20. Length of lines in coding editor		X		
21. Support of programming concepts				X
22. Quality of programming concepts support				X
23. Object oriented programming concepts				X
EFFICIENCY				
1. Robustness	X			
2. Level of detail			X	
3. Number of elements in the model		X		
4. Model reusability			X	
5. Model status saving			X	
6. Automatic saving				X
7. Interaction			X	
8. Adaptability	X			
9. Multitasking			X	
10. Model chaining ie. linking output from different models		X		

ORIGINS	Literature	Case Study	Experience	Survey
11. Exit to the operating system within the package		X		
12. Compilation time			X	
13. Model execution time			X	
14. Case sensitivity			X	
15. Conversion of numbers (real v integer)		X		
16. Queuing policies		X		
17. Number of queuing policies		X		
18. Time scale for model building			X	
19. Reliability		X		
20. Pre-existing generic models				X
21. Merging of models			X	
22. Editing partially developed models		X		
23. Automatic model building		X		
24. Ease of model editing		X		
25. Specification of part flow by a mouse			X	
<u>MODELLING ASSISTANCE</u>				
1. Prompting		X		
2. Quality of prompting		X		
3. Modularity	X			
4. Model and data separation			X	
5. Use of mouse			X	
6. On-line help			X	
7. Quality of on-line help			X	
8. Documentation notes		X		
9. Quality of facility for documentation notes		X		

ORIGINS	Literature	Case Study	Experience	Survey
10. Text editor as integral part of the package		X		
11. Automatic editing of data		X		
TESTABILITY				
1. Logic checks		X		
2. Error messages			X	
3. Quality of error messages			X	
4. Moment of error diagnosis		X		
5. Ease of debugging	X			
6. Display of function values		X		
7. Display of attributes		X		
8. Access to attributes		X		
9. Display of variables		X		
10. Display of element's state		X		
11. Dynamic display of capacity			X	
12. Display of the workflow path		X		
13. Display of events of the screen		X		
14. Display of part position within element		X		
15. Facility for immediate user actions		X		
16. List files		X		
17. Echo		X		
18. Trace files			X	
19. Explode function		X		
20. List of used elements		X		
21. Backward clock		X		
22. Step function (event to event jumping)			X	
23. Flow analysis		X		

ORIGINS	Literature	Case Study	Experience	Survey
24. Audible alarms		X		
25. Rejection of illegal inputs		X		
SOFTWARE COMPATIBILITY				
1. Integration with spreadsheet packages				X
2. Integration with statistical packages				X
3. Integration with word processors			X	
4. Integration with CAD software				X
5. Integration with DBMS				X
6. Integration with expert systems	X			
7. Integration with MRP II software				X
8. Integration with scheduling software	X			
INPUT/OUTPUT				
1. Menu driven interface			X	
2. Pull down menus			X	
3. Type of menu selection			X	
4. Selection buttons		X		
5. Dialogue boxes			X	
6. Multiple inputs		X		
7. Multiple outputs		X		
8. General output reports			X	
9. Static graphical output			X	
10. Dynamic graphical output			X	
11. User defined output		X		
12. Automatic rescaling of histograms and time series		X		
13. Quality of output reports			X	
14. Understandability of output reports			X	

ORIGINS	Literature	Case Study	Experience	Survey
15. Periodic output of simulation results			X	
16. Availability of results before end of simulation		X		
17. Input data reading from files		X		
18. Writing reports to files		X		
19. Writing reports to printer		X		
20. Writing reports to plotter	X			
21. Snapshot reports		X		
22. Summary reports for multiple runs		X		
<u>EXPERIMENTATION FACILITIES</u>				
1. Automatic batch run			X	
2. Warm-up period			X	
3. Re-initialization			X	
4. Re-start from non empty state			X	
5. Breakpoints		X		
6. Speed adjustment			X	
7. Experimental design capability			X	
8. Quality of experimental design capability			X	
9. Accuracy check		X		
10. Automatic determination of run length		X		
<u>STATISTICAL FACILITIES</u>				
1. Theoretical statistical distributions			X	
2. Number of theoretical statistical distributions			X	
3. User-defined distributions			X	
4. Random number streams			X	

ORIGINS	Literature	Case Study	Experience	Survey
5. Number of random number streams		X		
6. User specified random number streams		X		
7. Antithetic sampling			X	
8. Distribution fitting		X		
9. Goodness-of fit tests		X		
10. Output data analysis				X
11. Quality of data analysis facility			X	
12. Confidence intervals		X		
USER SUPPORT				
1. Documentation			X	
2. Quality of documentation	X			
3. Reference card		X		
4. Demo disk			X	
5. Tutorial			X	
6. Training course	X			
7. Duration of training courses			X	
8. Frequency of training courses			X	
9. Demo models			X	
10. Help-line		X		
11. User group meetings			X	
12. Frequency of user group meetings			X	
13. Newsletter	X			
14. Package maintenance			X	
15. Consultancy	X			
FINANCIAL AND TECHNICAL FEATURES				
1. Portability			X	
2. File conversion			X	

ORIGINS	Literature	Case Study	Experience	Survey
3. Price			X	
4. Installation costs	X			
5. Ease of installation			X	
6. Hardware requirements			X	
7. Availability of package on standard hardware	X			
8. Availability of package on standard operating systems	X			
9. Version of software for network				X
10. Virtual memory facility	X			
11. Security device		X		
12. Free software trials			X	
13. Free technical support	X			
14. Types of contract available	X			
15. Educational discount			X	
16. Quantity discount	X			
17. Life cycle maintenance costs	X			
18. Price of training course			X	
19. Consultancy fees	X			
20. Frequency of update	X			
21. Comprehensiveness of update			X	
PEDIGREE				
1. Age	X			
2. Genealogy	X			
3. Spread			X	
4. Success				
5. Availability of references			X	
6. Reputation of supplier			X	
7. Sources of information about the package	X			

ORIGINS	Literature	Case Study	Experience	Survey
GENERAL MANUFACTURING MODELLING FEATURES				
1. Problem areas tackled	X			
2. Applicability for manufacturing systems	X			
3. Equipment breakdown modelling			X	
4. Type of breakdowns		X		
5. Machine setup modelling			X	
6. Machine teardown modelling			X	
7. Rejects modelling			X	
8. Capacity of manufacturing equipment			X	
9. Shifts modelling			X	
10. Maintenance modelling		X		
11. Automatic increasing of buffer capacity		X		
12. Buffer delays		X		
13. Job lists		X		
14. Part attributes modelling			X	
15. Frequency of part arrival modelling			X	
16. Arrival of parts in batches		X		
17. Type of part arrival			X	
18. Variable conveyor speed		X		
19. Assembly operation modelling		X		
20. Disassembly operation modelling		X		
21. Containerization modelling	X			
22. Fixturing modelling	X			
23. Palletization modelling	X			
24. Evaporation of fluids modelling				X
25. Precipitation of fluids modelling				X

ORIGINS	Literature	Case Study	Experience	Survey
26. Fluid composition modelling				X
27. Inspection operation modelling			X	
PHYSICAL ELEMENTS				
1. Single machines			X	
2. Batch machines			X	
3. Production machines		X		
4. Assembly machines			X	
5. Multi-cycle machines		X		
6. Multi-station machines		X		
7. Buffers			X	
8. Workstation buffers	X			
9. Labour			X	
10. Automated guided vehicles (AGVs) and trucks			X	
11. Conveyors			X	
12. Types of conveyors		X		
13. Branching and looping of conveyors		X		
14. Conveyor buffers	X			
15. Fork-lifts	X			
16. Robots	X			
17. Automated storage retrieval system	X			
18. Tools			X	
19. Automated tool storage			X	
20. Pallets	X			
21. Fixtures	X			
22. Fixture stores	X			
23. Pallet shuttles	X			
24. Carousel-type magazines	X			

ORIGINS	Literature	Case Study	Experience	Survey
25. Cranes			X	
26. Tanks and fluids				X
SCHEDULING FEATURES				
1. Scheduling rules			X	
2. Number of scheduling rules provided			X	
3. Remaining processing time calculation	X			
4. Conditional routing		X		
5. Priority			X	
6. Preemption		X		
7. Push/pull from specific positions within the element		X		
8. Specification of quantity of parts to be moved between elements		X		
9. Batch index		X		
10. Predefined part routing		X		
11. Routing restrictions		X		
12. Type of part sequencing		X		
13. Departure scheduling for shipping area		X		
14. Vehicle scheduling	X			
15. Vehicle acceleration and deceleration	X			
16. Scheduling optimization		X		
MANUFACTURING PERFORMANCE				
1.Throughput			X	
2. Work in progress			X	
3. Utilization of production equipment			X	
4. Makespan			X	

ORIGINS	Literature	Case Study	Experience	Survey
5. Special user-defined reports		X		
6. Due dates monitoring	X			
7. Manufacturing costs analysis			X	
8. Schedule related output	X			
9. Transportation time of parts	X			
10. Rework and scrap level		X		
11. Interruption reports			X	
12. Production sequence summary		X		

Appendix I. Description of Evaluation Criteria

This appendix provides a description of criteria derived in Chapter 4. Criteria described are grouped in the groups of criteria and listed in the same order as they are specified in Chapter 4.

Section I.1 provides a description of general criteria that can be applied for the evaluation of both general and special purpose simulation packages. Criteria specific for evaluation of packages for manufacturing simulation are described in section I.2.

I.1. GENERAL CRITERIA

Criteria within this group are grouped in thirteen groups ((i)-(xiii)) according to their character. Chapter 4 provides a general description of each group of general criteria.

(i) General Features

1. Type of package

The classification of simulation software according to the type adopted in this research is the one proposed by Law and Kelton (1991). They classify simulation packages as simulation languages and simulators (with or without programming). The subject of this research are manufacturing simulators, and all criteria are described from their perspective.

2. Type of simulation

This criterion examines in which way the variables included in a model change value. Simulation packages are usually divided into those applying discrete change (the state of system changes at discrete points of time), continuous change (the state of system changes continuously during simulation), or a combination of both discrete and continuous change.

3. Purpose

Purpose relates to the application area of a package, ie. the types of systems that can be simulated by a particular package (transport, communication, manufacturing, resource planning etc). Simulation packages are usually general purpose, or special purpose. Examples of special purpose packages are packages (simulators) especially designed for simulation of manufacturing systems which are the subject of this research.

4. Terminology

This criterion refers to a terminology that is used within the package to represent elements of the model. The terminology should reflect the ideas and concepts used to develop the simulation model (Pidd, 1992a). Most of the packages designed for simulation of manufacturing systems use manufacturing terminology, which facilitates establishing a correlation between real systems and their models.

5. Modelling approach

Modelling approach reflects the way of changing the state of the model as time advances and the model runs (Ekere and Hannam, 1989). The following approaches are most often used: the EVENT approach, where a system is modelled through the events that occur, the ACTIVITY approach where the start, end and duration of operations are specified, and the PROCESS approach which characterizes a system by the sequences of events and activities that a particular model element follows, and a three phase based approach, where system changes state through three phases: A (time advance), B (bound activities) and C (conditional activities). Simulation languages and packages can also use a combination of different approaches.

6. Formal logic

This criterion reports whether any formal logic such as activity cycle diagrams, flow diagrams or network diagrams, is needed for model development. Simulation packages that require the development of conceptual models using one of the graphical methods prior to development of computer models, are usually more general purpose, although they are widely used for manufacturing applications (eg. HOCUS - (Szymankiewicz *et al*, 1988), SLAM II - (Abdin, 1986), (Acree and Smith, 1985), (O'Gorman *et al*, 1986) etc.). Data-driven manufacturing simulators usually do not require any formal logic to be used for model development.

Conceptual modelling using formal graphical methods might be useful for problem understanding, regardless of the software tool used for the development of computer models. However, it is usually faster to set up a configuration of the model, when its elements can be modelled directly and positioned on the screen (in the case of visual simulation), with specification of its parameters.

7. Representativeness of model

This is a general criterion which evaluates to what extent the models are a 'natural' representation of real systems. The level of representativeness is determined by several factors such as the type of visual display, the manner of modelling the elements and their behaviour, and the interaction between these elements.

8. Ease of conceptualization of simulation logic

This criterion evaluates how easy is it to transfer the conceptual model developed by a graphical diagrammatic method to a computer model.

9. Modelling transparency

When system elements are represented as data files and their interactions as procedures or algorithms, it is difficult to understand the relationship between computer code and the model behaviour. In that case, the modelling transparency is not adequate.

10. Hierarchical model building

This is an useful feature, especially when complex systems are to be modelled. Simulation package should, for example, enable modelling at different levels, where elements at higher levels are further modelled in detail and by selecting a certain element it should be possible to have an access to a more detailed model at a lower level.

11. Run-time applications

Packages with this ability enable creation of a run-time version of the model which is itself executable (.EXE file). Such model can then be given for public use without a fear of unauthorised use of the software.

12. Conceptual model generator

This criterion evaluates whether a package is capable of producing automatically a graphical representation of the model's logic using methods for conceptual modelling such as activity cycle diagrams, Petri nets etc.

13. The length of entity name

The length of the name of model entities is usually limited to 8 characters. In very large models with more 1000 elements, it might be useful to have the possibility of using longer names of entities to ensure that the names are unique and meaningful.

14. Entity name

This criterion examines whether the user can define names of the entities, or names are provided by the package. When a user can define his/her own names, those names can be meaningful and specific to the system being modelled.

15. Experience required for software use

This criterion evaluates the level of experience needed for using a specific package. This depends on many other factors such as the ease of use of a package, quality of documentation, on-line help, tutorial and type of system to be modelled.

16. Formal education in simulation required for software use

This criterion similar to the preceding one evaluates the extent of formal education in simulation required for proper use of a package.

17. User friendliness

This is a general criterion which comprises many criteria such as a menu driven interface, animation, interaction, modelling transparency and ease of use.

18. Ease of learning

This criterion judges how easy is to learn the package. This criterion overlaps with some other criteria: quality of documentation, training course, tutorial, on-line help, demonstration models etc.

19. Ease of using

This criterion evaluates how easy it is to use the package. It also overlaps with other criteria such as: the quality of documentation, modelling assistance, menu-driven interface, prompting, on-line help, user-support. etc.

20. Initialization

Simulation package should enable the user to specify where each entity is at the beginning of simulation, what are the contents of the queues and what activities are in progress, and to specify the initial values of attributes or variables.

21. Specification of time units

When this criterion is satisfied, then the user has a possibility to specify the meaning of time units (eg. one time unit can represent one second, one hour, one week, one month or one year).

22. Specification of length measures

This criterion evaluates whether the user has a possibility to specify the measures of length displayed on the screen, which is particularly useful for modelling materials handling systems (eg. measures in feet or meters).

(ii) Visual Aspects

1. Animation

There are many publications that discuss the benefits of animation and visual interactive simulation (Bell, 1991), (Bright and Johnston, 1991), (Hurrion, 1986), (Smith and Plat, 1987) etc. Animation provides a visual display of the logical behaviour of a simulation program. In a manufacturing system, for example, the display may clearly show the moving of parts from machine to machine, moving of vehicles, or position of labour. This feature facilitates model verification, eases the process of model development, and provides better communication between simulation analyst and client.

2. Type of animation

There are two basic types of animation. When animation is full, icons are moving smoothly across the screen. For example, moving of parts along a conveyor or moving of vehicles along tracks should be performed with full animation. In the case of semi-animation, icons are jumping from state to state, or from one element to another, which can reduce an understanding of the model's behaviour.

3. Timing of animation

There are two types of timing of animation: concurrent animation and post-processed animation. In the case of concurrent animation, animation is obtained at the same time as the model is running. On the other hand, when animation is post-processed, then the model is first run without animation, and animation can be invoked after the experiment is finished.

Concurrent animation is more appropriate when the model is still under the process of development and testing, so its behaviour can be seen immediately. When the model is tested, and many replications of experiments are needed, then experiments are usually run without animation in order to speed up the experimentation. In that case it is not so important which type of timing of animation the package possesses.

4. Type of graphical display

This criterion examines how the entities are graphically presented. Presentation may vary from realistic icons, to symbolic presentations and character display which are the least realistic.

5. 3-D graphics

When this criterion is satisfied, then it is possible to obtain more realistic 3-dimensional graphical presentation of models.

6. Integrity of graphics

Two cases are usually distinguished regarding this criterion: graphics is either an integral part of the package (eg. WITNESS, XCELL+ etc), or it is added to a package (eg. SIMAN/CINEMA).

7. Animation layout development

This criterion examines when the animation layout is developed: concurrently with model development, before, or after the development of the model's logic.

8. Multiple screen layout

This criterion evaluates whether several different screens can be used to present a graphical display of the same model, which can be useful in the case of complex models.

9. Animation with visual clock

When this criterion is satisfied, movements of icons are proportional to the time needed for a change in state. For example, when different speeds and the time for movement of the vehicles are specified, this difference should be reflected on the dynamic graphical display of the model. This feature provides a more realistic view into the behaviour of the model, and it can be useful for the verification of the model.

10. Icon editor

This feature is related to animation. Some packages can only use predefined graphical symbols to represents elements of the model. On the other hand, there are packages with an icon (graphic) editor, which enables the creation and modification of user's icons. This can make a graphical display more realistic, and provide a variety of icons for different types of model elements.

11. Screen editor

Whilst icon editors are usually capable of producing icons which can move on the screen or change display according to the state of the element they represent, screen editors enable the creation of static graphical displays that enhance the model appearance on the screen.

12. Ease of icon development

This criterion examines how easy it is to develop icons. An icon editor should enable easy and rapid development of icons in a variety of shapes and colours.

13. Ease of using screen editor

When a screen editor is provided, this criterion examines how easy it is to enhance the graphical display of the model by text and/or graphical constructs using the screen editor.

14. Type of icons

This criterion examines the type of icons provided by a package. Icons can be bit mapped where each picture element corresponds to one or more bits in memory providing great flexibility in the display (Dictionary of Computing, 1986), or they can be pixel based where graphical displays are digitized in two-dimensional arrays of data.

15. Icon library

Some packages provide an icon library, which enables storage of icons in special files, regardless of which model they were created in. These icons can be accessed and used in any model, which saves time needed for the development of graphical displays.

16. Merging icon files

This feature is useful when different icons are needed for a display of complex models. Merging icon files enables the connection of different icons files created in different models into an unique icon file, which might save time needed for the development of graphical displays of complex models.

17. Resizing of icons

Once the icons are created using the icon editor, it might be realized that their size is not appropriate, when they are placed on the screen together with graphical displays of other elements. Resizing of created icons eliminates the need for the new creation of such icons.

18. Rotating of icons

Sometimes created icon should be rotated in order to adequately present a particular element of the model. This feature can be particulary useful, for example, for rotating icons that represent automated guided vehicles, which move along tracks in various directions.

19. Changing the colour of the icons

In order to obtain a better graphical display of the model, sometimes it might be convenient to change the colour of an already created icon. If the package allows it, time can be saved in developing a graphical display of the model.

20. Zoom function

This is another useful feature related to a graphical representation of the model. Zooming enables the enlarging or the reducing of the size of the model displayed on the computer screen. Reducing the size of large complex models, enables the user to view entire model, whilst the enlarging of models provides a better observation of the model's details.

21. Panning

When this criterion is satisfied, then the package allows the shifting of the viewing window (screen) on a virtual screen.

22. Switching on/off the graphics

It might be very useful if the graphics can be switched off for long repetitive runs, because the execution speed is much faster when models are run without graphics.

23. Switching between screens

When this criterion is satisfied, then it is possible to switch between different animation screens during experimentation.

24. Switching between character and icon graphics

This criterion examines whether the user can alternatively choose character or icon graphics, according to the simulation purpose. This feature might be useful because when a graphical display of models is to be obtained quickly and models run fast then character graphics can be used. On the other hand, when a more sophisticated and realistic graphical display is needed, then the user can develop icon based graphics.

25. Print screen

A print screen facility enables a printed display of the model to be obtained, as it appears on the screen. This feature might be useful, for example, for model documentation or writing a report on the simulation model developed.

26. Virtual screen

A virtual screen enables the creation of a model display that exceeds the size of the screen. This feature is very useful when a model is complex, with many elements and details. In that case, the model can be created in a dimension larger than the screen size, and then reduced by a zooming function.

27. Indication of the element's status

The status of a particular element is usually indicated by a specific colour, which might be useful for model verification.

28. Changing the colour of the elements' status display

This criterion is satisfied when the user is able to specify the colour of the elements' status display.

29. Limitation on number of displayed icons

This criterion examines whether there is a limitation on the number of icons that can be displayed on the screen. It is better if such a limitation does not exist, and the only limitation is the size of the screen.

30. Number of icons stored in icon library

This criterion examines how many icons can be stored in an icon library. The more icons that can be stored, the better, especially in the case of detailed/complex modelling in industry.

31. Change of icons during simulation

This criterion examines whether it is possible to change the graphical presentation of a model's elements after some changes occur. For example, after the parts have been assembled into a new product, a different icon might be used to represent this change.

32. Icons with multiple colours

When this criterion is satisfied, each icon used in the model can be designed using different colours, which might increase the correlation of model display to reality.

33. Easy copying of icons

This criterion examines whether it is possible to copy icons easily either by mouse, keys, or adequate menu option, once they are placed on the screen.

(iii) Coding Aspects

1. Programming flexibility

Packages that satisfy this criterion allow pieces of code to be added into the model, in order to handle complex logic. This feature is very important, because it greatly enhances flexibility of the packages, enabling the user to model a variety of different types of manufacturing systems.

2. Program generator

This criterion evaluates whether a simulation package has a program generator, which provides simulation code on the basis of the specification of model's structure and logic. Simulation program generators are particulary useful for routine model programming (Hurrion, 1991). Program generation, and especially its modification can provide greater flexibility of modelling, but on the other hand demands a considerable programming effort.

3. Access to source code

This criterion reports whether the source code of the package can be accessed. Access to the source code can facilitate the understanding of how the package is designed and how it works, but it demands a considerable knowledge of the programming language in which the package is written. This feature is also required for integration of a simulation system with a data base management system, because this integration requires programming (Larsen and Alting, 1989). The majority of a data-driven simulators do not enable access to source code.

4. Readability of source code

When the source code is accessible, this criterion judges whether this code is readable (understandable) or not. Source code readability might ease an understanding of the functioning of the package, but it is questionable how important this feature is for an user.

5. Readability of added code

Different packages that allow programming use different languages and syntax for additional coding. Some packages use special programming languages in combination with predefined functions, which requires special knowledge for coding. Readability of the added code facilitates the process of coding, and eases debugging.

6. Self-documentation of added code

This feature is a characteristic of syntax, which evaluates the general notion of meaning of the written code. Code with a poor level of self-documentation implies that the meaning and the purpose of code can not be easily detected. In this context, the modellers should use meaningful names related to the structure of the system being modelled.

7. Precision of added code

This criterion estimates the precision of added code. When code with a poor level of precision is read, then the meaning and the purpose of the code can not be easily detected.

8. Comprehensiveness of added code

This criteria evaluates the comprehensiveness of added code. When the code is comprehensive, then a relatively smaller number of commands is needed to model a certain feature.

9. Link to a lower level language

This criterion examines whether it is possible to link a package to a lower level language such as FORTRAN, Pascal or C, in order to model specific logical features, which is particularly useful when complex real systems are being modelled.

10. Data storage, retrieval and manipulation facilities

This criterion evaluates whether the package has incorporated facilities for the storage of relevant data about the model, its retrieval and manipulation in subsequent use.

11. Quality of data storage, retrieval and manipulation facilities

The type and quality of facilities for data storage and retrieval are examined by this criterion. For example, a simulation package can store data only in the form of textual files with a limited manipulation capability, but it can also store data in data bases which is better, especially in the case of large complex models.

12. Built-in functions

Providing the built-in functions when additional programming is allowed, might speed up the model development, because the user only has to call these preprogrammed functions by specifying their names and parameters.

13. User functions

This feature is also connected with programming flexibility. When a user is allowed to develop his/her own functions, the logic of the model is handled more easily, and modularity of modelling is enhanced.

14. Global variables

Global variables can be accessed by any element in the model, and are often needed when additional programming is allowed to handle complex logic.

15. Names of functions, variables and attributes

This criterion examines whether the user can define the names of logical elements. In this case, the names can be meaningful, corresponding to the problem being modelled which can facilitate model testing and understanding.

16. Writing comments for logical elements

It might be useful for model testing and understanding if the user can write comments for logical elements, especially when models are complex and detailed.

17. Type of time variable

Integer time variables can cause problems in modelling of manufacturing systems, when precise modelling is required and different operations can have considerably different durations (Ekere and Hannam, 1989). For example, the movement of an AGV can take a few seconds, while some machining operations may take a number of minutes. In this case, using integer minutes in the model means that durations of shorter activities are not precise. Therefore, real time variable seems to be more appropriate.

18. Type of translation

Compilation should be performed quickly and reliably, without substantial memory requirements. There are two types of model translation. It is either necessary to compile part of or the entire model after each change, or when translation is interpretive, then the model can be run immediately after the amendments.

19. Text/code manipulation

When additional programming is allowed, a substantial amount of code could be added to the model depending on the model complexity. In that case, the possibility of manipulating that text (code) is valuable. Cut, copy or paste facilities can significantly increase speed of coding, especially when some repetitive functions are created.

20. Length of the lines in coding editor

It might be important that a sufficient line length is provided in the coding editor. This is especially important when complex situations and features are modelled, which involve complex commands that should be preferably placed in one command line.

21. Support of programming concepts

This criterion evaluates whether the package supports typical programming concepts such as multidimensional arrays, objects, and data structuring, which can facilitate modelling special features of the system when additional programming is allowed.

22. Quality of the support for programming concepts

The variety and extent of programming concepts support is examined by this criterion.

23. Object oriented programming concepts

When the package allows additional programming, then this criterion examines whether the object oriented programming concepts such as inheritance and encapsulation are supported, which can improve efficiency and reusability of added code.

(iv) Efficiency

1. Robustness

Robust packages enable a variety of systems and characteristics to be modelled. This criterion is linked with programming flexibility, because the possibility ofr additional coding is the best way of achieving robustness.

2. Level of detail

This criterion is connected to programming flexibility and robustness. While robustness mainly relates to types of systems that can be modelled by a particular package, this criterion is focused on a level of detail that can be incorporated in each specific model. Both of these criteria are better satisfied when additional programming is allowed.

3. Number of elements in the model

A restricted number of elements that can be defined in the model, can represent a significant limitation for modelling, especially when systems to be modelled are complex, which is usually the case with real manufacturing systems.

4. Model reusability

This criterion evaluates whether a particular package enables reuse of the models created previously. Models already developed, and especially some parts that relate to complex decision logic, should be used again in other models. This feature of the package is very important, because it can speed up the process of model development.

5. Model status saving

This is an useful feature, which enables saving not only the logic of the model, but also its status after experimentation. When experiments are to be continued, the model saved with its state can be run from the point when the last experiment was finished, which can save a lot of time.

6. Automatic saving

It may be useful if package can allow an automatic saving of the model during the process of its development and modification, either at regular intervals or after each modification. This (at present not very common) feature can be particularly valuable when a package allows additional programming. In that case, after a certain amount of code is added to the model, these amendments are saved, so in the case of model 'crashing', latest changes have not been lost.

7. Interaction

Interaction facilities permits users to actively participate in experimentation with the model. This feature enables interruption of experimentation, changing the model, and continuation of the experiments in order to observe the effects of changes.

8. Adaptability

This criterion is connected with interaction. It evaluates to what extent the package is able to adapt to changes made to the model, and to continue to run immediately after an interruption of experiments.

9. Multitasking

Multitasking is an useful feature which enables performing of different operations concurrently. For example, while the experiment is running with one model, another model can be edited. This characteristic can speed up the process of model development and testing various alternatives.

10. Model chaining ie. linking outputs from different models

This feature enables using the output from one model as an input to another model. Model chaining enhances modularity, and enables complex problems to be modelled in a series of smaller models.

11. Exit to the operating system within the package

It can be frustrating when after several amendments the user wants to save the model, but instead he/she obtains the message that this is not possible due to a lack of memory. Exit to operating system, or any other possibility to create additional memory space, and return back to the model, is an important feature and it can save time spent on model development.

12. Compilation time

Compilation time does not only depend on software characteristics. It also depends on the hardware used and the size of models developed. However, compilation time should be reduced as much as possible, and it should be predictable in advance.

13. Model execution time

This criterion is evaluated by the time required to execute a program over a certain simulated period.

14. Case sensitivity

A certain package is case insensitive if it does not matter whether the user types capital or small letter for the names of model elements, or built-in functions of added programming code. This can speed up the use of the package, because the user has to remember only the names and commands, without paying attention to the type of letters used.

15. Conversion of numbers (real v integer)

This feature enables automatic conversion from real to integer numbers and vice versa, which is also practical.

16. Queuing policies

The possibility of using different queuing policies (for example FIFO, LIFO, BY ATTRIBUTE, MINIMAL VALUE, MAXIMAL VALUE etc.) provides more flexibility in modelling, because it is very likely that they should be used in modelling real manufacturing systems.

17. Number of queuing policies

This criterion evaluates how many different queuing policies are provided and easily used by a package.

18. Time scale for model building

This is a general criterion that estimates the time needed for model building. This criterion depends on many other criteria, such as the ease of learning and use, user-friendliness of the package, modelling assistance, on-line help etc. It also depends on the experience of the user, and finally on the complexity of the system being modelled.

19. Reliability

This criterion evaluates how often a package is subject to 'crashing', and unpredictable behaviour. For example, when a package allows programming and new code is added with some logical mistake, or when problems with memory occur, some packages are simply stopped. The only thing to do then is to reset the computer, whilst losing all changes made after the model was previously saved.

20. Pre-existing generic models

It may be useful when pre-existing generic models are provided, which can be then modified as necessary.

21. Merging of models

When this criterion is satisfied, it is possible to merge different models, which enhances modularity of model development.

22. Editing partially developed models

This criterion examines whether it is possible to retrieve and edit partially developed models, or whether a model should be developed again from scratch.

23. Automatic model building

When this feature is provided, then a package provides automatic guidance through model development, prompting the user to provide necessary information.

24. Ease of model editing

This criterion examines whether it is easy to modify developed models. For example, it might be necessary to change the logic of model or add new elements, and this should be done easily.

25. Specification of part flow by a mouse

When this criterion is satisfied, the user can quickly define the flow of the parts using the mouse, which enhances the efficiency of modelling.

(v) Modelling Assistance

1. Prompting

This criterion appraises whether the package provides support when additional programming is allowed. This support relates to the prompt to syntax, where the package should give advice on which command should be added to a particular position in the model. Prompting can ease programming efforts when they are required, and therefore it can speed up model development.

2. Quality of prompting

The quality of prompting is expressed by clear advice on syntax of programming constructs that should be added to the model in order to model a particular logical feature.

3. Modularity

Modularity means that it is possible to develop simulation models in separate modules, stage by stage. Each separate module can be tested and debugged more easily, and once it is correct, it can be linked with other modules to comprise an entire model. This is an useful debugging tool, but it also facilitates an understanding of the problem during the modelling process, instead of jumping directly into the full complexity and level of detail of the entire model.

4. Model and data separation

Separation of the model's logic and data that determine a particular way of running the system (eg. the number and capacity of production equipment, inter-arrival time, and quantity of parts to be processed etc.) is an important feature of the simulation package. An easy access and modification of model data enables faster, dynamic and more flexible testing of alternative production strategies.

5. Use of mouse

Use of mouse instead of keyboard in a menu driven environment, can speed up the process of model development. This feature is particularly useful when an icon editor is used, and when the graphical display of the model is set up on the screen.

6. On-line help

On-line help, easily accessible by menus or function keys, with clear and complete explanations of package facilities, is also a valuable feature which can facilitate both learning and using a package.

7. Quality of on-line help

The quality of on-line help is mainly expressed by its comprehensiveness, understandability and accessibility.

8. Documentation notes

This criterion refers to writing documentation notes and comments as the model is developed. When a model is complex, with many details and programming constructs, it is useful to write documentation simultaneously with model development.

9. Quality of facility for documentation notes

A facility for documentation notes is of good quality if the description of the model is positioned at relevant logical points, if the manipulation with text is possible etc.

10. Text editor as integral part of the package

It might be convenient if the text editor is provided within the package for creation of input files, editing text files, editing output reports etc.

11. Automatic editing of data

When this feature is provided, data are edited automatically. For example, automatic conversion of small letters to capital letters and vice versa, automatic alignment of code, automatic insertion of END commands in conjunction with IF-THEN commands etc.

(vi) Testability

1. Logic checks

When a package can detect logical errors in the models, then this criterion is satisfied. A logical check can, for example, determine that the part is to be pushed to an element that does not exist, that in probabilistic routing of parts the sum of all probabilities is greater than 1 (or 100) and so on.

2. Error messages

This criterion evaluates whether the package provides error messages. They are very important for model verification, especially in the case of large complex models where it is more difficult to detect and correct errors.

3. Quality of error messages

Characteristics of error messages are examined by this criterion. Error messages should be comprehensive, completely documented, indicating precisely where and why a mistake has occurred, and preferably how an error can be corrected.

4. Moment of error diagnostics

This criterion examines how and when the errors are detected. In this context, error diagnosis can be provided at model entry, during compilation, during model execution, or they can provide error messages at any of these events. It is believed that it is better if the error is detected as soon as it is entered to the model and corrected immediately.

5. Ease of debugging

This criterion partially covers several criteria mentioned above, because error messages, prompting, and modelling assistance in general help to prevent errors, and help to correct them when they have already occurred. But there are some other factors that can help debugging such as an ease of access to the code, the quality of the editor used for creating and editing the code etc.

6. Display of function values

Display of function values is an additional feature that facilitates model testing. For example, if a function returns the name of the element from which the part will be pulled next, and if a graphical display of the model enables observation of both the current situation in the model and the value of this function, then a logical error can be more easily detected.

7. Display of attributes

The dynamic display of the values of attributes might be useful for validation. If this feature is not provided by the package, but the display of variables is allowed, then an attribute value can be assigned to a variable and the value of that variable displayed.

8. Access to attributes

In the case when is not possible to display the values of attributes, at least it should be feasible to access those attributes easily. This is particulary important for verification of the model, when parts are scheduled according to the values of their attributes.

9. Display of variables

The display of variables is an additional feature useful for validation. It enables observation of whether the model behaves according to the variable values displayed on the screen.

10. Display of element's state

Dynamic display of the states of different model elements is a valuable feature, which enables checking of the model's logic. Changing of state is usually showed by changing the colour of the icon that represents a certain element. For example, when a machine is busy, idle or set up, a specific colour should indicate what is going on with a particular machine.

11. Dynamic display of capacity

When a dynamic display of the current number of parts in a particular element together with the total number of parts that particular element can contain is provided, a potential bottleneck in the system can be more easily detected.

12. Display of the workflow path

Display of the path that parts follow through the system enables detection of logical errors on the basis of graphical display.

13. Display of events on the screen

This criterion examines whether the package provides a display of the events on the screen, when a model is run step by step.

14. Display of part positioned within element

When this criterion is satisfied, then it is possible to monitor when parts enter a certain element, which might be useful for model verification.

15. Facility for immediate user actions

This criterion enables the user to immediately obtain some information about the model and its parameters, or information about the current values of functions, attributes etc, if they are not already displayed. Such a feature is usually used in the form of dialogue boxes, where a user can specify which information he/she wants to obtain, and it is particularly valuable for the purpose of verification.

16. List files

List files are files that contain the entire logic of the model in textual form. All model elements and their parameters are listed, together with their interaction and additional code added to the model. Such files can facilitate model testing, and writing model documentation.

17. Echo

Echo provides information about the model, listing its elements and their parameters, but generally it is not so detailed as a list file.

18. Trace files

Trace files contain information about changes in the model's state that happened in the model during simulation at different moments, which might be valuable for model verification.

19. Explode function

This feature can provide information about the current status of any element at any moment of simulation. For example, it is possible to see whether the machine is busy, and if it is how many parts are currently processed, and which are the characteristics of these parts, which might be useful for model testing.

20. List of used elements

Such a feature provides information where a specific element is used or referenced within the model. This list might be valuable for checking the credibility of the model.

21. Backward clock

This (not very common) feature can be useful for verification. For example, a long experiment can be run without animation in order to test model credibility. When the experiment is finished, and the graphical display of the current state of the model obtained, it may be realized that before the end of the experiments a logical error caused a blockage in the system or some other problem (the moment when such a problem occurs might depend on a specific combination of random number streams). It could be useful to run the model backwards to reach the moment of blockage faster, rather than to run the model from the beginning which might take several hours.

22. Step function (event to event jumping)

A step function enables the observation of the change of the model's state one step at a time. Differences that occur after each step can be examined in detail, which is very useful for verification.

23. Flow analysis

A quick analysis of the flow of materials prior to real experimentation might be useful to discover bottlenecks in the system or other problems.

24. Audible alarms

When this criterion is satisfied, then the package provides audible signals, for example, when a certain condition is satisfied or when an error occurred.

25. Rejection of illegal inputs

When this criterion is satisfied, then any illegal type or format of input data is rejected, preferably with the message why the input was rejected.

(vii) Software Compatibility

1. Integration with spreadsheet packages

Spreadsheet packages can be used for additional calculations based on simulation results, which might be particularly useful when a package does not have a facility to analyze output reports.

2. Integration with statistical packages

Statistical packages enable further analysis of simulation results, and analysis of input data for a simulation model. This feature is worthwhile especially when a package does not have statistical facilities that could provide, for example, distribution fitting or Goodness-of-fit test.

3. Integration with word processors

Integration with word processors enables modification and improving the form of output results, creation of input data (when the package can read data from the files), or modification of the list files that relate to the logic of model.

4. Integration with CAD systems

A sophisticated, three dimensional display of the models of manufacturing systems can be created with systems for computer aided design (CAD), and then used and animated with a simulation package.

5. Integration with data base management systems (DBMS)

All data related to simulation model (input data, data related to the model's logic, or output data) can be stored in a data base and retrieved by a data base management system. This is particulary useful when a simulation system is connected to the control system of the factory, and uses real data to facilitate on line scheduling or handle random events and errors.

A data base management system can perform the following functions for a simulation package: storage of models and their results for further use, retrieval of data for presentation and post-run analysis, comparison of results when multiple runs are performed or different models simulated, or it can facilitate adaptive modelling in the form of information from previous models that can be used for development of new models.

6. Integration with expert systems

Integration with expert systems may provide an intelligent assistance to model development, experimentation, or analysis of simulation results (Kochhar, 1989).

7. Integration with MRP II software

Integration with MRP II (Manufacturing Resource Planning) software can provide, for example, performing 'what-if' evaluations utilizing real operational MRP II data (Gray, 1987).

8. Integration with scheduling software

With a growing number of software packages dedicated to production scheduling, it might be useful to have the possibility to integrate manufacturing simulators with scheduling software to combine advantages of each type of software. For example, real data used for scheduling might be utilized by a simulator to test the influence of changes in configuration to production scheduling.

(viii) Input/Output

1. Menu driven interface

A menu driven interface is important feature of user-friendly simulation packages. It speeds up and eases the process of model development, when a user can select options provided by menus. This approach represents a significant improvement, compared to the conventional approach where a user has to type all commands.

2. Pull down menus

This type of menu provides a listing of all options within a certain menu being selected.

3. Type of menu selection

This criterion examines how menus can be selected. Most often, menus are selected by keyboard keys or mouse.

4. Selection buttons

When this feature is provided, then different modelling options can be selected by selection of a corresponding button (usually by mouse), which can speed up model development.

5. Dialogue boxes

A dialogue box is a window which displays a series of controls, which usually appears in response to menu commands or points to some important condition, expecting the user to react. For example a dialogue box can ask the user whether he/she wants to save the model before leaving the system. This feature represents an useful assistance during the use of a package.

6. Multiple inputs

This criterion evaluates whether the package enables independent multiple arrivals of different part types, which can be very important when complex systems are modelled.

7. Multiple outputs

This criterion examines whether the package provides independent and simultaneous multiple outputs of different part types.

8. General output reports

This criterion evaluates the types and the variety of standard output reports such as queue lengths, waiting times, utilization of servers etc.

9. Static graphical output

Static graphical report relates to a graphical representation of simulation results in the form of histograms, timeseries, bar charts, pie charts etc. These graphical representations are obtained after simulation experiments. Graphical presentation is very valuable, because it quickly gives an impression about the measures of performance.

10. Dynamic graphical output

Dynamic graphical output provides a graphical display of specified measures of performance during the experimentation. This feature gives a prompt view into the behaviour of the model, and it is very useful for model testing.

11. User defined output

In addition to the reports provided by software, it is useful for the user to be able to request his/her own reports, depending on the purpose of simulation.

12. Automatic rescaling of axis Y in time series and histograms

Automatic increasing of axis y in graphical displays of simulation results is needed especially in the case of long experiments. It is not easy to predict in advance which maximum value of a certain measure of performance will be reached during experimentation.

For example, the value on axis y can represent the total throughput in a manufacturing system. In that case the maximum value reached depends on the duration of experiments, but it also depends on the set of random number streams. Therefore, it is more convenient if the maximum value displayed is increased automatically.

13. Quality of output reports

This criteria examines the characteristics of output reports: how they are presented, when they are obtained, whether they contain relevant information etc.

14. Understandability of output reports

Understandable output reports provide fast perception of model behaviour and credibility. Graphical presentation of results is an additional aid in presentation of results, but the results in numerical (and/or textual) form have also to be clearly presented and described.

15. Periodic output of simulation results

This is an useful feature, which might be used for the determination of steady state. A package should provide a specification of the interval after which output data will be written preferably to a special file. When this feature is not provided in a package, the user has to run the model for a certain period of time, observe and note results, run the model again for the same period of time, check the results and so on.

16. Availability of results before end of simulation

This criterion examines whether it is possible to obtain simulation results before an experiment is finished.

17. Input data reading from files

Reading ASCII files with input data speeds up the process of experimentation, which is particularly useful when a number of alternative data will be read and tested.

18. Writing reports to files

This feature of the software may be very useful, especially when many experiments are performed with large models, and a large quantity of output data is obtained. Writing output data to files can facilitate subsequent analysis and use of this data, preferably in integration with other software systems such as DBMS or spreadsheet packages.

19. Writing reports to printer

When this criterion is satisfied, the results can be printed as they are obtained.

20. Writing reports to plotter

This criterion examines whether the results can be sent to a plotter directly from the package.

21. Snapshot reports

This reports give information on the number and type of the parts currently in the system, and the current state of the element. The information provided by this report are similar to those obtained by the explode function, but not so detailed because individual values of part attributes are not displayed.

22. Summary reports for multiple runs

When this criterion is satisfied, a package provides summary reports for relevant measures of performance obtained during multiple independent runs.

(ix) Experimentation Facilities

1. Automatic batch run

This feature enables specification and automatic control of many replications of experiments, which enables faster running of a number of experiments, even when the analyst is not present.

2. Warm-up period

It is not typical to start simulation and collection of statistics when a system is empty. Therefore, it is advisable to run the model for a certain period of time (to warm it up), achieve the steady state of the system, reset statistics, and continue with experimentation.

3. Re-initialization

When this criterion is satisfied, the user can reset the statistics of the model at any time during experimentation.

4. Re-start in non empty state

This criterion examines whether the experimentation can be resumed in non empty state of model (after the model has been warmed up).

5. Breakpoints

This feature allows user to specify run length, and stop the simulation at a specified time or event. For example, progress of the system can be observed, model changed or animation turned on or off. This feature should be provided to allow the model to proceed repeatedly from specified decision points.

6. Speed adjustment

It is useful when a different speed of model running/animation can be specified. A slower speed can be used when model is still tested, while experiments can be run on the fastest possible speed, when model verification is finished and several and/or long replications are being made.

7. Experimental design capability

This criterion examines whether the software provides assistance for experimental design. This feature is especially helpful when there are many variables in the model that can influence the performance of the system.

8. Quality of experimental design facility

The quality of experimental design facility might be expressed by the level and comprehensiveness of control and advice on possible alternatives to be tested.

9. Accuracy check

It is useful if the software is capable to provide information about the accuracy of simulation output, obtained on the basis of several runs. This value might indicate that the run length is not sufficient, or there are some problems with the model parameters.

10. Automatic determination of run length

This criterion examines whether the package is capable to determine the run length that will provide the results of an adequate accuracy.

(x) Statistical facilities

1. Theoretical statistical distributions

It is necessary that the package allows a number of different statistical distributions to be used within the model. When real data are fitted into a theoretical distribution, it is important for the precision of estimations of measures of performance that random variables are indeed sampled from such a distribution.

2. Number of theoretical statistical distributions

When large complex models are developed, it is likely that a variety of different theoretical statistical distributions will be needed. Therefore, it might be useful if many of these distributions are provided.

3. User-defined distributions

When no theoretical distributions fit the real data, then the user has to specify his/her own distribution. A proper package has to allow for this.

4. Random number streams

A variety of different random number streams is needed to perform many replications of experiments with different random number streams for random variables, in order to obtain more precise estimations of measures of performance.

5. Number of different random number streams

When large models incorporate many sources of variation, and when several independent runs are to be performed, then it might be useful if a sufficient number of different random number streams is provided.

6. User specified seeds of random number streams

This criterion examines whether the user can specify seeds for random number streams.

7. Antithetic sampling

Antithetic sampling (for random number RN, antithetic sample is 1-RN) provides information whether the model is sensitive to a certain random number selection. If the results from both runs (with and without antithetic sampling) are similar, then the model is not sensitive to a certain selection of random numbers.

8. Distribution fitting

When real data are collected, it is convenient if an appropriate theoretical distribution can be determined and data analyzed by the simulation package.

9. Goodness-of-fit tests

It might be useful if a package can undertake the Goodness-of-fit testing. Such tests can be used to check how real data fits a theoretical distribution, or to what extent simulation results are close to the performance of a real system.

10. Output data analysis

Thee ability to analyze a series of simulation results, and produce a variety of statistical reports on multiple runs, is very valuable but not a very common feature. A lot of time can be saved if, for example, a package can give mean values, variances or confidence intervals for selected measures of performance obtained during many replications of simulation experiments.

11. Quality of data analysis facility

This criterion evaluates a level of data analysis that is provided, including the type of analysis that is provided, the understandability of the results of analysis, the number of different reports that is provided etc.

12. Confidence intervals

This criterion examines whether a package provides an estimation of confidence intervals for relevant measures of performance obtained in multiple runs, which might be useful for the evaluation of accuracy of simulation results.

(xi) User Support

1. Documentation

Documentation is very important. It facilitates learning and using of the package. Understandable and comprehensive documentation can save a lot of time and effort involved in the full utilization of a particular package.

2. Quality of documentation

This criterion is determined by several factors such as type of presentation (story like, very technical etc), the level of detail included, usefulness of index (if it is provided), the number of examples provided etc.

3. Reference card

It might be useful if a supplier provides a reference card with all the main information that can facilitate the use of a package such as the main commands, elements, and in-built functions.

4. Demo disks

Demonstration disks might be useful for software testing, or for research purposes.

5. Tutorial

A tutorial which can lead the user through the process of learning how to use a package is an useful feature, especially when a package is comprehensive.

6. Training course

Attending the training course where tuition is provided by the expert with substantial experience in using the package, is worthwhile especially when the package is comprehensive and not easy to learn.

7. Duration of training courses

In the case when a package is difficult to learn, then the duration of the course should be sufficient.

8. Frequency of training courses

When the number of new users of a certain package is constantly increasing, then it is useful to organize training courses regularly in shorter periods (for example each month).

9. Demo models

A number of different demonstration models are necessary for the first impression about the package. They could be also useful as examples of how certain features can be modelled, when a need for modelling such characteristics arises.

10. Help line

It is very useful if the supplier provides a help-line, where a specialist can be contacted when some problems occur, or when some special features have to be modelled.

11. User group meetings

It may be beneficial to attend user-group meetings, where it is discussed how to overcome deficiencies of the package, how to perform modelling more efficiently, to learn about plans for the release of a new software version, or behold different case studies where a particular package was used.

12. Frequency of user groups meetings

User group meetings should be held on a regular basis (eg. every few months), and at each meeting users should be informed about the time of the next user group meeting.

13. Newsletter

Supplying the users with a newsletter, which contains information about a release of a new version of the package, presentation of case studies where a particular package was used, or information about the user group meetings, might be useful.

14. Package maintenance

It is worthwhile if the supplier provides the user with updated versions of a package that contains some additional facilities not available before.

15. Consultancy

This criterion examines whether a supplier provides a consultancy services, which might be especially useful in the case of complex and detailed modelling in industry.

(xii) Financial and Technical Features

1. Portability

Portability means that packages can be used on different types of computers (with different operating systems). This might significantly reduce the costs of purchasing additional hardware, and provide more flexibility in using a package.

2. File conversion

Automatic conversion of files (models) created using different versions of software is a valuable feature, especially when nowadays an interval between releases of new software versions is decreasing.

3. Price

It is not easy to judge this criterion. However, improving the quality of the packages should increase the market for their use, which should lead to reductions in price.

4. Installation costs

This criterion evaluates whether additional funding is needed for the installation of the package.

5. Ease of installation

This criterion assesses how easy or difficult it is to install a package.

6. Hardware requirements

This feature relates to type of hardware that is needed for the use of a package (for example work station or PC, RAM, disk space, display facilities etc.).

7. Availability of package on standard hardware

This criterion examines whether a package is available on standard hardware platforms.

8. Availability of package on standard operating systems

This criterion explores whether the package is available for standard operating systems.

9. Version of software for network

This criterion examines whether it is possible to use a particular package on a network, which might be particularly useful when software is used for education.

10. Virtual memory facility

It may be very useful, especially when memory is limited, if the package is able to create virtual memory and extend the amount of memory available for its use.

11. Security device

A security device limits the use of a package to only one user at a time. Although it is not possible to prevent unauthorised use without such device, it should be in the supplier's interest to extend the number of users (especially in academic institutions), and create more potential users in the future.

12. Free software trials

This feature can be very useful especially when the user wants to try several different packages prior to purchasing, or when different packages are to be evaluated for research purposes.

13. Free technical support

This criterion examines whether a supplier provides free technical support, which might include package installation, assistance in case of technical problems etc.

Description of evaluation criteria

14. Types of contracts available

This criterion evaluates which types of contracts are provided by the supplier.

15. Educational discount

Although this criteria can be considered as a part of the criteria related to the types of contract available, it is listed separately due to its significance. It is very important if educational institutions can get a substantial discount on the price of the software. This will not only increase the present use of such software, but it might also create many potential users in the future, because some students will continue to use software after graduation and perhaps recruit new users.

16. Quantity discount

This criterion examines whether a supplier provides a discount when larger amounts of software licences are purchased.

17. Life cycle maintenance costs

This criterion evaluates whether a package requires considerable maintenance costs.

18. Price of training course

This criterion examines how expensive it is to attend a training course provided by the supplier.

19. Consultancy fees

This criterion evaluates how expensive it would be to use consultancy services provided by the supplier.

20. Frequency of update

This criterion evaluates how often the supplier releases an updated version of the software, which is important when users' requirements for simulation software are constantly increasing.

21. Comprehensiveness of update

This criterion examines how extensive is the update of a package. This means that it should be assessed how many new features are incorporated and how good these features are.

(xiii) Pedigree

1. Age

This criterion examines when a package was first released on the market.

2. Genealogy

This criterion explores the origin of the package, how its development started, and which software (programming language) has been used for its development.

3. Spread

The width of use (the number of users) is examined by this criterion.

4. Success

This criterion examines the record of successful use of a package in a variety of simulation studies.

5. Availability of references

It is significant when there are many references available on a particular software, especially those that describe achievements provided by the use of a package.

6. Reputation of supplier

The good reputation of the supplier regarding the manner of performing the business and the type and level of user support provided might be important features to consider when purchasing a certain simulation package.

7. Sources of information about the package

This criterion examines which sources of information about a package are available. Perhaps the best ones are those provided by other independent users who can provide information regarding both the positive features and weaknesses of the software.

I.2. CRITERIA SPECIFIC FOR MANUFACTURING SIMULATION PACKAGES

Criteria within this group are also grouped according to their nature. There are four groups ((i)-(iv)) of criteria specific for the evaluation of manufacturing simulation packages. A general description of these groups is provided in Chapter 4.

(i) General Manufacturing Modelling Features

1. Problem areas tackled

This criterion appraises whether a package can be used only for simulation of traditional manufacturing systems, or it can be also used for simulation of specific type of manufacturing systems such as flexible manufacturing systems, automated storage retrieval systems, material handling systems, warehouses, production lines etc.

Whilst packages that enable modelling traditional manufacturing systems incorporate basic features such as parts, machines or labour, packages dedicated for simulation of specific types of manufacturing systems enable modelling of special elements such as guided vehicles, pallets, robots etc.

2. Applicability for manufacturing systems

This general criterion examines to what extent a certain package is suitable for modelling manufacturing systems. Many factors influence this criterion such as terminology used, physical elements typical for manufacturing provided, scheduling facilities, measures of manufacturing performance obtained after simulation etc.

3. Equipment breakdown modelling

Modelling of breakdowns that occur in random intervals may be essential in order to assess performance of the system, especially when production equipment is subject to failures.

4. Type of breakdowns

This criterion examines which types of breakdowns can be modelled. For example, clock based breakdowns occur on the basis of simulation time, usage based breakdowns occur on the basis of active use of a certain element, cycle based breakdowns occur after a certain number of cycles have been finished on a particular machine etc.

5. Machine setup modelling

In many real systems it might be necessary to prepare machines before new type of parts can be processed (for example, a tool has to be changed, machine cleaned etc.). Therefore, it is worthwhile if a package enables modelling of this activity both when one batch is finished and when new batch already arrives at a machine.

6. Machine teardown modelling

This not very common feature enables modelling of activities needed to be performed after processing a certain batch and before setup for the new batch begins. In the case that package does not enable modelling of this situation, time needed for this activity might be added to time needed for setup.

7. Rejects modelling

Another important feature to be modelled in many systems is rejection of output that needs rework. This feature is usually modelled according to the specified probability.

8. Capacity of manufacturing equipment

A package for manufacturing simulation should enable the user to easily specify the capacity of manufacturing resources such as machines, buffers, conveyors etc. Although this feature may be taken for granted, it is listed here because of its significance in assessing appropriate manufacturing capacity.

9. Shifts modelling

When a package enables specification and simulation of different shift patterns which precisely describe working hours and break time for each type of labour, it is possible to assess more accurately how many workers are needed, and in how many shifts the factory should operate.

10. Maintenance modelling

When system maintenance occurs at regular intervals, it may be worthwhile to monitor this process in order to see how much it reduces production, and what impact it has on labour utilization.

11. Automatic increasing of buffer capacity

In many cases due to the random behaviour of a system it is not possible to assess the buffer capacity which is sufficient to accommodate all parts to be stored in the buffer. If the capacity is not automatically increased during the simulation, blockages and other problems may occur. For example, only part of the batch that is arriving in the system might be placed in a buffer if its free space is smaller than the batch size.

It might be more convenient if capacity is increased automatically in order to avoid this problem, and this information about the increased capacity is reported to the analyst, so a better estimation about the needed capacity could be made in future.

12. Buffer delays

Sometimes it might be necessary to model situation when parts have to spend some time in buffer before they can leave it. For example, parts may be prepared for processing while they wait in buffer.

13. Job lists

A list of jobs performed by each type of labour can be useful both for the assessment of labour performance and for model verification.

14. Part attributes modelling

This is an essential feature because in the majority of manufacturing systems parts have a variety of different characteristics, which have to be used to control the logic of the model. For example, parts can be routed to different machines, depending on their size, weight, colour, priority etc.

15. Frequency of part arrival modelling

This criterion examines whether it is possible to specify the frequency of arrival of a particular part type.

16. Arrival of part in batches

This criterion examines whether it is possible to model the arrival of parts in batches, rather then to model them arriving individually.

17. Type of part arrival

The type of part arrival that can be modelled by a package is examined by this criterion. Parts can be either generated on a regular basis, or created when they are needed in the model.

18. Variable conveyor speed

This criterion examines whether it is possible to model variable conveyor speed, depending on the current conditions in a system.

The following criteria evaluate whether some operations, typical for certain types of manufacturing systems can be modelled. A simulation package should either possess the facility for explicit modelling of these operations, or it should be easily adapted to enable modelling of these features:

19. Assembly operation modelling

This operation refers to the case when several parts are connected together producing one part as a result of this operation.

20. Disassembly operation modelling

When one part is detached into separate components, several parts are obtained as a results of this operation.

21. Containerization modelling

This operation relates to storing the parts in containers and transporting these containers within the system, or removing them from the system.

22. Fixture modelling

This operation relates to attaching the parts to the pallets by fixtures.

23. Palletization modelling

This operation relates to placing parts to pallets which serve as trays on which the parts are transported through a system.

24. Evaporation of fluids modelling

This operation relates to changing into vapour (disappearing) of fluids.

25. Precipitation of fluids modelling

This operation is to be modelled when there is an occurrence of separation of a solid substance from the liquid in which it is held.

26. Fluid composition modelling

This operation relates to modelling of the mixture of different types of fluids with different characteristics.

27. Inspection operation modelling

When this operation is modelled, then two types of output are obtained: parts with good quality, and rejected parts which did not satisfy the quality standard.

(ii) Physical elements

1. Single machines

Machines of this type are capable of taking one part and producing one part at a time. Examples are machines that process large components in the machine tools industry, car industry etc.

2. Batch machines

Batch machines are capable of processing many parts at the same time. An example of this type is an oven which can be used for baking painted components.

3. Production machines

Machines of this type take one component and produce several components. An example of such a disassembly process might be the case when several parts jigged together and processed as one unit are separated on a special station.

4. Assembly machines

The opposite situation arises when assembly machines have to be used. These machines take many parts and produce one part. Many examples of assembly processes can be found such as assembling the components in the car industry, computer industry etc.

5. Multi-cycle machines

This type of machines is used when one or several parts have to be processed on the same machine in multiple stages. An example of this situation is the case when parts are processed on one side in the first cycle, and on the other side in the second cycle.

6. Multi-station machines

When there are several identical machines with identical parameters, modelling them as a multi-stations machines could be useful, and could save modelling time.

7. Buffers

Buffers which represent storage space are a very common feature in many manufacturing systems. Therefore, detailed modelling of buffers may be needed in various situations.

8. Workstation buffers

Whilst buffers usually represent common storage areas, a workstation buffer is a special storage area provided for workstation.

9. Labour

A detailed modelling of labour requirements is important especially when there are many different types of labour utilized in the system.

10. Automated guided vehicles (AGVs) and trucks

Automated guided vehicles are common element in advanced manufacturing systems, so package used for simulation of these systems should enable modelling of vehicles and their paths.

11. Conveyors

Conveyors are another common feature of advanced manufacturing systems. Two basic types of conveyors are queuing or accumulating conveyors where parts are accumulated on the conveyor until its capacity is reached, and fixed conveyors where the distance between parts is fixed, and if the part cannot be pushed from the first position then entire conveyor becomes blocked.

12. Types of conveyors

This criterion examines which types of conveyors can be modelled by a package. The most common types are queuing conveyors where parts can accumulate and queue for further progress, and conveyors with carriers where parts are loaded on the carriers which maintain a fixed distance between parts.

13. Branching and looping of conveyors

This criterion examines whether it is possible to model branching, where conveyors are divided into logical sections, and looping, where conveyor systems are designed as closed loops.

14. Conveyor buffers

If pallets are moved through a system on conveyors, then a special storage place is needed to store pallets between conveyor and work stations. This feature can also be modelled using ordinary buffers.

15. Fork-lifts

This is another special means of transport in manufacturing systems.

16. Robots

This is a special feature typical for automated manufacturing systems. Robots can be used both for transport and machining operations.

17. Automated storage retrieval system

This feature is also typical for automated manufacturing systems. A computer operated, automated storage retrieval system might be included in a flexible manufacturing system for the purpose of fixture stores, or to store raw material and finished products.

18. Tools

Tools are an inherent part of computer numerically controlled (CNC) machines, which are often used in automated manufacturing systems (and especially in flexible manufacturing systems). For example, when the needed number of specific types of tools has to be assessed, a package should allow the modelling of this in a sufficient level of detail.

19. Automated tool storage

It might also be necessary to assess the capacity of an automated tool storage. The package should allow this. When this facility is not available, this feature can be modelled approximately using buffers.

20. Pallets

This is a common feature in flexible manufacturing systems. Pallets are used as trays on which parts are positioned, until they became final products.

21. Fixtures

Fixtures are used to fix a part on a pallet, so they can be safely transported and processed.

22. Fixture stores

In the case that it is necessary to assess the capacity of place where fixtures are stored, this feature might be needed. However, it can be also approximately modelled using buffers.

23. Pallet shuttles

Pallet shuttle is a rotary mechanism with usually one inner position that interacts with a work centre, and one outer position that interacts with the vehicle (Carrie, 1988), in order to transfer a pallet to and from machining operations. A special package dedicated for FMS simulation should enable this element to be modelled in detail, but if not, this feature could be also modelled using buffers.

24. Carousel-type magazines

This element is very similar to pallet shuttles, with the difference that there are usually several inner and outer positions, that can accommodate even more than ten pallets. In the absence of this element, such a component of the model can be also modelled using buffers.

25. Cranes

In some systems cranes might be used as a means of transport. It might be useful if in that case a package enables explicit modelling of such a feature.

26. Tanks and fluids

This feature is needed when a system to be modelled contains fluid and tanks, which might be the case in the chemical industry, food industry etc.

(iii) Scheduling Features

1. Scheduling rules

Different scheduling rules describe different ways of running a factory. For example, a scheduling rule has to decide which part will be processed next on a particular machine, which machine will be used for processing a particular part, or which vehicle will be used for the transport of parts. It is valuable if a package provides a number of different scheduling rules to be selected, because this can save time needed for model development.

2. Number of scheduling rules provided

This criterion evaluates how many different programmed scheduling rules are provided and easily used by a package.

3. Remaining processing time calculation

Calculation of remaining processing time might be useful for scheduling purposes, especially if such a feature is not provided by one of the scheduling rules provided.

4. Conditional routing

In real manufacturing systems, it often happens that parts have to be routed according to some conditions. For example, the values of particular attributes, variables, or functions have to be checked and depending on these values, parts are pushed to or pulled from specific elements. A simulation package should enable the user to specify these conditions, because even if preprogrammed scheduling rules exist, they can not incorporate all possible situations and conditions.

5. Priority

The possibility to specify different priorities for model elements such as parts, machines, and conveyors, is a useful feature for modelling situations that arise in real manufacturing systems. For example, different priorities can be assigned to parts according to their due dates.

6. Preemption

This feature enables the interruption of a certain operation, and transfer of resources to another element of the model (eg. machine) in order to perform an operation with a higher priority.

7. Push/pull from specific positions within element

In many situations in real systems it is necessary to push to or to pull a part from a specific position within an element (machine, buffer, conveyor etc.). For example, it has to be checked which positions within the buffer contain parts with certain values of attributes, and then parts should be pulled from these positions.

8. Specification of quantity of parts to be moved between elements

This criterion examines whether it is possible to model a situation when several parts are moved together between certain locations (elements) in the model.

9. Batch index

Assigning an unique index to each batch within the storage space can facilitate manipulation of the batch, because the batch can be treated as one unit. If such an assignment is not possible, each individual part within the batch has to be accessed. This feature can be useful, for example, when a complete batch whose parts have specific values of attributes has to be removed from any position within the storage area, and placed in another model element (buffer, conveyor, machine etc.).

10. Predefined part routing

This criterion relates to a predefined path for a particular part type, used for routing the parts through the model. When there is a significant number of parts or batches that follow the same route in the system, a specification of their route in advance could save some time both in model development and in experimentation. In that case it is not needed to specify and test all complex conditions for routing of a particular batch, because all destinations for parts are already defined.

11. Routing restrictions

When routing restrictions do not exist, parts can be pulled or pushed between different elements regardless of the type of these elements. Examples of restrictions are when parts cannot be pushed from one conveyor to another, or when buffers are passive without the ability to push or pull parts. A simulation package should not contain such restrictions, because they can complicate modelling of certain features that might occur in real systems.

12. Type of part sequencing

This criterion examines which types of part sequencing are provided by a package. Examples are probabilistic, conditional and deterministic part sequencing.

13. Departure schedule for shipping area

It might happen that the departure of finished products has to be modelled in more detail, rather than simply push the part out of the system once it is finished. For example, a customer may want to know how often, and with how many vehicles with a certain capacity the products have to be dispatched in order to avoid a significant level of inventory. In that case, simulation packages should allow this feature.

14. Vehicles scheduling

When vehicles are used in the system, it should be possible to model different scheduling strategies for these vehicles, in order to determine which vehicle will go to which destination.

15. Vehicle acceleration and deceleration

For a more precise assessment of vehicle performance and its impact on the performance of entire system, it might be useful to model vehicle acceleration and deceleration.

16. Scheduling optimization

When this criterion is satisfied, a package has a capability to calculate an optimal production schedule which has, for example, a minimal production time or maximal throughput, on the basis of specification of alternative part routing.

(iv) Manufacturing Performance

1. Throughput

This measure of performance provides information about the number of parts produced in a simulated period, and because of its importance should be provided in any package used for manufacturing simulation.

2. Work in progress

Information about the level of inventory in the system is also important for the assessment of manufacturing system performance.

3. Utilization of production equipment

The percentage of simulated time which a particular machine spent busy, idle, being setup or broken is also useful information.

4. Makespan

The time spent by a part in the system from the moment of its arrival as raw material to the moment of leaving the system as a final product, also gives an useful hint about the manufacturing system performance.

5. Special user-defined reports

In addition to standard manufacturing reports, the user should be allowed to specify his/her own special reports such as information on lateness of the parts, duration of failures, a number of specific operations and so on.

6. Due dates monitoring

Although some packages may alow the user to request monitoring of due dates performance by adding the code, this criterion evaluates whether the package has the facility for automatic recording of information if the parts are processed on time, or they are processed after the due date.

7. Manufacturing costs analysis

When several manufacturing strategies are being tested, it might also be needed to assess financial performance of each strategy.

8. Schedule related output

Although there is a growing number of packages designed especially for manufacturing scheduling, it might be useful if a package can provide some basic reports related to scheduling such as Gantt charts.

9. Transportation time of the parts

Calculation of transportation time of the parts might be useful, especially when the layout of the system and performance of transport facilities are to be assessed.

10. Rework and scrap level

It might be useful if a package can provide information about rework and scrap level, especially when a company is concerned about the quality of products.

11. Interruption report

It might be useful if a package provides a report on interruptions due to breakdowns, maintenance, breaks in shift patterns etc.

12. Production sequence summary

This criterion examines whether it is possible to obtain a summary information about the sequence of production of all part types in the system.

QUESTIONNAIRE

NAME: ______________________________

COMPANY: ______________________________

ADDRESS: ______________________________

____

____

TEL: ______________________________

FAX: ______________________________

Please answer the following questions (for questions 2,4,6,7,8,9 and 10 please use a separate sheet of paper if necessary)!

QUESTIONS:

1. Which kind of simulation software is used in your company?

   a. simulators YES NO

   b. simulation languages YES NO

   c. other (please specify): ______________________________

   ____

2. If you use simulators, which one (ones) do you use?

   a. WITNESS YES NO

   b. SIMFACTORY YES NO

   c. SIMAN/CINEMA YES NO

d. ProModelPC	YES	NO
e. XCELL+	YES	NO
f. INSTRATA	YES	NO
g. other (please specify):	________________	

3. Do you use simulation for:

a. modelling real systems	YES	NO
b. education	YES	NO
c. both a. and b.	YES	NO
d. other (please specify):	________________	

4. What is your general opinion about each simulation package being used in your company?

package: ________________

opinion: ________________

package: __________

opinion: __________

package: __________

opinion: __________

package: __________

opinion: __________

5. Which types of systems have you simulated with these packages?

   a. manufacturing YES NO

   b. other (please specify): ______________________

   ____

   ____

6. Did you manage to model all features of the system you wanted to include in the models or did you have to make considerable approximations due to software limitations? If yes, please give some details (eg. whether the results were significantly influenced by those approximations).

____

____

____

____

____

7. What are the main weakness and limitations of these software packages?

____

____

____

____

____

____

____

____

____

8. What are the most important positive features that the simulation packages you have used so far possess?

9. What are the most important features that you would like to be included in the existing simulation packages that are not yet provided?

10. Other comments

Please send the completed questionnaire to:

Ms. Vlatka Hlupic

The London School of Economics
Information Systems Department
Houghton Street
London WC2A 2AE

FAX: 071- 955 7385

REFERENCES

REFERENCES

Abdin P.F. (1986), "Solution of Scheduling Problems of Job-Shop Type FMS with Alternative Machine Tools", *Computers and Industrial Engineering*, 2:1-4, *Proceedings of the 8th Annual Conference on Computers and Industrial Engineering*, Pergamon Press, pp. 241-245.

Acree E.S. and Smith M.L. (1985), "Simulation of a Flexible Manufacturing System -Applications of Computer Operating System Techniques", *Proceedings of the 18th Annual Simulation Symposium*, (Tampa, Florida), IEEE Computer Society Press, Los Alamitos, pp.205-216.

Au G. (1990), "*A Graphics Driven Approach to Discrete Event Simulation*", Ph.D. Thesis, University of London, London.

Avgerou C. and Cornford T. (1993), "*Developing Information Systems - Concepts, Issues and Practice*", The Macmillan Press Ltd., Basingstoke, U.K.

Banks J., Aviles E., McLaughlin J.R., and Yuan R.C. (1991), "The Simulator: New Member of the Simulation Family", *Interfaces*, 21:2, pp.76-86.

Bassett E. (1989), "A general review of powder coatings", in "*Thermoset Powder Coatings*", John Ward (ed.), FMJ International, Publications, Redhill, UK, pp.22-24.

Bell P.C. (1991), "Visual interactive modelling: The past, the present, and the prospects", *European Journal of Operational Research*, 54, pp.274-286.

Ben-Arieh D. (1986), "Manufacturing System Application of A Knowledge Based Simulation", *Computers & Industrial Engineering*, 11:1-4, pp.459-463.

Bevans J. (1982), "First, choose an FMS simulator", *American Machinist*, May 1982, pp.143-145.

Bollino A. (1988), "Study and realisation of manufacturing scheduler using FACTOR", in the *Proceedings of the 4th International Conference on Simulation in Manufacturing*, (November 1988, London), IFS Conferences, New York, pp.9-20.

Bovone M., De Ferrari V., and Manuelli R. (1989), "How to choose an Useful Simulation Software", in the *Proceedings of the 1989 European Simulation Multiconference*, (June 1989, Rome, Italy), Simulation Computer Society, San Diego, pp.39-43.

Bright J.G. and Johnson K.J. (1991), "Whither VIM? - A developer's view", *European Journal of Operational Research*, 54, pp.357-362.

Carrie A. (1986), "The Role of Simulation in FMS", in "*Flexible Manufacturing Systems: Methods and Studies*", Kusiak A. (ed.), Elsevier Science Publishers B.V., pp.191-208.

Carrie A. (1988), "*Simulation of Manufacturing Systems*", Wiley, Chichester.

Cellier F.E. (1983), "Simulation software: Today and tomorrow", in the *Proceedings of the IMACS International Symposium*, (May 1983, Nantes, France), Burger J. and Jarny Y. (eds.), Elsevier Science Publishers, Amsterdam, pp.3-19.

Čerić V. and Hlupić V (1993), "Modelling of the Solid Waste Processing System by Discrete Event Simulation", *Journal of Operational Research Society*, 44:2, pp.107-114.

Chikofsky E.J., Martin D.E., and Chang H. (1992), "Assessing the state of tools - assessment", *IEEE Software*, May 1992, pp.18-21.

Christy D.P. and Watson H.J. (1983), "The Application of Simulation: A Survey of Industry Practice", *Interfaces*, 13:5, October 1983, pp.47-52.

Clark M.F. (1991), "WITNESS: Unlocking the power of visual interactive simulation", *European Journal of Operational Research*, 54, pp.293-298.

Cornford T. (1992), "Methodologies: What, Why and Where", in the *Information Systems Design Methodologies notes*, The London School of Economics, London.

Conway R. and Maxwell W. (1987), "Modelling Asynchronous Materials Handling in XCELL+", in the *Proceedings of the 1987 Winter Simulation Conference*, Thesen A., Grant H. and Kelton W.D. (eds.), (December 1987, Atlanta, Georgia), Simulation Computer Society, San Diego, pp.202-206.

Conway R., Maxwell W., McClain J., Worona S. (1990), "*User's Guide to XCELL+*", Release 4.0, 3rd Edition, The Scientific Press, South San Francisco.

Davis L. and Williams G. (1993), "Evaluating and Selecting Simulation Software using the Analytic Hierarchy Process", forthcoming in the *Integrated Manufacturing Systems*.

Deaver R.A. (1987), "Selecting a Manufacturing Simulation System", *CIM Review*, 3:3, pp.6-8.

Ekere N.N. and Hannam, R.G. (1986), "A comparison of two approaches to the simulation of manufacturing systems", in the *Proceedings of the twenty-sixth International Machine Tool Design and Research Conference*, (September 1986, Manchester), Davies B. (ed.), National Machine Tool Builders Association, pp.185-190.

Ekere N.N. and Hannam, R.G. (1989), "An evaluation of approaches to modelling and simulating manufacturing system", *International Journal of Production Research*, 27:4, pp.559-611.

Farnsworth K.D., Norman V.B. and Norman T.A. (1987), "Integrated software for manufacturing simulation", in the *Proceedings of the 1987 Winter Simulation Conference*, Thesen A., Grant H. and Kelton W.D. (eds.), (December 1987, Atlanta, Georgia), Simulation Computer Society, San Diego, pp.195-201.

Gelenbe E. and Guennouni H. (1991), "FLEXSIM: A flexible manufacturing system simulator", *European Journal of Operational Research*, 53, pp.149-165.

Ghandforoush P., Huang P.Y. and Taylor B.W. (1985), "A criteria decision model for the selection of a computerized manufacturing control systems", *International Journal of Production Research*, 23:1, pp.117-128.

Gilman A.R. and Billingham C. (1989), "A Tutorial on SEE WHY and WITNESS", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.192-200.

Goble J. (1990), "Introduction to SIMFACTORY II.5", in the *Proceedings of the 1990 Winter Simulation Conference*, Balci O., Sadowski R.P and Nance R.E. (eds.), Simulation Computer Society, San Diego, pp.136-139.

Grant F.H. (1988), "Simulation in Designing and scheduling manufacturing systems", in "*Design and Analysis of Integrated Manufacturing Systems*", W.D. Compton (ed.), National Academy Press, Washington D.C., pp.134-147.

Grant J.W. and Weiner S.A. (1986), "Factors To Consider In Choosing A Graphical Animated Simulation System", *Industrial Engineer*, 31, August 1986, pp.37-40,65-68.

Grant F.H. (1989), "Tutorial Scheduling Manufacturing Systems with FACTOR", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.277-279.

Gray C.D. (1987), "*The Right Choice, A Complete Guide to Evaluating, Selecting, and Installing MRP II Software*", Oliver Wight Limited Publications, Essex Junction.

Greenwood N.R. (1988), "*Implementing Flexible Manufacturing Systems*", Macmillan Education, Hong Kong.

Haider S.W. and Banks J. (1986), "Simulation Software Products For Analyzing Manufacturing Systems", *Industrial Engineer*, 31, July 1986., pp.98-103.

Harrison M. (1990), "*Advanced Manufacturing Technology Management*", Pitman, Avon, Great Britain.

Heinonen R., Jokinen A., Ranta J., and Wahlstrom B. (1986), "On the Simulation of Flexible Manufactuirng systems", in the "*Modelling and Simulation in Engineering*", Wahlstrom B. and Leiviska K. (eds.), Elsevier Science Publishers, Amsterdam, pp.307-312.

Herrell C.R. and Tumay K. (1990), "ProModelPC Tutorial", in the *Proceedings of the 1990 Winter Simulation Conference*, Balci O., SadowskiR.P and Nance R.E. (eds.), Simulation Computer Society, San Diego, pp.128-131.

Hlupić V. and Paul R.J. (1992a), "Software Packages for Manufacturing Simulation:A Comparative Study", in the *Proceedings of the 14th International Conference "Information Technology Interfaces"*, (September 1992, Pula, Croatia), University of Zagreb Computing Centre - SRCE, Zagreb, pp.387-391.

Hlupić V. and Paul R. (1992b), "Simulation Modelling of Flexible Manufacturing Systems using Activity Cycle Diagrams", accepted for the *Journal of Operational Research Society*.

Hlupić V. and Paul R.J. (1992c), "FMS Scheduling Strategies Using the Simulation Package SIMFACTORY II.5", in the *Proceedings of the 8th International Conference on CAD/CAM, Robotics and Factories of the Future*, (August 1992, Metz, France), ISPE, pp.1672-1686.

Hlupić V. and Paul R.J. (1993a), "Selecting Software for Manufacturing Simulation", in the *Proceedings of the 15th International Conference "Information Technology Interfaces"*, (June 1993, Pula, Croatia), University of Zagreb Computing Centre - SRCE, Zagreb.

Hlupić V. and Paul R.J. (1993b), "Evaluating the Simulation Package "WITNESS" on an Automated Manufacturing System", submitted to the *Journal of Computing and Information Technology*, April 1993.

Hlupić V. and Paul R.J. (1993c), "A Critical Evaluation of Modelling an Automated Manufacturing System using the Simulation Package WITNESS", accepted for the *9th International Conference on CAD/CAM, Robotics and Factories of the Future*, New Yersey, USA, August 1993.

Hlupić V. and Paul R.J. (1993d), "Simulating an Automated Paint Shop in the Electronics Industry", accepted for the *Simulation Practice and Theory*, April 1993.

Hlupić V. and Paul R.J. (1993e), "Simulation Modelling of the Automated System for Electrostatic Powder Coating of Metal Components", accepted for the *WSC'93-Winter Simulation Conference*, Los Angeles, USA, December 1993.

Hlupić V. and Paul R.J. (1993f), "Simulation Software in Manufacturing Environments: A Users' Survey", submitted to the *Journal of Computing and Information Technology*, April 1993.

Hlupić V. and Paul R.J. (1993g), "Software for Manufacturing Simulation: A Users' Survey", accepted for *ESS'93 - European Simulation Symposium*, Delft, Netherlands, October, 1993.

Hlupić V. and Paul R.J. (1993h), "A Methodology for Selecting of a Manufacturing Simulator", accepted for *UKSS'93 - The EUROSIM CONFERENCE*, Lake District, UK, September, 1993.

Holder K. (1990), "Selecting simulation software", *OR Insight*, 3:4, October-December 1990, pp.19-24.

Holder R.D. and Gittins R.P. (1989), "The Effects of Warship and Replenishment Ship Attrition on War Arsenal Requirements", *Journal of the Operational Research Society*, 40:2, pp.167-175.

Hollocks B. (1989), "Simulation and Advanced Manufacturing Systems", in the *Proceedings of the 1989 European Simulation Multiconference*, (June 1989, Rome, Italy), Simulation Computer Society, San Diego, pp.21-23.

Hollocks B. (1992), "A well kept secret? _ Simulation in manufacturing industry reviewed", *OR Insight*, 5:4, Oct-Dec 1992, pp.12-17.

Hurrion R.D. (1986), "Visual Interactive Modelling", *European Journal of Operational Research*, 23, pp.281-287.

Hurrion R.D., (1991), "Intelligent Visual Interactive Modelling", *European Journal of Operational Research*, 54, pp.349-356.

Hutchison J. (1991), "Current and Future Issues Concerning FMS Scheduling", *OMEGA International Journal of Management Science*, 19:6, pp.529-537.

Kiran A.S. and Smith M.L. (1983), "Simulation studies in job shop scheduling; a survey", in "Simulation in Inventory and Production Control" - *Proceedings of the Conference on Simulation in Inventory and Production Control*, San Diego, Simulation Computer Society, San Diego, pp.46-51.

Kirkpatrick P. and Bell P.C. (1989), "Visual Interactive Modelling in Industry: Results from a Survey of Visual Interactive Model Builders", *Interfaces*, 19:5, September-October 1989, pp.71-79.

Kleine H. (1970), "A survey of users' views of discrete simulation languages", *Simulation*, May 1970, pp. 225-229.

Kleine H.(1971), "A second survey of users' views of discrete simulation languages", *Simulation*, August 1971, pp. 89-94.

Kochhar A.K. (1989), "Computer Simulation of Manufacturing Systems - 3 Decades of Progress", in the *Proceedings of the 3rd European Simulation Congress*, (October 1989, Edinburgh), Simulation Computer Society, San Diego, pp.3-9.

Kochhar A.K. and Ma X. (1989a), "Use of Computer Simulation: Aids for Solving Production Management Problems", in the *Proceedings of the 3rd European Simulation Congress*, (October 1989, Edinburgh), Simulation Computer Society, San Diego, pp.516-522.

Kochhar A.K. and Ma X. (1989b), "Discrete Event Simulation Software Tools for the Simulation of Advanced Manufacturing Systems", in the *Proceedings of the 1989 European Simulation Multiconference*, (June 1989, Rome, Italy), Simulation Computer Society, San Diego, pp.13-18.

Larsen N.E. and Alting L. (1989), "Simulation: A concept for production scheduling", *1989 IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference Proceedings*, Atlanta, IIE, pp.517-522.

Law A.M. and Haider S.W. (1989), "Selecting Simulation Software for Manufacturing Applications: Practical Guidelines & Software Survey", *Industrial Engineer*, 34, May 1989, pp.33-46.

Law A.M. and McComas G. (1989), "Pitfalls To Avoid In The Simulation Of Manufacturing Systems", *Industrial Engineering*, 31, May 1989, pp.28-31,69.

Law A.M. and Kelton W.D. (1991), "Simulation Modelling & Analysis", 2nd Edition, McGraw-Hill, Singapore.

Lenz J.E. (1983), "MAST: a simulation tool for designing computerised metalworking factories", *Simulation*, 2, pp.51-58.

Lenz J.E.(1989), "The MAST Simulation Environment", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.243-248.

Murgiano C.(1990), "A Tutorial on WITNESS", in the *Proceedings of the 1990 Winter Simulation Conference*, Balci O., SadowskiR.P and Nance R.E. (eds.), Simulation Computer Society, San Diego, pp.177-179.

O'Gorman P., Gibbons J., and Browne J. (1986), "Evaluation of Scheduling Systems for a Flexible Transfer Line Using a Simulation Model", in "*Flexible Manufacturing Systems: Methods and Studies*", A. Kusiak (ed.), North Holland, pp.209-222.

O'Grady P.J. and Menon U. (1986), "A Concise Review of Flexible Manufacturing Systems and FMS Literature", *Computers in Industry*, 7, pp.155-167.

O'Keefe R.M. and Haddock J. (1991), "Data-driven generic simulators for flexible manufacturing systems", *International Journal of Production Research*, 29:9, pp.1795-1810.

"*Operating instructions for the electrostatic GEMA-powder coating equipment TYPE 726-VSE*" (1985), RANSBURG-GEMA Ltd., St.Gall, Switzerland.

"*Operating and maintenance instructions for electrostatic powder coating plant supplied and installed at BICC VERO*" (1985), BICC Vero Electronics, Eastleigh, Uk, internal report.

Oxford Advanced Learner's Dictionary (1989), 4th edition, (Cowie A.P., ed.), Oxford University Press, Oxford.

Parnaby J. (1979), "Concept of a manufacturing system", *International Journal of Production Research*, 17:2, pp.123-135.

Paul R.J. (1991), "Recent Developments in Simulation Modelling", *Journal of the Operational Research Society*, 42:3, pp.217-226.

Pidd M. (1989), "Choosing discrete simulation software", *OR Insight*, 2:3, July-September 1989, pp.22-23.

Pidd M. (1992a), "*Computer Simulation in Management Science*", Third Edition, Wiley, Chichester.

Pidd M. (1992b), "SKIM: a simulation sketchpad for plant design", *Journal of the Operational Research Society*, 43:12, pp.1121-1134.

Pidd M. (1992c), "Guidelines for the design of data driven generic simulators for specific domains", *Simulation*, 59:4, pp.237-243.

Pollacia L.F.(1989), "A survey of discrete event simulation and state-of-the-art discrete event languages", *Simulation Digest*, 20:3, pp.8-25.

Poorte J.P. and Davis D.A. (1989), "Computer Animation with CINEMA", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.147-154.

ProModel PC User Manual (1991), Version 5.0, PROMODEL Corporation, Orem.

"*PYRENE metal treatment services to industry - Technical Service Data Sheet*" (1990), Pyrene Chemical Services Ltd, Iver, UK.

QUATRO PRO - User's Guide (1991), Borland International, Scotts Valley, USA, Version 3.0.

Ramasesh R. (1990), "Dynamic Job Shop Scheduling: A Survey of Simulation Research", *OMEGA International Journal of Management Science*, 18:1, pp.43-57.

Rohrbough M.C. (1989), "Introduction to SIMFACTORY II.5", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.201-203.

Shannon R.E. (1975), "*Systems simulation - The Art and Science*", Prentice-Hall.

Shannon R.E., Mayer R., and Adelsberger H.H. (1985), "Expert systems and Simulation", *Simulation*, 44, pp.275-284.

Shannon R.E. (1986), "Intelligent Simulation Environments", in the *Proceedings of the Conference on Intelligence Simulation Environments*, Luker and Adelsberger (eds.), (23-25 January, San Diego, California), Simulation Computer Society, San Diego, pp.150-156.

SIMFACTORY II.5 Reference Manual and User's Guide (1990), San Diego, California, CACI Products Company.

Simulation in U.K. Manufacturing Industry, Study Report, Simulation Study Group, 1991.

Singhal K., Fine C.H., Meredith J.R., and Suri R. (1987), "Research and Models for Automated Manufacturing", *Interfaces*, 17:6, pp.5-14.

Smith R.L. and Platt L. (1987), "Benefits of animation in the simulation of a machining and assembly line", *Simulation*, 48:1, pp.28-30.

STATGRAPHICS - Statistical Graphics System by Statistical Graphics Corporation (1986), STSC Inc., Version 4.0.

Sturrock D.T. and Pedgen C.D. (1989), "Introduction to SIMAN", in the *Proceedings of the 1989 Winter Simulation Conference*, MacNair E.A., Musselman K.J. and Heidelberger P. (eds.), Simulation Computer Society, San Diego, pp.129-139.

Suri R., Derleth D., and Tomsicek M.(1990), "Modelling Manufacturing Systems using ManuPlan and SimStarter - A Tutorial", in the *Proceedings of the 1990 Winter Simulation Conference*, Balci O., SadowskiR.P and Nance R.E. (eds.), Simulation Computer Society, San Diego, pp.168-176.

Swain J. (1991), "Simulation Software Survey: World of Choices", *OR/MS Today*, 10, pp.81-103.

sysPACK VS7 User Guide (1990), sysPACK Ltd., London.

Szymankiewicz J, McDonald J, and Turner K. (1988), *"Solving Business Problems by Simulation"*, McGraw Hill, London.

"Thermoset Powder Coatings" (1989), ed. by John Ward, FMJ International, Publications, Redhill, UK.

Thome H.G. (1988), "Planning and monitoring of FMS by simulation", in the *Proceedings of the 7th International Conference on FMS, and 20th Annual IPA Conference*, (Sttutgart, Germany), Warnecke J., and Steinhilper R. (eds.), IFS Conferences, New York, pp.137-149.

Tocher K.D.T. (1961), "Some Techniques of Model Building", *Proceedings of the IBM Scientific Symposium on Simulation Models and Gaming*, IBM Corp., New York, NY.

Tocher K.D. (1965), "Review of Simulation Languages", *Operational Research Quarterly*, 16:2, pp.189-217.

Van Breedam A., Raes J., and Van de Velde K. (1990), "Segmenting the simulation software market", *OR Insight*, 3:2, April-June 1990, pp.9-13.

Valcada A. and Masteretta M. (1984), "Comparison of different computerized design tool for flexible manufacturing", in the *Proceedings of the 3rd International Conference on FMS and 17th Annual IPA Conference*, Warnecke H.J. (ed.), North-Holland, Boeblingen, pp.387-395.

WITNESS User Manual (1991), Version 3.07, AT&T ISTEL LTD., Redditch.

"Zinc phosphate pre-treatment process" (1991), BICC Vero Electronics, Eastleigh, UK, internal report.