MODELLING MANUFACTURING PLANNING AND CONTROL SYSTEMS: THE APPLICATION OF OBJECT-ORIENTED PRINCIPLES AND DISCRETE-EVENT SIMULATION

NICHOLAS JOHN BOUGHTON

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

September 1995

The University of Aston in Birmingham

Modelling manufacturing planning and control systems: the application of object-oriented principles and discrete-event simulation

Nicholas John Boughton **Doctor of Philosophy 1995**

Summary

Manufacturing planning and control systems are fundamental to the successful operations of a manufacturing organisation. In order to improve their business performance, significant investment is made by companies into planning and control systems; however, not all companies realise the benefits sought. Many companies continue to suffer from high levels of inventory, shortages, obsolete parts, poor resource utilisation and poor delivery performance. This thesis argues that the fit between the planning and control system and the manufacturing organisation is a crucial element of success. The design of appropriate control systems is, therefore, important.

The different approaches to the design of manufacturing planning and control systems are investigated. It is concluded that there is no provision within these design methodologies to properly assess the impact of a proposed design on the manufacturing facility. Consequently, an understanding of how a new (or modified) planning and control system will perform in the context of the complete manufacturing system is unlikely to be gained until after the system has been implemented and is running.

There are many modelling techniques available, however discrete-event simulation is unique in its ability to model the complex dynamics inherent in manufacturing systems, of which the planning and control system is an integral component. The existing application of simulation to manufacturing control system issues is limited: although operational issues are addressed, application to the more fundamental design of control systems is rarely, if at all, considered. The lack of a suitable simulation-based modelling tool does not help matters. The requirements of a simulation tool capable of modelling a host of different planning and control systems is presented. It is argued that only through the application of object-oriented principles can these extensive requirements be achieved.

This thesis reports on the development of an extensible class library called WBS/Control, which is based on object-oriented principles and discrete-event simulation. The functionality, both current and future, offered by WBS/Control means that different planning and control systems can be modelled: not only the more standard implementations but also hybrid systems and new designs. The flexibility implicit in the development of WBS/Control supports its application to design and operational issues. WBS/Control wholly integrates with an existing manufacturing simulator to provide a more complete modelling environment.

Key words: manufacturing planning and control systems, object-oriented, discrete-event simulation, modelling, control system design.

Acknowledgements

I would like to take this opportunity to thank the many people who have helped me during my research.

Firstly to my wife, Suzanne, without whom none of this would have been possible: for her constant support and encouragement during the research and the subsequent preparation of this thesis.

There are many members of the research group at Aston University, both past and present, to whom I am grateful. In particular, thanks to Andrew Jackson and Peter Ball for patiently reading earlier drafts of this thesis. I would also like to thank Peter for reacting quickly to those 'odd' software problems which occurred.

I am grateful to Doug Love, my supervisor: for offering me the chance to undertake this research project and for his support during the research. I have learned a lot.

I would like to acknowledge my sponsors, Lucas Engineering & Systems Ltd., and in particular Alan Hyman, for their support and suggestions.

Finally I would like to thank The Department of Mechanical Engineering, at Aston University, for supporting my research.

Table of Contents

Appendices

List of Figures

List of Tables

1. Introduction

This chapter will introduce the wider context in which this research is based and will outline the format of the subsequent chapters.

1.1 Background

The research comprising this thesis forms part of a wider research programme undertaken by members of the Integrated Design and Manufacture research group, in the Department of Mechanical & Electrical Engineering, Aston University.

Love et al. (1992) introduce the concept of simulating the whole manufacturing organisation through the combination of real software systems and specialist simulation elements: the term 'whole business simulation' (WBS) has been coined. It is suggested that in order to properly evaluate the effects of a design or operational decision on the manufacturing business, one should model (i.e. simulate) the whole business. In this way it would be possible to ensure that decisions are made in the context of the whole organisation, in contrast to decisions being made to improve one aspect of the business but at the expense of the whole.

The following provides a simple example of problems of this type: consider the decision by the finance department to reduce stock holding costs, implemented by purchasing without due consideration of manufacturing, only to find a reduction in customer service levels. The decision to reduce inventory is not uncommon but when made in 'relative' isolation it can have adverse effects from the original intentions. WBS has been developed to support the evaluation of decisions, from potentially anywhere in the organisation, in terms of the performance of the whole company.

One of the core elements of WBS is a manufacturing planning and control system: others include manufacturing (shop floor) operations, production engineering, finance and accounts. There are two ways in which the planning and control functionality can be provided: either through the inclusion of a real control system or through the provision of specialist simulation elements. The inclusion of real (i.e. commercial) systems may not always prove to be a practical, or desirable, alternative. The problem thus becomes one of providing the specialist simulation elements to model manufacturing planning and control systems. This is the subject of this thesis.

1.2 Modelling manufacturing planning and control systems

The provision of specialist simulation elements to model manufacturing planning and control systems raises two interesting questions: what are the approaches currently adopted to model these systems and, secondly, in what context is the modelling currently performed?

The different ways in which simulation has been applied to model and investigate planning and control systems are reviewed within this thesis. The general conclusion which can be made is that the current approaches do not lend themselves to the investigation or evaluation of 'design' decisions. The issues of design are important. One of the potential applications of WBS is to evaluate 'design' decisions: this may be with regard to a new product, a new manufacturing facility or a new, or modified, planning and control system.

It is this latter application domain which is pertinent to the work reported here. This research has identified a distinct lack in proper attention to the evaluation of the impact of a planning and control system design on the performance of the manufacturing facility. Consequently, an understanding of how a new (or modified) planning and control system will perform in the context of the complete manufacturing system is unlikely to be gained until after the system has been implemented and is running. This is not a satisfactory situation; Rolstadås (1998a) summarises the problem well:

> *"Even if the* [production management] *system is fully defined through the user and system specifications, it is difficult for end-users to fully understand the properties of the system until after they have tried it in real-life. Experience shows that after implementation many user viewpoints are created, calling for a changed design."*

The lack of attention to such design issues is reflected in the literature. In general, the context in which simulation modelling is applied to planning and control systems is one of investigation of operational issues: for example the evaluation of different lot sizing policies or scheduling rules. The application of simulation to the fundamental design of control systems is rarely, if at all, considered. One of the reasons for this is likely to be the lack of a suitable simulation-based modelling tool which is able to represent different planning and control system designs.

It is the purpose of this thesis to introduce a novel approach to modelling manufacturing control systems, from both design and operational perspectives: one which exploits object-oriented principles and discrete-event simulation.

1.3 The format of the thesis

The following introduces the content of the subsequent chapters.

Chapter 2 examines the fundamental role of a planning and control system within any manufacturing organisation. There has been, and continues to be, significant investment made by organisations in computer-based support of the planning and control functions. Unfortunately, many organisations have failed to realise the promised benefits of their computer-aided production management (CAPM) system. Although the emphasis has previously been on the process of implementation, this chapter concludes by arguing that the fit between the planning and control system and the manufacturing environment is a crucial element of success.

The relationships between the planning and control system, the manufacturing facilities and the products sold by an organisation are explored in chapter 3. It is argued that the control system is an integral component of the manufacturing system and should be designed in this context. Close investigation of the requirements of the different manufactured products of an organisation suggest that a single planning and control solution is unlikely to be adequate. The most appropriate solution is often a mixture of different approaches.

Chapter 4 continues the theme that the planning and control system should be designed in the context of the complete manufacturing system. The different and varied approaches to the task of designing control systems are reviewed. Although the common aim of most of the design methodologies is to design, or select, the most appropriate planning and control system for an organisation, there is no way of assessing whether or not this has been achieved. In other words, there is no way of predicting the performance, or behaviour, of the control system in the context of the manufacturing system. The chapter not only identifies the need to predict the performance of a planning and control system but identifies the different opportunities which exist.

The applicability of different modelling techniques are examined in chapter 5. In order to assess the impact of a proposed planning and control system design on the performance of the manufacturing system, discrete-event simulation is considered to be the most suitable.

The application of simulation to modelling planning and control systems cannot be claimed to be new. However, chapter 6 establishes that the current application of simulation to planning and control issues is limited. The particular focus of attention is towards operational issues: for example the evaluation of different lot sizing or scheduling policies. The application of simulation to the more fundamental design of manufacturing control systems is rarely considered. It is suggested that the current state of simulation tools does not encourage matters; the chapter concludes by defining the specification of a simulation-based tool which will provide the necessary functionality to address control system design, as well as operational, issues.

Of the different approaches to build simulation models, it is argued in chapter 7 that the route utilising manufacturing simulators is the most appropriate. However, an investigation of the functionality offered by current simulators does not adequately extend to the domain of planning and control system modelling: the emphasis being on shop floor functionality. The limitations identified exclude current manufacturing simulators from control system design applications. It is suggested that to remedy this situation, the necessary planning and control functionality be added to an existing, but appropriate, manufacturing simulator.

Chapter 7 raises two important questions: how will the planning and control functionality be provided and which of the current manufacturing simulators can support the integration of this functionality once developed? Chapter 8 addresses these questions. Essentially, the development of the required functionality means the development of computer software. The development of computer software, and in particular simulation software, is not a simple exercise: typically systems are both large and complicated. The principles of object-orientation are, however, becoming recognised as an approach which supports the development of such systems. Furthermore, object-oriented methods are finding successful application in the manufacturing domain. Object-oriented methods will be adopted in order to provide the planning and control functionality. It is important to integrate the developed functionality with an existing, but appropriate, manufacturing simulator: the Advanced Factory Simulator (AFS) (Ball, 1994) has been chosen for this purpose.

Chapter 9 discusses the design of a planning and control class library called WBS/Control. Following an overview illustration of the type of functionality which will be provided by WBS/Control, the design route to this functionality will be explained. The types of classes which will populate the library will be explained as well as how these classes meet the important requirements placed upon them. Chapter 10 details the implementation of WBS/Control: explaining how the designs have been transformed into a workable system. The relationships between WBS/Control and AFS are described in the appropriate context of the discussion.

Chapter 11 describes the application of WBS/Control from two perspectives: building simulation models and extending the current functionality. This chapter will demonstrate that it is possible to model planning and control systems through the application of an object library. A series of models will be described to demonstrate the scope and potential of WBS/Control. Furthermore, the functionality is not static: new or modified classes can be added to the library. This important property will be illustrated.

Finally, chapter 12 will summarise the conclusions which have been reached from this research project. In the light of these conclusions, several opportunities for further work will be outlined.

2. The significance of a manufacturing planning and control system to the manufacturing organisation

This chapter will discuss the fundamental role of a planning and control system within any manufacturing organisation. It will be apparent that it is a role which not only affects the manufacturing operations of an organisation but the operations, and performance, of the whole organisation. Typically, computer-based systems have been used to support the planning and control functions; significant investment has been, and continues to be, made in computer-aided production management (CAPM) systems. Unfortunately few organisations have realised the promised benefits from the implementation of such systems. It is argued here that the correct match, or fit, between the planning and control system and the manufacturing organisation is crucial

2.1 The planning and control system in the manufacturing organisation

2.1.1 An essential role

The following authors and practitioners offer useful perspectives on the role, or purpose, of a manufacturing planning and control system. Burbidge (1971) provides a befitting starting point:

> *"Production control is the function of management which plans, directs and controls the material supply and processing activities of an enterprise, so that specified products are produced by specified methods to meet an approved sales programme; these activities being carried out in such a manner that labour, plant, and capital available are used to the best advantage."*

Corke (1977) suggests there are two main objectives of production control:

> *"... to enable good delivery dates to be offered, and to get customers' orders completed on time ..."*

He continues with the following rider to these objectives:

> *"... consistent with keeping stocks and work-in-progress at an acceptably low level and the utilisation of plant, manpower and materials at an acceptably high level."*

Vollmann et al. (1992) summarise a planning and control system as one which

> *"... provides information to efficiently manage the flow of materials, effectively utilize people and equipment, coordinate internal activities with those of suppliers, and communicate with customers ..."*

Examples of typical internal activities which would be supported by a control system include:

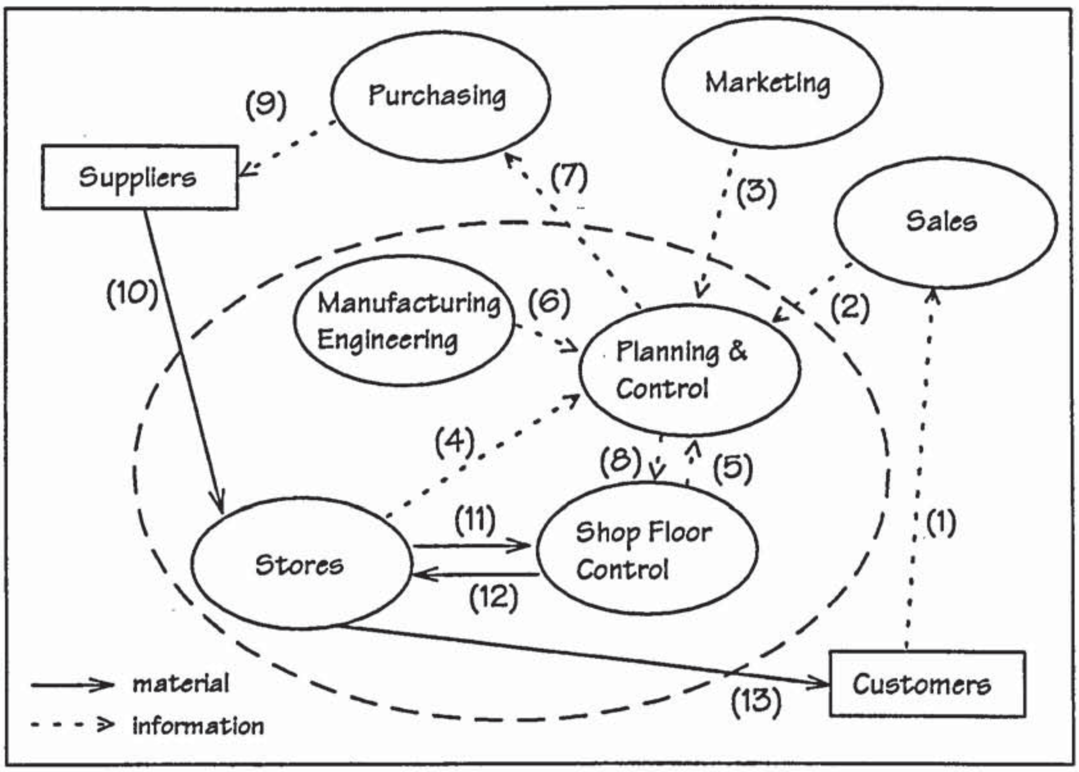
- determining suitable delivery dates for customer orders;
- planning capacity requirements in order to meet market demand;
- efficiently scheduling internal production activities;
- maintaining appropriate inventory levels;
- tracking resource utilisation on the shop floor;
- providing feedback when things do not go according to plan;
- providing information to other functions of the business.

More recently, Kochhar et al. (1994) have stated:

> *"Manufacturing control systems are implemented to ensure smooth and timely execution of all the activities which are undertaken from the time of receipt of an order to the eventual delivery of the required products."*

The above perspectives are in many ways complementary, emphasising a role which seeks to control efficiently the operations of the manufacturing system. However, the scope of the control system is not confined to the manufacturing operations of an organisation but involved with wider functions of the business: for example, marketing, sales, personnel, finance and purchasing. Furthermore, it extends beyond the boundaries of an organisation to include both customers and suppliers; essentially planning and control systems reconcile the demands placed upon the organisation by their customers by (efficiently) co-ordinating the internal resources as well as placing demands on suppliers.

Figure 2.1 illustrates some of the information and material flows, relating to planning and control, which are typical within a manufacturing company. The many interactions and information exchanges can be traced from the initial receipt of an order from a customer, through determining material and capacity requirements, to co-ordinating with suppliers and organising production. These interactions, which occur during the day-to-day, hour-by-hour operations, make for a very complicated system.



No.	Comment
1	Orders for goods, enquiries, customer specifications.
2	What to make - actual orders.
3	What to make - forecast of demand.
4	Current inventory status - on hand, due in, allocated stocks.
5	Current state of shop floor - levels of WIP, resource utilisation.
6	How to make the product - BOM, routing, standard times.
7	What to buy in - raw materials, components, consumables.
8	What to make - works orders.
9	Orders to suppliers - order quantity and delivery date.
10	Delivery of bought-in parts (raw materials and components).
11	Issuing of inventory to the shop floor.
12	Raw materials, components, finished goods to stores.
13	Finished goods, spares to customers.

Figure 2.1: Typical material and information flows in a manufacturing organisation.

This complexity is further compounded by the dynamics and uncertainties typical of the manufacturing environment, for example the dynamics of both the market place and supplier performance (Scott, 1994). These problems manifest themselves in terms of, for example, changes in customer expectations and the variations in product demand. Alternatively, with respect to supplier performance, there may be variations in delivery lead time and quality.

2.1.2 A note on the term 'planning and control'

Although the term 'production (manufacturing) control' is used, the definitions themselves offer a much wider perspective and include elements of both planning and control. In fact, in many texts the terms 'control' and 'planning and control' are used interchangeably; this will be continued in this thesis.

It will be useful at this point, however, to distinguish between the terms 'planning' and 'control'. A manufacturing planning and control system will, as the name suggests, involve both elements. The 'planning' processes will develop the production, capacity and material plans to which the company should adhere. These plans may be based on actual demand (i.e. customer orders) or forecast demand. These plans would be successful in themselves if nothing ever went wrong.

Unfortunately, as is often the case in manufacturing, events do not always happen as expected: for example, customer orders may change, actual demand may vary against forecast, capacity (machine or people) may become unavailable or suppliers may not deliver on time. Some form of 'control' is necessary when such deviations from plan occur. A good control system provides the necessary feedback information on which to base decisions and subsequent actions so that the operations can be brought back into line with the plan, or the plan revised appropriately. It is important for the controlling mechanisms to minimise any disruption to the manufacturing operations and attempt to maintain the planned deliverables.

2.1.3 A hierarchical view of manufacturing planning and control

Manufacturing planning and control is conveniently viewed as a hierarchy (Browne et al., 1988; Vollmann et al., 1992; Slack et al., 1995). Figure 2.2 illustrates such a hierarchical view over the long, medium and short-term horizons. This identifies three distinguishing features: the time horizon employed at each level, the amount of detail included in the planning information, and the relative balance of planning and control employed at each of the levels.

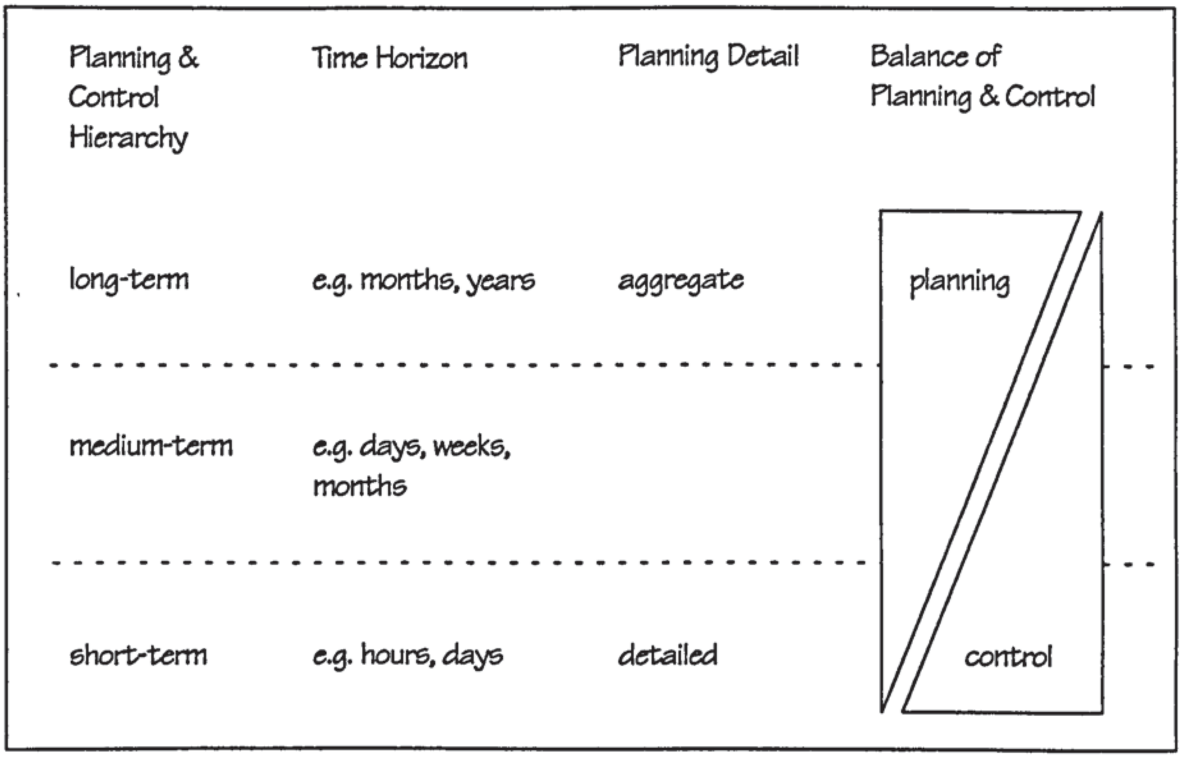


Figure 2.2: A hierarchical view of planning and control activities.

Over the long-term the time horizon is, by definition, greatest. As an organisation determines its long-term future operations, the emphasis is on planning and in particular aggregate planning. In many cases, this level of planning is based on forecast demand and set against financial targets and product families rather than a fixed number of product units. Over the medium-term the planning horizon is much closer and, thereby, decisions are made over a shorter time period. Accordingly the planning process involves a greater level of detail. The expected and/or confirmed demand, for example, will be more evident and based on actual quantities of distinct product items. The emphasis remains on planning although it is likely that the original forecasts will be subject to modification.

At the lowest level of the hierarchy, the day-to-day operations are managed and, accordingly, the information and production requirements are based on a completely disaggregated basis. The time scales are shortest and, typically, plans are subject to a great deal of disturbance: customers may request an earlier delivery of their order or there may be internal production problems such as machine breakdown. Consequently, the emphasis is necessarily on controlling such disturbances in order to maintain performance and reduce any knock-on effects.

Although the objectives of each level in the hierarchy described will not be the same, the decisions are neither disjointed nor unrelated. Decisions at each level are constrained and directed by those at the higher level. Moreover, if appropriate control is to be achieved, decisions at the lower level will obviously need to support those at the higher level.

To summarise, the role of the planning and control function is fundamental not only to the manufacturing operations but to the business as a whole. It is evident that the activities involved in the planning and control system process form a chain of inter-related events, operating within different time horizons and at different levels of detail. If effective operations are to result they all must occur and function as an integrated system.

2.2 Planning and control systems are a significant investment

Manufacturing companies have been using computer-based support for production planning and control for some time (Kochhar, 1984; King, 1972). This is only to be expected considering the scope and complexity of the tasks involved. There should be no doubt, however, that the implementation and operation of a manufacturing planning and control system will incur substantial costs to a company.

A recent survey, 'Computers in Manufacturing' (Benchmark Research, 1994a), identified manufacturing investment in information technology (IT) to be almost £1.5 billion. This figure, covering the period August 1993 to July 1994, represents spending on hardware, software and services for manufacturing related applications and includes both the engineering sector and process industries. The engineering sector alone accounts for 52% of the total spend, i.e. £0.78 billion.

A more detailed breakdown of the total spend reveals materials requirements planning (MRP) and production support accounting for £0.83 billion (55%) and shop-floor data collection and process control accounting for £0.28 billion (19%). The remaining £0.39 billion (26%) is attributed to computer-aided design and manufacture (CAD/CAM) and research and development. The survey suggests that, when compared to previous years, there does appear to be an increase in investment albeit a cautious one. In particular, organisations are becoming more and more determined to see a return on their investment.

The basic capital costs of a CAPM system include hardware, software, installation, data preparation and the system conversion process, however the costs of CAPM systems do not stop at the point of purchase. The implementation process itself incurs costs in terms of time and people as well as money. These costs are perhaps less quantifiable: for example, existing staff may receive training in the operation of the new system, additional staff may need to be recruited and the new system may need to be tested or perhaps run in parallel with the old system. Luscombe (1993) suggests by way of a guide that items such as education, training and outside assistance add a further 50% to the costs.

There are also costs involved with the day to day operation of a planning and control system which should not be underestimated. It has been noted that the production and material planning systems and related activities make a significant contribution to the indirect or overhead costs of a manufacturing business (Miller & Vollmann, 1985), in particular, those related to the procurement and co-ordination of raw materials, components and finished products required for production. Also included would be the salaries of the relevant employees, for example production planning, purchasing and materials management. These overhead costs account for approximately one third of the total product related costs (Murphy & Braund, 1990).

The direct costs of a business are also affected by the planning and control system. The derived production plans and schedules determine what to make and when and, thereby, co-ordinate operators and equipment as well as material. If production problems occur then short-term measures may be necessary, for example sanctioning over-time working, co-ordinating sub-contract work or sourcing another material supplier at short notice. Many aspects of the manufacturing operations are driven by the planning and control system and thereby have a significant impact on the financial performance of the business. Although a planning and control system does not add value, a good fit between the control system and the host manufacturing environment can facilitate those activities which do (Newman & Sridharan, 1995). Conversely, poor planning and control will have obvious adverse consequences.

In summary, substantial costs are likely to be incurred throughout the life of an organisation's planning and control system. It would be sensible, therefore, to make every attempt to ensure that the system matches the needs of the organisation.

2.3 The benefits of a control system are not guaranteed

Many manufacturing companies install planning and control systems in order to improve their business performance. This is often translated into performance targets such as improved schedule adherence, greater resource utilisation, reduced inventory (raw material, work-in-progress and finished goods), reduced lead-times and improved throughput (Wharton & Reid, 1990; Turner & Saunders, 1994).

Organisations are often tempted by CAPM system vendors who promise improvements such as those listed, providing impressive case examples of successful implementations as some form of 'proof'. There is no doubt that some companies have realised important benefits from their investment in a control system, for example reduced inventories, improved on-time delivery, reduced manufacturing lead-times and a reduction in shortages (Rose, 1991; Schlussel, 1990). Of course all such benefits are not seen by all successful companies; different organisations achieve varying degrees of success in different areas.

Unfortunately, not all of the available evidence is as optimistic as system vendors would like to suggest. Various studies indicate that the level of success achieved through the implementation of CAPM systems is not very high (Waterlow & Monniot, 1986; Duchessi et al., 1989; Thomson & Graefe, 1989; Wharton & Reid, 1990; Lopez & Haughton, 1993; Turner & Saunders, 1994). It is not uncommon to find manufacturing businesses still suffering from long lead-times, high levels of inventory, shortages, obsolete parts, poor resource utilisation, poor delivery performance and poor customer service. Organisations with such symptoms are not realising the control aimed for (Hayward, 1992), despite significant investment.

Wharton & Reid (1990) studied US manufacturing organisations over a five year period and found mixed success: many companies had seen some improvements in their levels of manufacturing performance, however, few had achieved the desired levels set by their original goals. Thomson & Graefe (1989) had earlier suggested that only about 20% of implementations are likely to achieve the expected performance levels. Lopez & Haughton (1993) identified that of the electronics companies they investigated, few were realising the range of benefits expected. In a study of the state of the art in CAPM in the UK industry, Waterlow & Monniot (1986) identified a number of deficiencies in commercially available systems. For example, they revealed that because users did not sufficiently understand how best to use their system they were not gaining the full benefits. In a more recent survey of small to medium enterprises (SMEs), Turner & Saunders (1994) noted that despite substantial investment in planning and control systems, manufacturers were still experiencing particular difficulties with schedule adherence.

It is difficult for organisations, or system vendors, to conceive that a control system which may appear on the surface to be an appropriate solution could fail to deliver the desired benefits. Waterlow & Monniot (1986) suggested that one of the problems lay with CAPM itself: *"CAPM was deceptive."* Although the principles of CAPM are not perceived to be difficult, they rely on the assumption that a manufacturing company's operations are both structured and organised. This assumption is not always true.

Informal systems are often found where the implemented manufacturing planning and control system is failing to deliver the desired benefits. Informal systems are adopted in order to counter any deficiencies found in the official system or, alternatively, mechanisms are used to bypass what are perceived to be unnecessary system disciplines (Kochhar, 1987; Hicks & Braiden, 1994; Turner & Saunders, 1994). Examples include the maintenance of informal material stores, borrowing parts from assembly kits, not recording stock transactions and working to local production schedules rather than those of the official system. In the extreme, the official system may be discarded and a replacement sought (Hayward, 1992; Webster & Williams, 1991).

The problems and criticisms attributed to the official system are, more often than not, compounded by the existence of an informal system. Hicks & Braiden (1994), for example, found that although companies operated computerised planning and control systems, effort was also directed towards the operation of informal manual systems, the outcome being poor data accuracy in the official system. Such situations lead to a lack of confidence in the information provided by the official system, reinforcing the need for informal mechanisms. Conversely, successful systems have proved to be self-reinforcing (Waterlow & Monniot, 1986).

This section has illustrated the marked contrast between companies which do realise the benefits from their planning and control system and those which do not.

2.4 Implementation - necessary but not sufficient

One of the consequences of systems not delivering the expected benefits has been a focus of attention on the process of implementation. Because early 'implementations' of CAPM systems were more technology driven and could be more aptly described as installations, the perceived solution, therefore, lay with improving the implementation process. It will be argued here, however, that a correct implementation alone is necessary but not sufficient to ensure the success of a planning and control system.

Material requirements planning (MRP) and manufacturing resources planning (MRPII) feature prominently in the literature concerning implementation and their outcomes. This is not surprising. Two recent surveys (Little & Johnson, 1990; Benchmark Research, 1994a) both indicate that MRP and MRPII systems are the main form of control employed, particularly in the larger companies. Furthermore, investment in such systems continues to be significant. The tendency, therefore, is to focus on why some organisations are more successful than others when implementing MRP and/or MRPII systems; for example, Duchessi et al. (1989) and Ghobadian (1992). Because the principles are not exclusive this discussion will take a more general position.

One of the more obvious misconceptions, apparent from earlier implementations, was that simply by buying a planning and control system all of the manufacturing problems of an organisation would disappear. It has been observed, however, that successful companies move towards technology only after more conventional improvements have been made (Roth & Miller, 1992; Ingersoll Engineers, 1985). Simply adding technology is unlikely to solve more fundamental problems. For example, Burbidge (1985) suggests that little improvement in production efficiency can be obtained without first simplifying the material movement within manufacture.

Many related issues have since emerged about the introduction of CAPM systems. For example, the implementation should not be viewed from a narrow technological perspective but should also include a much wider business and strategic perspective (Kirkwood et al., 1989). The nature of the problem, therefore, is far greater than one of installation, and should not be considered as an isolated exercise. In fact, several of the factors critical to successful implementation are essentially managerial and organisational (Lockett et al., 1991; Sillince & Sykes, 1993).

The term 'implementation' is often used more as an umbrella term than restricted to the process of an implementation. For example, Duchessi et al. (1989) identified three key success factors for successful implementation: top management commitment, the process of implementation itself and choice of software and hardware. They further breakdown top management commitment into managerial and organisational support, operational use and performance measurement; the implementation process into education and training, project planning, data accuracy and formal material transaction controls; the choice of software and hardware to include both system selection and vendor support.

Although Luscombe (1993) generally concurs with the viewpoint of Duchessi et al. (1989), he considers a complete understanding of the system concept (in this case MRPII) as well as a commitment to it to be the more fundamental elements of a successful implementation.

The importance of the fit between the system and the requirements of the organisation is rightly acknowledged (Newman & Sridharan, 1995). In particular, Ghobadian (1992) states:

> *"The realisation of the full potential of the system is dependent on the degree of fit between the technology and the environment in which it operates."*

Unfortunately, the implication here is that the system will be represented solely by the technology. This cannot be assumed to be the case. Firstly, there is more to the manufacturing planning and control system of an organisation than simply the CAPM system. The solution is more likely to be a balance between the CAPM system and the supporting infrastructure (Maull & Childe, 1993), in other words, the various policies and management procedures which are essential for the successful operation of the planning and control function(s). Secondly, the choice of planning and control system (discussed in Chapter 3) does not necessarily have to rely on a technology-based solution; kanban is an obvious example. In this context Newman & Sridharan (1995) provide the more pertinent observation:

> *"... managers are concerned with the fit between their infrastructure support system and the manufacturing environment."*

When the authors do acknowledge the importance of a good match between the system and the organisational requirements, their discussions tend towards the evaluation and selection of software systems with no indication of how an appropriate fit is to be attained. However, it would seem more appropriate that, before embarking on the actual process of implementation, organisations should already have confidence that the designed system will match their requirements; this is especially pertinent if the 'square peg in a round hole' syndrome is to be avoided. To properly implement the wrong system can prove to be a costly exercise in more ways than one, as Clark et al. (1991) indicate:

> *"... there is little point in vigorously implementing an inadequate system which is costly to maintain."*

The following example, from Berry & Hill (1992), further illustrates this point:

A company, manufacturing a wide range of products for both the original equipment and the spares market, invested over £5m in a comprehensive MRP system and shop-floor control system; standard computer software was adopted for both requirements. Following a lengthy implementation process, it was recognised that much of the new system was not required to plan and control the spares business. Over half of the original investment and a large part of the current operational costs were attributable to this part of the company's activity; they were not, however, necessary or appropriate for its effective control. Although a detailed implementation process was followed the system did not meet the requirements for all of the business; the system provided effective control for only one aspect of the business.

It is reasonable to conclude that although a pre-requisite, good implementation will not improve a poor choice, or more appropriately a poor design, of planning and control system. To reiterate, implementation is necessary but not sufficient.

2.5 Summary

The role of a planning and control system is fundamental to the successful operations of a manufacturing organisation. Although these systems prove to be a significant investment for any organisation, not all companies realise the operational benefits originally sought. Attention to the process of implementation is in itself not sufficient to achieve success: how well the control system matches, or fits, the host organisation is a crucial determining factor.

3. Placing the manufacturing planning and control system in context

This chapter will discuss the relationships between the manufacturing organisation, the products which it sells, the production facilities and their physical layout, and the production planning and control system. The control system is an integral component of the manufacturing system and, therefore, should be designed in this context. It is argued that single overall solutions are unlikely to adequately satisfy the planning and control requirements of a company; the best solution for a particular company is often a mixture of different approaches.

3.1 Planning and control systems in the context of the manufacturing system

3.1.1 Organisations are different

If all manufacturing organisations adopted the same manufacturing process or set of processes then matching the control system to the company would be straightforward. This, however, is not the case, as Dale & Russell (1983) argue:

> *"Each type of manufacturing with its inherent characteristics requires a different approach in the design of the production planning and control system, despite the fact that there are underlying similarities."*

The situation is further compounded since, as Ptak (1991) points out, the planning and control solution for one company is not necessarily the answer for all companies in the same type of industry.

Because the characteristics and attributes of manufacturing companies are different, even between those competing in what appears to be the same markets, it is important to understand the relationships which exist between the control system and the manufacturing system, i.e. how the organisation of the manufacturing processes influences the nature of the planning and control system. Waterlow & Monniot (1986) emphasise the importance of understanding the planning and control system in the context of the manufacturing system as a whole. In this way manufacturing systems engineers should be able to develop complete manufacturing systems which best suit the needs of the business.

3.1.2 The control system in the context of the manufacturing system

Parnaby et al. (1986) provide the following definition of a manufacturing system:

> *"An integrated combination of processes, machine systems, people, organisational structures, information flows, control systems and computers whose purpose is to achieve economic product manufacture and internationally competitive performance. The system has defined but progressively changing objectives to meet, some of which can be quantified and others, such as those relating to responsiveness, flexibility and quality of service, which whilst being extremely important are difficult to quantify. Nevertheless, the system must have integrated controls which systematically operate it to ensure that the competitiveness objectives are continually met and which adapt to change."*

There are two points to notice from this definition. The first is that the manufacturing system is itself composed of several sub-systems all operating as an integrated whole; the production processes and the planning and control systems are included as sub-systems.

The second point emphasises the key role of the control system: the dynamic processes of the manufacturing system need to be ordered and controlled if they are to efficiently meet the objectives placed on the company and, thereby, the system. Since one of the primary functions of a manufacturing system is to deliver the products demanded by the customers, the planning and control system(s) will need to provide effective support if the company is to be competitive. The role of the planning and control systems, including the different information and material flows, was discussed in chapter 2.

It is interesting to note a further point from this definition which is relevant to subsequent discussion in this thesis. There is an explicit recognition that the manufacturing system and, thereby, the supporting sub-systems do not operate in isolation but that the operating environment is subject to change from several sources. For example, customer needs and market requirements are continually changing. These are external influences on an organisation; internal effects typically include machine breakdown, material shortages and operator absenteeism. Such changes are representative of the dynamics of the manufacturing environment. The system and sub-systems not only need to be adaptable but also resilient to these changes.

3.2 The products, the manufacturing facility and the control system

The products which a company manufactures and the nature of the product demand are key considerations in relation to the design of the manufacturing system. Put simply, the product design and demand influence the combination of the manufacturing processes in the manufacturing system, this in turn influences the type and complexity of planning and control system which should be employed. The more components a product contains and the more stages of manufacturing involved, the greater the complexity in controlling the manufacture of the product. It follows that the production management tasks tend to be simpler in the case of a single fabricated part and more complex in the case of a customised product with many options.

It will be apparent that volume and variety are additional factors which are influential when considering the manufacturing facilities and control systems. It has already been mentioned that the choice of facilities is not straightforward, even within the same manufacturing sector. Furthermore, it will be evident that the organisation of the manufacturing operations has serious implications on the company's competitive position: for example, how well the company responds to market demands and the levels of investment required (Hill, 1993).

3.2.1 Job, batch or mass production?

When considering discrete parts manufacture, i.e. the production of individual items, there are traditionally three ways in which production can be viewed: job, batch or mass production. A brief explanation of each of these manufacturing processes will be included here, more detailed discussion can be found in, for example, Bennett et al. (1988), Hill (1993) or Wild (1989).

In the jobbing system, the variety of products manufactured will be wide but the volume low, tending towards one-off production; typical examples would be specialist tool manufacture and customised furniture. In a batch system, a number of similar parts are manufactured in batches, or lots, following a series of operations; usually the complete batch is processed before moving on to the next operation. Although batch manufacture covers a wide range of different production volumes it is likely to offer less product variety than jobbing production; examples include components for the automotive industry and 'off-the-peg' clothing. Mass production produces goods in high volumes but within a narrower product range. The demand levels are such that dedicated production facilities, or lines, can be introduced. Automobile assembly and consumer electronics are examples of mass production.

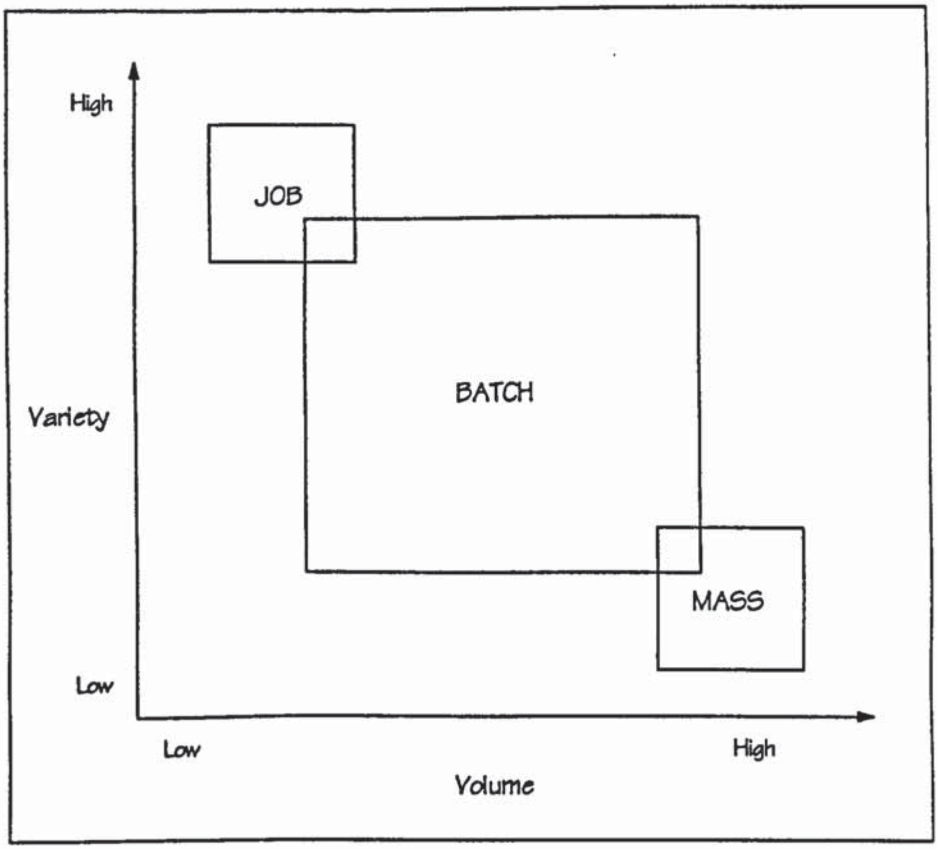


Figure 3.1: The volume-variety characteristics of job, batch and mass production.

Figure 3.1 (above) illustrates the product volume-variety characteristic in relation to job, batch and mass production. Batch production deliberately occupies a greater proportion of the volume-variety mix. Furthermore, when batch sizes are small the situation looks similar to that of jobbing; manufacturing with large batch quantities tends towards the more repetitive nature of mass production.

Table 3.1 summarises some of the important characteristics of job, batch and mass production and their relationships to the planning and control function.

It should not be assumed that all of discrete manufacturing can be neatly partitioned into job, batch or mass production. As with most classification schemes of this nature, the reality proves to be more complicated. The majority of manufacturing companies would place their production facilities at some point along what is essentially a continuum between job and mass production. In fact, to add further complication, it will be likely that within the same organisation different and/or hybrid manufacturing processes will be in operation (Little & Johnson, 1990; Benchmark Research, 1994a).

	Production Type:		
CHARACTERISTIC	JOB	BATCH	MASS
process technology	universal	general	dedicated
process times	long	→	short
process flexibility	high	→	low
product volumes	low	→	high
product variety	high	→	low
WIP levels	high	very high	low
finished goods levels	low	→	high
control of operations	complex	highly complex	easy
control of capacity	difficult	→	easy
product focus	low	→	high
process decoupling	high	→	low

Table 3.1: The characteristics of job, batch and mass production (after Hill, 1991 and Browne et al., 1988).

3.2.2 Facilities design

The arrangement of the manufacturing facilities themselves also influences the nature of the planning and control system. Browne et al. (1988) offer a general view based on two dimensions: the degree of process decoupling and the degree of product focus. The former considers the extent to which the manufacturing process is divided into distinct operations and separated (decoupled) by inventory buffers; product focus considers the extent to which the production facilities are dedicated to specific products.

The characteristics of these two dimensions are also included in Table 3.1. A jobbing environment would involve high process decoupling and low product focus whereas mass production with, for example, dedicated assembly lines would involve low process decoupling and high product focus. Traditional batch manufacture, often characterised by high levels of work-in-progress, would involve a high level of process decoupling and tend towards greater product focus than that found in jobbing production.

When discussing batch manufacture, a common classification of facilities layout is functional (or process) and cellular. Functional manufacture arranges similar machines together, for example all the drilling machines would be arranged at the same location. When a product or component is made, batches would visit each process location in the order dictated by the product routing. This approach to manufacture is often criticised for complex material flows and characterised by high levels of work-in-progress.

Cellular manufacture is an approach in which the manufacturing facility is divided into manufacturing units; Wemmerlöv & Hyer (1989) consider a manufacturing cell to be

> *"... a cluster of dissimilar machines or processes located in close proximity and dedicated to the manufacture of a family of parts."*

This approach offers a greater degree of product focus than that found with a functional layout. Cellular manufacturing, the origins of which lie with Group Technology (GT) (Burbidge, 1975), seeks to overcome the disadvantages associated with functional (or process based) layouts (Burbidge, 1975). The advantages of GT include simplified material flows, a reduction in inventory and work-in-progress, a reduction in throughput times and a reduction in material handling.

There continues to be a move towards the use of cellular manufacturing principles (Ingersoll Engineers, 1990; Fritz et al., 1993). This becomes particularly true when considering the implementation of just-in-time (JIT) based manufacture: it has been suggested that cells are one of the essential characteristics of JIT (Schonberger, 1986). However, the adoption of a functional layout is not as obsolete as Burbidge (1992) would like: Turner & Saunders (1994) observed that the functional layout appears to remain a common form of work centre organisation. It was also noted that organisations were operating more than one form of layout. This situation has also been observed by Wemmerlöv & Hyer (1989): in the light of this they suggest that companies might have to operate two control systems - one for the cell system and one for the remainder of the factory.

In summary, this section has reviewed the relationships between the products that a company manufactures and the type of production facilities layout appropriate for manufacture. These relationships influence the type of planning and control system which may be used and, therefore, require appropriate attention if the most suitable control system for the business is to be designed.

The following discussion further explores the relationships between products and control systems and suggests the adoption of composite or hybrid planning and control systems rather than a single, overall solution.

3.3 The problems with standard control solutions

Manufacturing has been offered several solutions to its planning and control problems over the past three decades: material requirements planning (MRP) (Orlicky, 1975) was offered as the replacement to the more traditional stock control systems; manufacturing resources planning (MRPII) (Wight, 1981) extended the functionality and control offered by MRP to provide a business-wide solution; the rise to prominence of Japanese manufacturing brought with it the just-in-time (JIT) philosophy (Schonberger, 1982) and the accompanying kanban control system (Mondon, 1994); the introduction of optimised production technology (OPT) (Fox, 1982) has focused on throughput and improving productivity. This has continued with computer integrated manufacturing (CIM) (Thomson & Graefe, 1989) and enterprise resource planning (ERP) (Waldren, 1992).

Since a manufacturing control system represents a significant investment decision for an organisation and one which is unlikely to be repeated with any great frequency, the choice of system should not be left to the adoption of the 'current trend'. However, the commercial drive behind these 'solutions' provides a persuasive argument. Hill (1993) suggests that many investment decisions have been based on perceived benefits rather than a good business fit and is, therefore, not surprised at the poor success rate of such system solutions.

One of the problems is that standard, or panacea, solutions fail to properly recognise the differences between manufacturing sectors and product markets and, thereby, organisational differences. It was pointed out earlier that organisations are different and, consequently, choice should be based on a much wider analysis. Unfortunately it is evident (for example, Turner & Saunders, 1994) that many organisations adopt a single system which adequately supports the needs of their major products but constrains the planning and control of the remaining products. The following section argues that this is not the right approach because different product types require the application of different planning and control solutions.

From a wider perspective, Hill (1993) argues that a manufacturing planning and control system is unlikely to be successful if it is not congruent with the business needs of the organisation, whereas successful businesses will have organised their manufacturing processes and infrastructures in order to support their manufacturing strategy. Hill's work on manufacturing strategy (Hill, 1993) promotes a framework to understand the important relationships and implications between manufacturing, marketing and financial decisions. He argues that if a company is to be competitive it must develop a manufacturing strategy which not only reflects but supports the characteristics of the chosen markets; decisions concerning manufacturing investment should then be made in line with this manufacturing strategy. This is a view shared with Parnaby (1986) when considering the design of competitive manufacturing systems; he suggests that a clear definition of the details of the business strategy should precede the design of the manufacturing system.

The choice of manufacturing facility, i.e. how the organisation will manufacture its products, will need to be decided in line with the manufacturing strategy. Correspondingly, investment in the infrastructure of an organisation, of which production planning and control system is an obvious example, must retain this focus if successful results are to be attained. This may not necessarily be achieved through the implementation of a single control system across the whole company.

3.4 Composite planning and control systems

Different products and different manufacturing processes can require the application of different planning and control systems, even within the same company (Zäpfel & Missbauer, 1993; Ptak, 1991; Dale & Russell, 1983). Giauque & Sawaya (1992) suggest that:

> *"Companies need to identify key characteristics of their own environment, then match them with the characteristics of each* [planning and control] *system to develop a mix that works best for them."*

They argue that there are advantages and disadvantages to all systems and, consequently, the best solution for a particular company is often a mixture of different approaches. They present an interesting case example in which a company has incorporated aspects of JIT, OPT and MRP to produce a control system which appropriately supports their needs. Aspects of the JIT philosophy were incorporated with the MRP concepts of dependent demand, co-ordinated ordering and lead-time offset; insights from OPT were important in overall capacity management.

Such mixed solutions are not unique. In their survey, Turner & Saunders (1994) identified companies which operate a combination of different control systems to drive manufacture: for example some products are controlled using re-order point (ROP) control, others kanban and others MRP. A similar situation is described by Larsen & Alting (1993). Particular hybrids systems, such as JIT-MRP (Parnaby et al., 1987) are also popular (Little & Johnson, 1990; Devereux et al., 1994).

In the light of such mixed solutions to the manufacturing control problem, the point made by Turner & Saunders (1994) is considered appropriate: planning and control systems do not need to be constrained by the functionality of a single proprietary system, but constructed to support the requirements of individual products. In this context composite systems will result.

The following sections will identify the different product types which may exist within the same manufacturing organisation and the relationship to the design of the planning and control system.

3.4.1 Runners, repeaters and strangers

Parnaby et al. (1987) do not consider that relying on the purchase of packaged (standard) solutions to manufacturing control problems is necessarily the most suitable:

> *"Different products require different control systems and any overall business control system must be a composite if it is to be truly effective."*

The basis for this is that the products of a company will fall into one of three product types: runner, repeater or stranger (Parnaby et al., 1987). This product classification is based on the frequency with which the products are demanded. Runners are products or components which have a regular demand and are produced frequently, for example weekly; repeaters are produced regularly but at longer time intervals than those associated with runners; strangers are products which are produced at irregular time intervals and with an element of unpredictability about their frequency. The point to notice is that each of the categories requires a different form of control system. Lewis (1994), for example, suggests that in situations where there is a high proportion of runners, kanban would be suitable for the runners and period batch control (Burbidge, 1985) used for the repeaters and strangers.

Most manufacturing companies will find a proportion of their products in each of the categories; furthermore it is likely that a single product will move through each of these categories as its life-cycle develops.

3.4.2 ABC classification

The choice of control system does not have to be based on the runner, repeater and stranger classification: the ABC inventory classification (Lewis, 1981; Bennett et al., 1988; Slack et al., 1995) has also been influential. It is common for companies to discriminate between the stocked items (raw material, components, finished goods) so that suitable control systems can be applied to each item. This is based on the observation that in any inventory system some products will be of more significance to the company than others. The most significant products would involve careful planning and control, whereas the least significant items would involve less rigorous control. For example, products of high value would demand careful control in order to avoid a build up of excessive inventories. Equally it is important to ensure availability of those components which are common to many of the company's saleable items if good customer service levels are to be maintained. Attention should also focus on those products which are problematic to source, for example those items whose supplier may be based overseas.

One common way to discriminate between products is to use Pareto analysis. Items are ranked highest to lowest according to some criterion, for example by value of usage; it is then observed that a small proportion of the inventory items account for the larger proportion of total inventory value. This is often referred to as the 80/20 rule, whereby 20% of the stocked items account for 80% of the total value. In ABC classification class A items would be those 20% of items, class B items would account for the next 15%-30% of items and class C items would account for the remaining products and account for little of the total value.

In terms of control systems, tighter planning control systems would operate for class A items whereas less rigour would be adopted to control class C items. Bennett et al. (1988), for example, suggest a periodic review policy for class A items, re-order level policies for class B items and a two-bin approach for class C items. Of course such suggestions will vary according to the characteristics and requirements of the company considered. What is evident, however, is that different planning and control systems will operate within the same organisation, in fact composite solutions are recommended.

3.4.3 Make-to-order and make-to-stock

It has been observed that much of the research and subsequent discussion relating to planning and control systems has focused more on the needs of make-to-stock (MTS) companies than those in the make-to-order (MTO) environment (Hendry & Kingsman, 1989). However, there are substantial differences between the planning problems associated with MTO and those of MTS manufacturing. Perhaps one of the basic distinctions between MTO and MTS is the timing of the initial order relative to the production of the finished product (Hicks & Braiden, 1994). Table 3.2 provides a more detailed comparison between MTO and MTS companies.

	Company Type:	
CHARACTERISTIC	make-to-stock (MTS)	make-to-order (MTO)
product demand	demand for standard products, greater predictability	demand volatile, little predictability
product mix	many standard products	few standard products
resources	greater specialism of labour and equipment	flexible work force, multi-task equipment
capacity planning	based on forecast demand, planned well in advance, adjust as necessary	based on receipt of customer order, difficult to plan in advance
product lead time	set internally, of less importance to customer	agreed with customer, vital for customer satisfaction
prices	fixed by producer	agreed with customer prior to manufacture

Table 3.2: A comparison of MTS and MTO companies (after Hendry & Kingsman, 1989).

Traditionally jobbing companies have been considered representative of MTO production; the nature of the product and the frequency of demand mean that production is not initiated until a particular order is received. Those organisations who supply products which are more standard and have a pattern of demand which offers greater predictability and regularity operate along the lines of MTS.

A clear distinction between MTO and MTS within the same organisation is not always apparent. For example, a MTO company may have a range of finished products which are configured from a number of common components or sub-assemblies. Rather than hold stocks of the finished products, for which the demand is likely to be unpredictable, it would be reasonable to hold stocks of the components. It is then possible to make the finished item from the components on receipt of a customer order. This assemble-to-order approach increases the speed at which the company can respond to customer demand as well as reducing the amount of finished goods.

Clearly, as with the case of runners, repeaters and strangers, there is an opportunity to support different classes of products with appropriate control systems within the same organisation (Riis, 1994). This would appear sensible: Little & Johnson (1990) noted that, of the companies they surveyed, the majority utilised both MTO and MTS.

3.5 Summary

The importance of the relationship between the products sold by a company, the manufacturing facility and the planning and control system has been emphasised. It is essential that adequate attention to such detail is made if a successful match between the manufacturing control system and the organisation is to achieved. The results of an analysis of this kind are unlikely to point to the adoption of a single standard solution; it is argued that appropriate planning and control systems should be designed to support the different classes of products and manufacturing processes within an organisation.

4. The need to assess the impact of a planning and control system design

This chapter will identify both the need and the opportunity to assess the impact of a control system design on the host manufacturing system. The relationships between the planning and control system and the manufacturing facility have already been established. It is crucial, therefore, to design control systems in the context of the complete manufacturing system if adverse results are to be avoided. The varied approaches to the design of manufacturing planning and control systems are discussed; it is concluded that no design method includes proper assessment of performance. Assessing the impact of a planning and control system design on the performance of the manufacturing system will not only provide important insight but may avoid implementing the wrong system.

4.1 Change and the manufacturing environment

Manufacturing organisations are becoming increasingly accustomed to change: a characteristic which in many ways has been caused by the highly competitive nature of the manufacturing environment. In particular, manufacturing companies are today facing intense competition in the world markets, especially from Japan and the Far East. This situation has forced manufacturing organisations to re-assess their manufacturing operations; there is a growing requirement to produce a wide diversity of products for an increasingly demanding market place.

Table 4.1 summarises the shift in manufacturing emphasis over the last three decades. The emphasis has moved towards increased product variety, reduced (but reliable) delivery times, shorter product life-cycles and a recognition of the importance of quality. These characteristics have profound implications on the way that manufacturing processes are organised and controlled. For example, the old notions of high levels of inventory and economies of scale are no longer seen as valid.

In many cases, companies are seeking to improve and simplify their production operations: the introduction of cellular manufacturing is one example. It is important that any modifications to the manufacturing system should include appropriate changes to the planning and control system. The remainder of this section will examine the problems which can arise if the design of the planning and control system is not considered along with that of the manufacturing system.

CHARACTERISTIC	EMPHASIS 1960s	 1990s
philosophy	production led	market led
techniques	simple	complex
product range	narrow	wide
tooling	dedicated	flexible
fixed costs	low	high
labour costs	high	low
product life cycle	long	short
competition	national	global
customers	stable	demanding
inventory	order point	just-in-time
pricing	cost plus	market driven

Table 4.1: The shift in manufacturing emphasis (after Scott, 1994).

The importance of planning and control systems to the success of cellular manufacture has been observed, for example in Sinha & Hollier (1984). Dale & Russell (1983) have also commented:

> *"It is absolutely vital that the requirements and characteristics of the manufacturing system are taken into account when designing the production control system and vice versa."*

These conclusions stemmed from studies which had found that some companies were still experiencing production control difficulties after they had implemented (GT) cells: for example in scheduling and loading of work through the cells. The benefits from the adoption of GT principles were not realised and the overall effectiveness of production was reduced. The difficulties were attributed to the poor design of the production planning and control system.

It was observed that in many cases, companies had re-organised their shop floor operations according to GT principles and then attempted to implement the existing, or a modified, control system. Passler et al. (1983) identified similar situations in which the match between the new manufacturing facility and the old planning and control system becomes no longer valid. The old production control system, more likely to have been designed to support a functional approach to manufacture, would not exploit the simplified material flow and reduced throughput times typical of GT cells.

There are perhaps two options with respect to the planning and control system when the shop floor is being reorganised: either the existing system becomes inappropriate (a new system therefore needs to be designed to replace it) or, with modifications, the system can be retained. The assumption behind the latter option is that the modifications are designed such that the system does support the redesigned manufacturing facility. For example, the move to cellular operations means that the number of planning and control reporting points is reduced. Furthermore, the detailed scheduling of work within cells by a centralised planning and control system is unnecessary: it can be left to the cell supervisor to organise the operations of the cell.

The decision to remain with an existing planning and control system after modifications have been made to the manufacturing facility, if made for reasons of expediency rather than design, is not a satisfactory option. Berry & Hill (1992) review two similar situations where this has been the case. In both cases each organisation decided to retain its existing control system although the manufacturing processes and operations had changed. The basis of the decisions were not unreasonable: one company wished to reduce the number of manufacturing changes which were taking place while the second hoped to minimise the overall investment. Unfortunately, though not surprisingly, the old control systems were unable to support the new manufacturing directions and the associated demands. The poor performance which resulted comprised increased lead-times, coincident shortages and excesses in stocks and reduced levels of customer service.

Rather than remaining completely with an existing planning and control system, one well tried approach is to make the 'necessary' modifications to the system in order to adapt it to the new way of doing things. This approach is particularly popular with computer-aided production management (CAPM) systems. However, continued modifications themselves lead to problems: systems become overly complex and prove difficult to accommodate future modifications (Benchmark Research, 1994b).

Changes in the manufacturing system, in particular the organisation of the production processes, are not uncommon in today's manufacturing climate. The manufacturing system design must be accompanied by an appropriate design of the planning and control system if positive results are to be achieved.

4.2 Designing manufacturing planning and control systems

4.2.1 Introduction

The significance of the planning and control system design problem can be judged by the volume of literature on the subject. This section includes a review of the some of the varied approaches to the problem of designing (and implementing) manufacturing planning and control systems. This review does not claim to be exhaustive, however, it provides sufficient coverage to demonstrate the range and scope of the current approaches. Table 4.2 summarises the characteristics of the different approaches to planning and control system design.

4.2.2 Macro and micro design approaches

Before discussing Table 4.2 it is appropriate to define what is meant by the 'design' of planning and control systems. Following a study of the literature, it is reasonable to suggest that there are two approaches to planning and control system design: the 'macro' approach and the 'micro'. On the one hand there are those authors who attempt to choose the 'best' planning and control option for a company from 'standard' approaches: MRP, MRPII, ROP, kanban and OPT. To support this task selection rules, company profiles and guidelines are devised; they are utilised to determine a control system which fits the characteristics of the organisation. The work of Grünwald et al. (1989) and Hill & Brook (1994) are two examples of the macro approach.

The micro approach takes a more fundamental view of the design problem. It is defined here to mean that the solution is determined by taking a more complete and low level perspective. The solution is not assumed to be one of, for example, MRP, ROP or kanban but may be a hybrid, or composite, solution or even a completely 'new' system altogether. The 'runners, repeaters and strangers' classification (Parnaby et al., 1987) provides an appropriate illustration (see also chapter 3). The products of an organisation are classified either as a runner, repeater or stranger and each of the categories requires its own control system: this system may well be one of the 'standard' solutions or a new bespoke system. The micro approach is not as widely addressed by the literature, however Bertrand et al. (1990) and Berry & Hill (1992) are two examples.

In essence, the difference between the two approaches is one of a 'top-down' view (macro approach) versus a 'bottom-up' view (micro approach). For example, a methodology which adopts the macro approach may conclude that the implementation of a kanban system would be suitable for an organisation. The micro approach would consider more fundamentally the actual design of the kanban system: Lewis & Love (1993), for example, offer a more detailed approach to the design of kanban-based systems. The authors make the point that there is more to the design of a kanban system than determining the number of kanbans.

4.2.3 The different approaches to the design of planning and control systems

Table 4.2 summarises the varied approaches which can be attributed to the design and implementation of manufacturing planning and control systems. Some of the salient points will be discussed here.

Many of the approaches suggest a design alternative from the more standard planning and control solutions, i.e. MRP, MRPII, ROP, kanban and OPT. This choice is based on the characteristics of the company, general selection guidelines or a series of selection rules. An attempt is made to match the characteristic and requirements of the company to those of the system and thereby determine a good 'fit'. Grünwald et al. (1989), for example, have developed a framework to compare various production control concepts and attempt to simplify answers to the question: which concept fits which production system? It is considered that this approach is really one of selecting control systems rather than designing them.

A focus on information system and CAPM system design is evident: a range of different solutions are offered. Schofield & Bonney (1981) have used manufacturing profiles to predict the need for computerisation of various planning and control sub-systems. The work of Muhlemann et al. (1990, 1995) has utilised software templates and fourth generation languages in order to design and implement production information systems for small to medium enterprises. Rolstadås (1988b) adopts what could be considered a micro approach to the design of flexible production planning systems. He seeks to define the fundamental data and operational requirements of a planning system: different systems could then be configured from these basic elements.

	Manufacturing Planning and Control Design Approach:													
CHARACTERISTIC	1	2	3	4	5	6	7	8	9	10	11	12	13	14
suggests type of planning and control system		•	•	•					•	•		•		
information system, CAPM, or software oriented	•				•	•	•	•			•		•	
detailed design methodology or framework available			•	•	•		•		•	•	•		•	
selection rules (S), guidelines (G), checklist (C), profiles (P)	• P			• S, C, G				• P		• C, G		• G		• G
supporting tools available			•	•	•		•			•			•	
included in a wider manufacturing system design methodology		•	•											
target manufacturing sector (T) or origins of work (O) specified	O, b	T, c	O, e		O, e		O, *	O, b	O, *	T, b				

Key:-

1	Schofield & Bonney (1981)
2	Lucas 5 step methodology (Lucas Engineering & Systems, 1991)
3	DRAMA II (Bennett & Forrester, 1993)
4	Kochhar et al. (1992), Kochhar & McGarrie (1991)
5	Maull et al. (1990), Maull & Childe (1993)
6	Rolstadås (1988b)
7	Muhlemann et al. (1990, 1995)
8	Barber & Hollier (1986a, 1986b)
9	Banerjee et al. (1993)
10	Hill & Brook (1994)
11	Bertrand et al. (1990), Bertrand & Wijngaard (1986)
12	Grünwald et al. (1989), Van der Linden & Grünwald (1980)
13	Doumeingts (1985, 1988)
14	Berry & Hill (1992)

e - electronics, b - batch, c - cellular manufacture, * - small to medium enterprises

Table 4.2: Different approaches to the design of planning and control systems.

Many of the approaches listed include a detailed methodology, or description, in order to support the design or selection of a control system. Additionally, many of the approaches include design tool(s) which support either all or part of the methodology. Maull et al. (1990), for example, have developed a work book. Perhaps one of the most developed of the design tools is the set of expert systems developed by Kochhar et al. (1991). They have developed three knowledge bases: one assists the selection of an appropriate planning and control system, a second performs a 'gap' analysis (to determine the difference between the current company capabilities and those required by the selected system) and a third provides a structured 'help' in order to fill the 'gaps' identified at the gap analysis stage.

Two of the approaches summarised in Table 4.2 form part of a wider manufacturing system design methodology, i.e. the Lucas 5 step methodology (Lucas Engineering & Systems, 1991) and DRAMA II (Bennett & Forrester, 1993). DRAMA II, like the Lucas methodology, includes a planning and control system design stage but offers more detailed support to the selection of an appropriate control system: this is achieved through 'design option guides' (DOGS) and a 'generalised design methodology'. Both these attributes are not confined to control system selection, they are included in all stages of the DRAMA II methodology.

To summarise, a common characteristic of all the approaches considered is that they all seek to determine the most suitable planning and control solution for a particular organisation; what is interesting is the varied approaches taken to achieve this. In general, however, the emphasis is towards the macro approach to the design problem; essentially the methodologies seek to 'select' systems and modify them according to the specific characteristics of the host organisation. This is in contrast to adopting a micro approach to designing planning and control systems.

Although the approaches attempt to design, or select, the most appropriate system, there is no way of assessing whether or not this has been achieved: none of the methodologies include proper assessment of the design. In other words, no attempt is made to check how well the system performs, or behaves, in the context of the manufacturing system. A comparison can be drawn between this situation and that of the Lucas manufacturing system design methodology (Lucas Engineering & Systems, 1991). Included in this methodology are stages which test the robustness of a design against variations in certain design parameters: for example, machine performance, product volume and variety. An assessment of performance in this context is not suggested by any of the planning and control systems design methodologies. This is perhaps not surprising: as subsequent chapters will demonstrate, the lack of a suitable modelling tool means that properly assessing the impact of a control system design is not feasible.

4.3 Assessing the impact of a design

Waterlow & Monniot (1986) noted that because of the complexity of CAPM systems, organisations found it difficult to understand exactly how the control system affected manufacturing performance. This point is also relevant when one considers the design of a new planning and control system.

The design methods and approaches previously discussed do, by definition, attempt to determine the most appropriate planning and control system for an organisation. However, there is no provision for assessing how the system will behave in an operational sense. It is not just a question of whether or not the designed system will deliver the desired results, but also one of understanding how the design will affect manufacturing. Sinha & Hollier (1984) make the point that it is important to understand how the system behaves under different and varying conditions. The fit between the control system and the manufacturing system is reflected in the performance of the manufacturing operations (Sridharan et al., 1989). Consequently, assessing the impact of a planning and control system on manufacturing performance, in advance of implementation, will provide important and timely insight in to the potential system behaviour.

By way of illustration, if a company decided to implement a MRP system it would be useful to investigate the effects of the choice of certain operational policies, for example lot sizes or safety stocks, on the manufacturing facility. Equally, the effects of different shop floor scheduling policies may also be investigated. It will be evident from chapter 6 that these examples prove to be the more conventional of modelling applications where assessment of performance is considered.

The assessment of a complete planning and control system design offers a significantly different challenge. If, for example, a more fundamental design of an organisation's planning and control system arrived at a combination of aspects from several different systems, then this would provide the basis of an interesting evaluation. The manufacturing facility of an organisation could, for example, be organised along cellular principles and with each cell utilising a different planning and control mechanism, combined with a centralised system interpreting the customer demand. The company described by Giauque & Sawaya (1992) is also a good example (see chapter 3). The impact of such designs would not be straightforward to predict.

The timing of the analysis is important. If there are problems with the design, assessment of performance once the planning and control system has been implemented will be too late. These problems may lead to a mismatch between the control system and the manufacturing system. To rectify the situation may require software and/or policy modifications. Assessing the performance in advance of any implementation, however, will provide an opportunity to smooth out any problems which may occur. The same is true of intended changes to operational or policy issues. If there appear to be problems between the control system design and the host environment, they can be resolved before attempts are made to implement the system. In this way system designers and system purchasers can gain greater confidence that the designed control system will not only support manufacturing operations but also deliver the benefits sought through its introduction.

The idea of assessing, or predicting, the performance of a design in advance is not restricted to planning and control systems, it has also been applied to manufacturing system design and control system design for flexible manufacturing systems (FMS). Shimizu (1991) discusses the application of an integrated modelling approach to the design and analysis of manufacturing systems, whereas Bilberg & Alting (1991) utilise simulation modelling in the development of FMS control software.

The mechanisms by which different planning and control systems can be assessed will be discussed in subsequent chapters.

4.4 The opportunity to assess the impact of control system designs

Of the design approaches reviewed above, two will be expanded here in order to emphasise the opportunities which exist to assess the performance of any proposed planning and control system(s). The two approaches are the Lucas manufacturing systems design methodology (Lucas Engineering & Systems, 1991) and that devised by Maull et al. (1990). The Lucas methodology includes a control system design stage in a wider manufacturing systems design methodology, applied to the design of cellular manufacturing system. The methodology developed by Maull et al. addresses the design and implementation of resilient CAPM systems. The discussion begins with the Lucas methodology.

4.4.1 The Lucas manufacturing systems design methodology

The adoption of a methodology in order to support the design and implementation of manufacturing systems is essential; one such methodology has been used extensively by Lucas Industries and is applied to the design of cellular manufacturing systems. The five-step methodology (Lucas Engineering & Systems, 1991), which takes a manufacturing systems perspective (Parnaby et al., 1986), is illustrated in Figure 4.1.

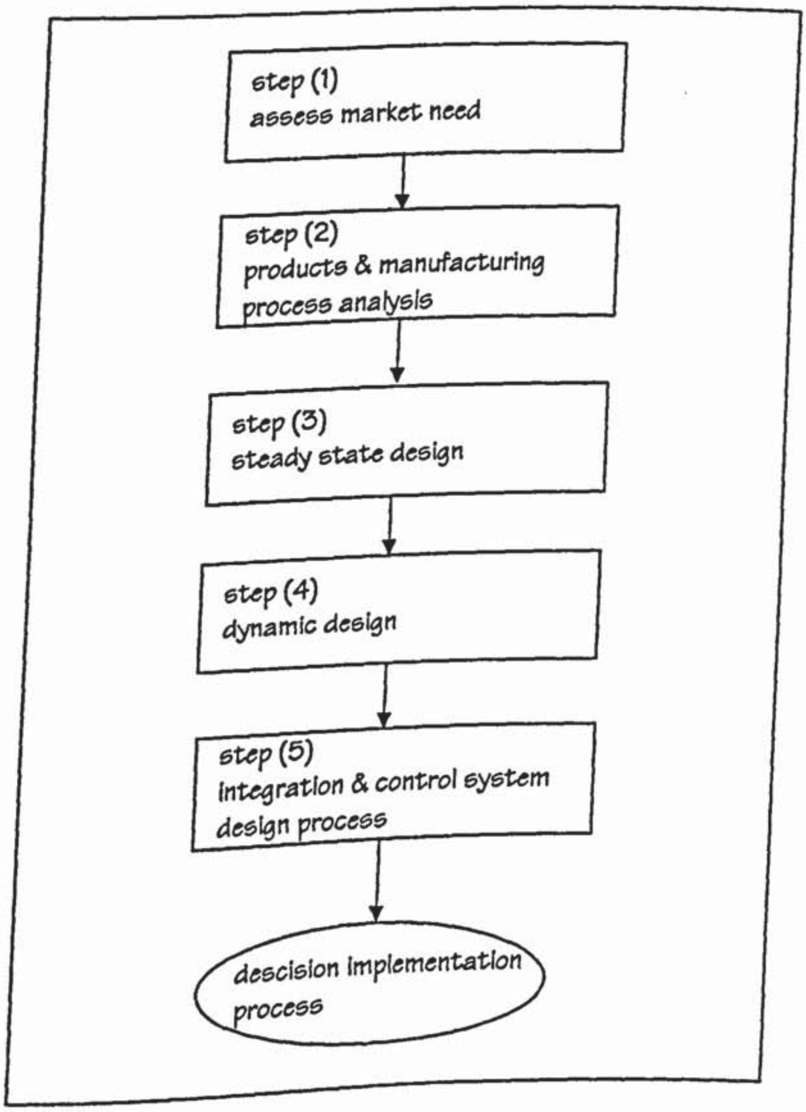


Figure 4.1: The Lucas manufacturing system design methodology (after Lucas Engineering & Systems, 1991).

The inclusion of a planning and control system design stage within this methodology recognises the importance of such systems in relation to the physical manufacturing facility. The shop floor material control systems are designed along with the mechanisms to derive more realistic production plans. Attention is paid to the runners, repeaters and strangers classification and levelled schedules. Kanban and period batch control (see for example Burbidge, 1988) are common candidates for material flow control, as is the JIT-MRP approach.

Although the application of the Lucas manufacturing system design methodology has resulted in successful manufacturing system implementations (for example, Kellock, 1992), there have been exceptions (Lewis, 1994). In a series of extensive studies Lewis has identified several problems with the above methodology which have been reflected in poor manufacturing system performance. Although the adverse results identified cannot be attributed solely to problems with the production planning and control system, several elements of particular relevance will be discussed.

It is evident from Lewis's studies that the concentration of design effort was directed more towards the shop floor. This resulted in the implementation of innovative material flow systems and simplified local control systems. However, little attention was paid to the interface between the new shop floor system and the existing top-down planning system, for example MRP. Additionally, the necessary policy changes needed to support the manufacturing system design were not properly investigated. The resulting poor performance included increased shortages and expediting, overloading of resources, increased inventory and unrealistic master production schedules. These results have some similarity with those reported by Dale & Russell (1983) and Berry & Hill (1992).

One of the problems associated with the control system design stage is that it occurs late in the overall manufacturing system design process, following the design and analysis of the physical layout. This can lead to practical difficulties: if the time-scales are short for the design and implementation process, or there is little time remaining within the project, then the design of the control system is unlikely to receive adequate attention (Lewis, 1994).

It is interesting to note that included in the dynamic design stage of the methodology is an assessment of the performance of the (steady state) design. The design resulting from the steady state design stage is based on average conditions for attributes such as operator and machine performance, product volume and variety. The dynamic design stage then tests the capability of this design to cope with the more realistic variations in these parameters; such variations are to be expected (Sinha & Hollier, 1984). Changes are then made to the design in order to improve the systems robustness to the more natural dynamics of the environment. The previous chapter emphasised the role of the planning and control system in the operations of the manufacturing system as well as its relationships to the production facilities design. In essence, the control system is a major component of the complete system subjected to variations in the above parameters and thereby should be included in the dynamic design stage. Furthermore, in the light of the problems observed by Lewis (1994), Berry & Hill (1992) and Dale & Russell (1983) it would appear fundamental to include the planning and control system in this stage.

If it were possible to assess the impact of the control system design on the manufacturing system before implementation then it is believed that some, if not all, of the problems outlined above may have been avoided. At the very least the problems, or more likely the symptoms, could be identified and then appropriate measures taken to rectify the situation.

4.4.2 Resilient CAPM systems

One of the projects which was initiated by the 'state of the art of CAPM in UK industry' study (Waterlow & Monniot, 1986) addressed the issues surrounding the design and implementation of resilient CAPM systems. This work is described in Maull et al. (1990) and Maull & Childe (1993) and initially considered the causes of CAPM failure. With this understanding, they developed a methodology to identify the most appropriate planning and control system (i.e. CAPM) solution for an organisation. Although the research focused on the electronics sector, the authors believe it to be applicable across the manufacturing sectors.

The initial part of their research identified four factors which may lead to CAPM system failure:

- the system requirements were defined incorrectly;
- the system requirements were defined correctly, but the wrong system was implemented;

- the system requirements were defined and implemented, but the implementation was badly managed;
- the system requirements were defined correctly and implemented, but these requirements changed over time.

In the light of the discussion of chapters 2 and 3, the first three causes of failure are hardly surprising. However, the final factor does provide some cause for concern, as Maull et al. (1990) suggest:

> *"It is important because even when requirements are defined correctly and the correct system is specified and implemented successfully, subsequent changes in requirements can lead to failure."*

This problem also extends to systems which are currently in operation; unless these systems are able to respond to new requirements or demands placed upon them, then there is a potential for failure. New demands may be placed on a manufacturing organisation due to, for example, changes in product volumes or alterations in the company strategies. It is essential that these new requirements are supported by the planning and control system if an appropriate match is to remain. As the introduction to this chapter indicated, 'change' is a common characteristic of today's manufacturing environment. Consequently, systems of the kind described here need to be designed with changeability in mind (Pels & Wortmann, 1990).

Manufacturing organisations are themselves becoming more aware of the problems caused by changing requirements and recognise the need for greater flexibility within their CAPM solutions. The recent 'Computers in Manufacturing Survey' (Benchmark Research, 1994a) indicated that companies were seeking to purchase systems which will not only support their current needs but also any changes that may need to be made in the future. Consequently, CAPM systems need to possess sufficient flexibility to be able to cope with future changes (Maull et al., 1991).

The framework of the user-led methodology developed by Maull et al. is illustrated in Figure 4.2. The methodology aims to address the issue of resilience to change by adopting a combination of top-down and bottom-up approaches. The top-down approach integrates the business strategy and manufacturing system requirements with those of the prospective CAPM system. It considers the likely future changes and assesses their impact on both manufacturing and control. The bottom-up approach matches the CAPM requirements against possible solutions to ensure a good business fit.

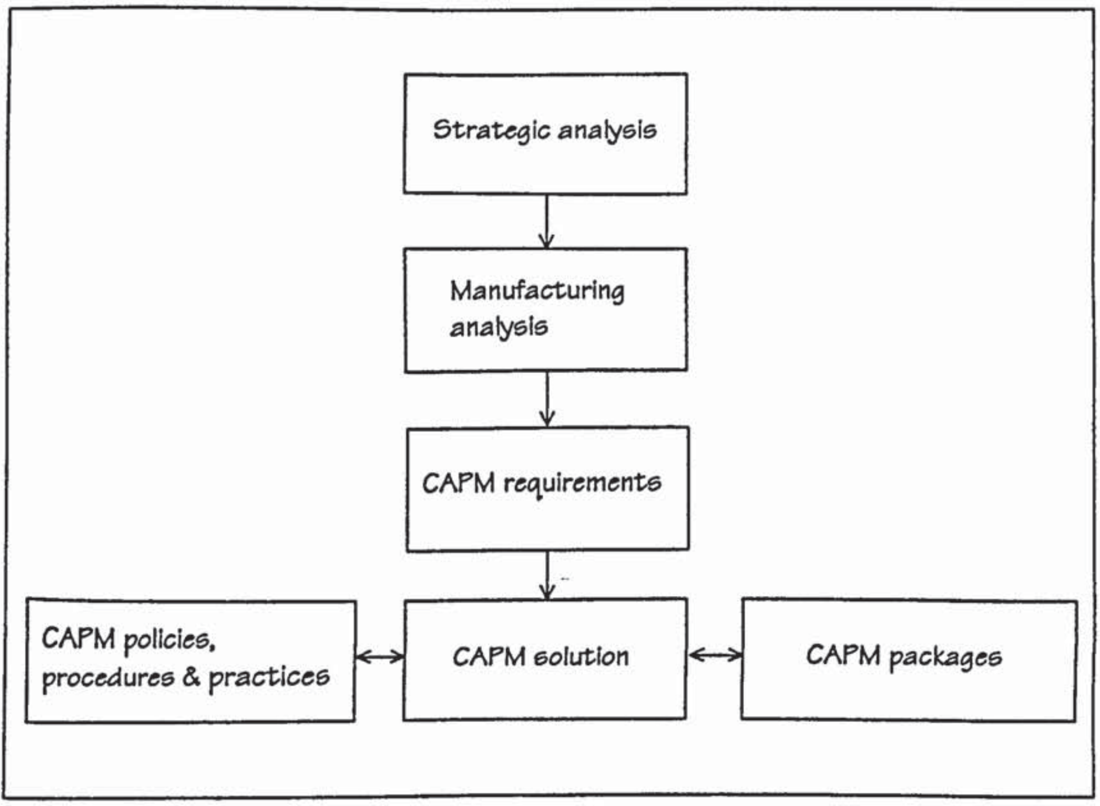


Figure 4.2: A methodology for the design of resilient CAPM systems (after Maull & Childe, 1993).

This approach is interesting because it recognises that the planning and control system is greater than simply a software solution. The role of *"policies, procedures and practices"* (Maull et al., 1990) should not be ignored since they essentially direct the different tasks of the system. These aspects of a company's infrastructure have the potential to either enhance or restrict the operations of the planning and control system. The eventual solution recognises the difference between those tasks which require computerisation and those which do not.

Although a proprietary CAPM solution is likely to be selected, the methodology acknowledges that the software package is likely to require modification. This will be required in order to improve the fit between the system and the organisation. They further suggest that the implementation of the system should be organised according to good implementation practice.

What is missing from their work is the inclusion of some form of assessment of the derived system. It would appear essential to be able to test the robustness of the control system design and assess its performance with respect to the manufacturing system. The methodology seeks to arrive at a resilient planning and control system, one which has sufficient flexibility and thereby best suits the current and future needs of the business. This, however, is not tested, or demonstrated to be so. By assessing the performance of the system (in advance) there is an opportunity (or requirement) not only to test the fit of the system but also its resilience to change.

4.5 Summary

This chapter has reviewed the area of manufacturing planning and control system design and identified both the need and opportunity to assess the performance of designs. The design and, thereby, the performance should be assessed in the context of the manufacturing system as a whole: adverse results may result if there is a mismatch between the manufacturing system and its planning and control system. The ability to evaluate control systems, in advance of implementation, will support the design process as well as providing valuable insight into how the control system affects manufacturing performance.

5. Modelling planning and control systems using discrete event simulation

The previous chapter identified the need to assess the impact of a proposed planning and control system design on the performance of the manufacturing system. This chapter will examine the applicability of different modelling techniques for analysis of the kind described. Of the different modelling techniques, discrete-event simulation is the most suitable. The discussion will begin by illustrating the manufacturing environment and thereby the context in which planning and control systems should be studied.

5.1 Some environmental characteristics

The manufacturing environment can not be considered simple; Parnaby (1986) provides an apt scenario:

> *"Imagine the complexities of, say, a factory manufacturing motor vehicle lighting systems with over 500 products each having over 100 sub-assembly or precision component parts made via a sophisticated variety of processes in a range of modern materials from metals to dough-moulded plastics with sophisticated surface treatments. Producing these products in tens of thousands per week and ensuring all the component parts of the correct dimensions and quality appear in the right place and at the right time for each lighting system to be assembled, in spite of step change transients initiated by customer programme changes, is an extraordinarily difficult task."*

The manufacturing environment is characterised by complex dynamic behaviour, whereby the dynamics themselves are represented by the different sources of uncertainty which exist. Common examples of uncertainty include sales demand variation, changes in product mix, urgent or rush orders, variations in supplier quality, supplier lead-time variability, machine breakdown, scrap and rework, operator performance variability and operator absenteeism. As the examples illustrate, sources of uncertainty exist internally and externally to the manufacturing system; unfortunately their effects often cross the internal/external boundaries. For example, an increase in a customer's order quantity will necessitate changes to production plans, material promised to other orders may be used and competing jobs may need to be re-scheduled.

Alternatively machine breakdown may lead to delays in manufacture which may affect delivery performance to customers. It is against such sources of uncertainty that the robustness of the manufacturing system must be assessed.

Assessing the performance of a planning and control system is made more difficult since the control system cannot be considered in isolation. The relationships between the planning and control system and the manufacturing facility, discussed in chapter 3, are tightly entwined. It should also be recognised, however, that the planning and control systems contribute to the dynamics of the manufacturing system. For example, the generation of work orders will determine what products are to be manufactured and thereby which materials will be consumed; the scheduling of these jobs will then allocate operators, machines, tooling and materials. If (or when) any problems occur, such as machine breakdown, the control system will attempt to bring the factory back into line with the production plan. Depending upon the fit between the control system and the manufacturing system, the dynamics of the situation may be dampened or intensified. Any analysis of the control system must, therefore, pay special attention to the relationships between the control system and the manufacturing facility.

It will be apparent from the above discussion that events which take place on the shop floor, or from sources external to the manufacturing organisation, prompt response from the planning and control system. The effects of these responses are then reflected in the subsequent behaviour of the manufacturing operations. If the two systems were considered in isolation, then these effects would not be observed. It is by addressing the system in a more holistic way that these 'emergent properties' can be identified.

It is within the context described that planning and control systems should be studied. Consequently, any modelling technique used to assess such systems must be capable of capturing the richness of the system's dynamics if meaningful results are to be obtained. Furthermore, it is not enough to model the planning and control system and the manufacturing facility in a disjointed manner. Any modelling technique must be able to adequately address both areas in order to build comprehensive and representative models; it is only via such models that the important emergent properties can be properly understood.

5.2 Which modelling technique?

A simple spectrum of manufacturing systems modelling techniques, illustrated in Figure 5.1, has been offered by Jackman & Johnson (1993) and provides a suitable starting point for discussion. They suggest that most practical modelling techniques will be found between the two extremes.

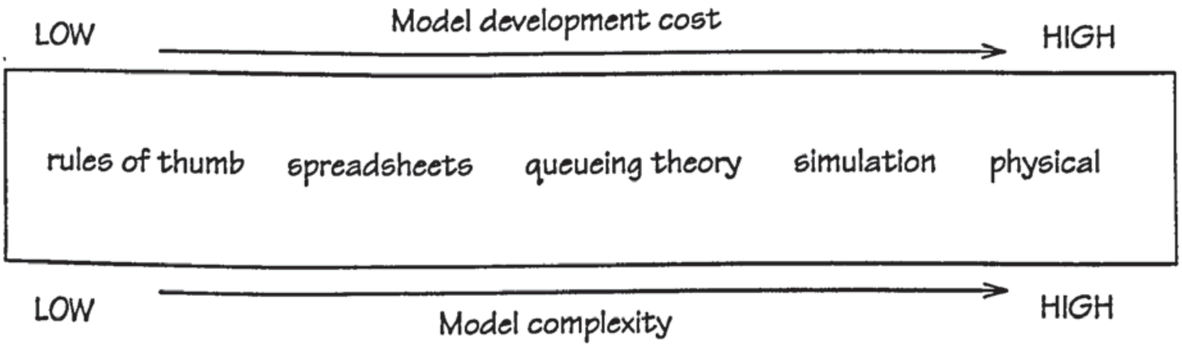


Figure 5.1: A spectrum of manufacturing systems modelling techniques (after Jackman & Johnson, 1993).

5.2.1 'Rules of thumb'

The 'rules of thumb' approach would appear to be the least rigorous and one typically used by experienced practitioners. It is suggested that those design approaches, discussed in chapter 4, which are more oriented towards selection and best practice of planning and control systems, lend themselves towards the 'rules of thumb' category. This is especially relevant if the designer/engineer has experience in designing similar systems. An analysis of the particular manufacturing circumstances would identify that control system 'x' be the most suitable: for example, a steady demand pattern suggests the use of a pull type of control system. If all aspects of the company are sufficiently addressed then an overall solution will be derived.

Although models of this kind imply suitability they do not evaluate how the design will perform before proceeding to the implementation stage. The characteristics of one manufacturing organisation differentiate it from other organisations, which means that the generality implied by this type of model can be misleading. Ptak (1991) noted that the planning and control solution for one company may not necessarily be appropriate for other companies. This is particularly relevant in the light of the characteristics discussed earlier.

5.2.2 Queueing theory models

Queueing theory models, which belong to the family of mathematically based (or analytical) modelling techniques (Papadopoulos et al., 1993), have become popular in recent years (Askin & Standridge, 1993). Queueing models are among the techniques which provide a means of including the dynamics and stochastic nature typical of manufacturing systems. In the manufacturing context, each work station is viewed as a simple queueing system and thereby represents the complete manufacturing facility as a network of queues.

Models can be built and evaluated with both relative ease and speed using queueing theory, particularly if specialist software tools such as MANUPLAN (Suri & Diehl, 1985) are used. There are, however, a number of assumptions which restrict the application of queueing theory models (Jackman & Johnson, 1993; Askin & Standridge, 1993; Chaharbaghi, 1990). These underlying assumptions include:

- steady state operations;
- service times defined by statistical distributions;
- no restrictions on buffer (queue) sizes;
- processing of queues based on the 'first-in-first-out' (FIFO) rule;
- job arrivals are independent.

These assumptions are somewhat unrealistic. For example, it is often the case that in order to reduce the problems of setting machines, similar jobs will be scheduled together; job arrivals are, therefore, more likely to be influenced by other jobs in the system. As Jackman & Johnson (1993) indicate, queueing network models are unable to model scheduling and shop-floor control procedures, relying on FIFO. This is unlikely to be the case in actual manufacturing situations. Consequently, the operational aspects of the manufacturing system, i.e. planning and control, are not adequately catered for. The assumption of an infinite capacity buffer will make queueing theory unsuitable for modelling situations which require limited buffers, for example those controlled by kanban. Steady state operation is seldom, if ever, reached. McKay et al. (1988) studied many manufacturing companies and found that the shop floor rarely maintained stable operations for longer than half an hour, unexpected events being commonplace.

Evaluating systems using steady state conditions fails, by definition, to capture the transient nature of manufacturing production. There exist many instances of transience triggered by, for example, machine breakdown, customer demand, operator performance and supplier lead-times. The effects of these problems need to be studied if the robustness of a manufacturing system is itself to be understood. As Pidd (1992a) points out, it is not the steady state operations which are of particular interest but what happens to operations after an event such as a machine breakdown.

There are additional drawbacks: the mathematical details of queueing theory can be quite complicated and, as indicated by Papadopoulos et al. (1993), the technique may only be clear to the model developer(s). This has obvious disadvantages if users are to be encouraged to participate, if not lead, the design process.

Queueing network theory is more appropriate for the preliminary planning and design stages of a manufacturing system (Suri & Diehl, 1985; Jackman & Johnson, 1993): the effects of the underlying assumptions are less contentious. It is interesting to note that Suri & Diehl (1985) suggest that MANUPLAN should precede detailed simulation (to be discussed later), the latter being used to finalise a design. They further suggest the use of MANUPLAN and simulation as complementary tools, examples of which can be found in Shimizu (1991) and Huettner & Steudel (1992).

Queueing theory does provide an opportunity to investigate the dynamics of a manufacturing system. However the underlying assumptions restrict its use in detailed modelling of manufacturing systems and, more pertinently, the inclusion of appropriate planning and control system models.

5.2.3 The IDEF methodology

The Integrated Computer Aided Manufacturing (ICAM) program of the US Air Force identified a need to better describe, specify and analyse their manufacturing activities; the IDEF (ICAM Definition) methods were subsequently developed. Although there are now several different IDEF methods or techniques (Plaia & Carrie, 1995), IDEF0, IDEF1 and IDEF2 are perhaps the most well known. Maji (1988) summarises the purpose of the IDEF methodology as one which captures, through graphical means, the characteristics of the manufacturing environment and obtains solutions to the following questions:

- what functions are being performed?
- what are the information and data requirements needed to support these functions?
- how do the functions and information change over a period of time?

The three methods (IDEF0, 1 & 2) attempt to answer these questions by providing three different viewpoints. IDEF0 is used to produce a functional view (or model) of the area under investigation; IDEF1 models the information and data requirements in order to support the functions described by IDEF0; the third, IDEF2, describes the time-varying behaviour of the manufacturing system. These models are intended to complement each other.

Of the three IDEF modelling approaches briefly described, IDEF0 has perhaps received the most attention in the UK: see for example Colquhoun et al. (1993). The approach is often used by companies to support changes to the way they do business. There have been several different applications of IDEF0: in process planning (Colquhoun & Bains, 1991); planning and control system design (Banerjee et al., 1993); business process re-engineering (Maull et al., 1994). The technique is based on structured analysis techniques and produces a hierarchical description of the functional aspects, and their relationships, within an organisation or department. The models are developed to represent an understanding of current operations ('as-is') and to express the structure of future operations ('to-be'). The graphical representation enables more focused discussion between modeller and user.

Although there are advantages to a structured analysis tool of this kind, there are limitations to IDEF0; as Plaia & Carrie (1995) point out:

> *"IDEF0 allows the user to describe what an organisation does, but it does not let us consider the specific logic or timing associated with these activities."*

Busby & Williams (1993) voice additional criticisms of this technique. The important point in the context of this discussion is that both IDEF0 and IDEF1, the information and data model, are unable to address the dynamics of the system being investigated. This will not allow the performance of a planning and control system to be properly assessed. IDEF2, however, has been specifically developed to examine the time-dependent characteristics of a system.

According to Bravoco & Yadav (1985) IDEF2 can be used in two ways: as a descriptive tool and as an analysis tool. As a descriptive tool it can be used to identify the elements of the system that affect or cause the system's behaviour to change over time; in this context IDEF2 documents the dynamics. However, as an analysis tool an IDEF2 model can be used to assess the performance of the system in the light of the identified time-varying characteristics. This experimentation, it is suggested, is achieved using simulation. The implication from the text, however, is that the IDEF2 model is used to specify a simulation model rather than perform the analysis. The emphasis remains, therefore, with description and documentation.

As with the case of queueing network models, discussed previously, IDEF models themselves do not adequately provide the means to assess the performance of manufacturing systems - neither the manufacturing facility nor the planning and control system.

5.2.4 The GRAI methodology

The GRAI methodology is one of the few techniques developed specifically for use in production management (Rolstadås, 1988a). A brief explanation of the GRAI method will be included here, a more complete explanation can be found in Doumeingts (1985) and Doumeingts (1988). There are three major components to the GRAI method: a conceptual model, GRAI grids and GRAI nets.

The GRAI conceptual model disaggregates the production system into three subsystems: the physical subsystem, the decision subsystem and the information subsystem. The structure of the conceptual model, illustrated in Figure 5.2, provides a visual representation of the relationships between the decision centres at the different levels within the organisation (Bennett & Forrester, 1993).

The physical subsystem represents the machines, materials and operators, and is organised along cellular manufacturing principles. The decision subsystem has a hierarchical structure of decision levels; each decision level is determined according to a planning horizon (i.e. the length of time for which the decision is valid) and a review period. The decision levels are composed of one or more decision centres which interact with each other. It is the information subsystem which provides the necessary information to the different decision levels and provides the links between the decision and physical (operating) subsystems. Together the decision and information subsystems form the production management system.

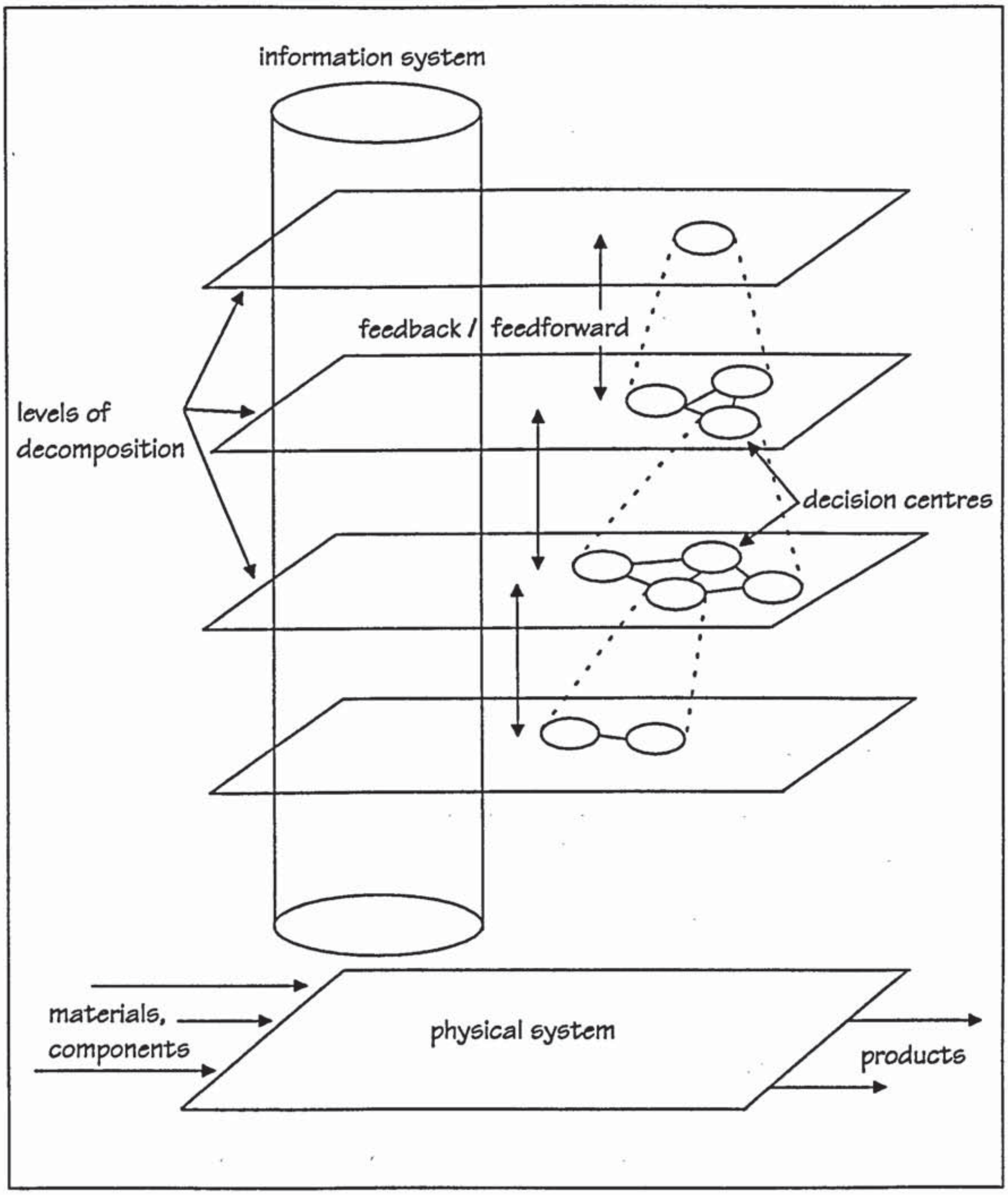
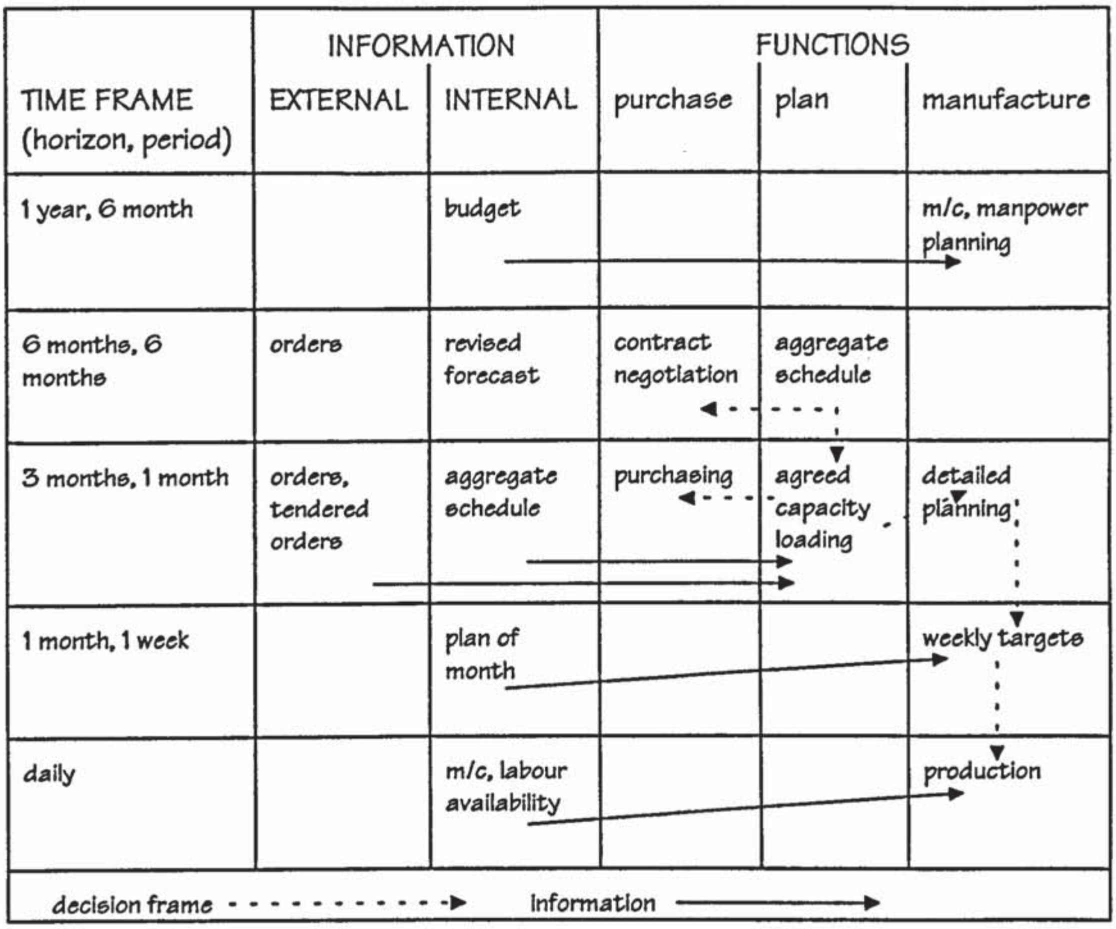


Figure 5.2: The GRAI conceptual model (after Bennett & Forrester, 1993).

A GRAI grid (Figure 5.3) provides a hierarchical representation of the different decision centres of the system and the links between them; the grid maps out the different activities through time. The columns of the grid represent the functional activities; the rows represent the time dimensions; the column-row inter-sections represent the decision centres of the conceptual model. A top-down analysis of each of the activities over the different time frames, which identifies the relationships between the decision centres and the information flows, is mapped on to the grid. The detailed structure of the decision centres is obtained through a bottom-up analysis and represented by a GRAI net.



	INFORMATION		FUNCTIONS		
TIME FRAME (horizon, period)	EXTERNAL	INTERNAL	purchase	plan	manufacture
1 year, 6 month		budget			m/c, manpower planning
6 months, 6 months	orders	revised forecast	contract negotiation	aggregate schedule	
3 months, 1 month	orders, tendered orders	aggregate schedule	purchasing	agreed capacity loading	detailed planning
1 month, 1 week		plan of month			weekly targets
daily		m/c, labour availability			production

decision frame - - - - - - - - - - -▸ information ——————▸

Figure 5.3: An example GRAI grid.

The assessment of the GRAI grid and associated nets provides an opportunity to identify inconsistencies in the production management system and develop recommendations for improving the structure and operation (for example, Wainwright et al., 1994; McCarthy et al., 1994). Whilst these benefits, which are also attributable to the IDEF methodology, are seen as important they do have limitations. If, for example, there are no inconsistencies it does not mean that the designed system will deliver the desired performance. Suppose the analysis identifies that all of the information required at a particular decision frame is accounted for, then no modifications will be recommended. However, problems may not be occurring because of a lack of information but because of either the timing of its receipt or its accuracy. The same problem can be attributed to any recommended modifications.

Ho & Ridgway (1994) discuss the use of the GRAI methodology for the design and analysis of planning and control systems in the context of cellular manufacturing systems. It is proposed by the authors that through such analysis it is possible to satisfy operational criteria: for example, responsiveness to new customer orders and a reduction in lead times. These aspects cannot be assessed without inclusion of some form of assessment of the dynamics. Although the time-varying characteristics are documented by the GRAI grid it is not the same as including the stochastic elements such as the late arrival of material. The representations offered by the GRAI methodology assume the timing of events occur as expected; although manufacturing companies may strive to behave in this way, they are rarely able to achieve it.

The lack of attention to the dynamics of the environment remains a significant omission. One of the criticisms of the GRAI conceptual model is that it only concerns itself with structure and provides only a static view (Bennett & Forrester, 1993). Furthermore, as MacIntosh (1992) points out, the understanding gained from GRAI grids and nets is based on interpreting static models of what is essentially a dynamic environment; he goes on to suggest extending any analysis to include simulation.

5.2.5 Discrete event simulation

Chaharbaghi (1990) noted that:

> *"Discrete event simulation has emerged as one of the most appropriate modelling tools to plan the design and operation of production systems."*

This is a view echoed by Wu (1994). Both authors have recognised that simulation, in comparison with other analytical modelling techniques, is the only one capable of adequately dealing with the complexity and dynamics found in manufacturing systems. Shannon (1992) makes the point that simulation is one of the most powerful tools available to those who are involved in the design and operation of complex systems. Manufacturing planning and control systems, as an integral component of a manufacturing system, are an important application domain for simulation: there are opportunities to investigate both design and operational issues (Browne, 1988).

There are essentially two forms of simulation, continuous and discrete, which correspond to the two forms of change which can take place over time (Law & Kelton, 1991). In a batch manufacturing situation events happen at discrete points in time:

material arrives, operators begin their shift, machines are switched on, an order is released, and so on. It is only at these points in time that changes take place; the number of works orders changes only when a new order is released or when an existing order is complete. There are changes which take place continuously with respect to time: for example, changes in temperature and velocity. This is not to say that all systems are either continuous or discrete, many systems involve both forms and combined models do exist. However, where the emphasis is on discrete change, discrete simulation is used whereas if the emphasis is on continuous change then continuous simulation is used (Law & Kelton, 1991; Carrie, 1988).

Typically continuous simulation involves the use of difference equations (Law & Kelton, 1991); the mathematics describes how the variables included in the model change over time. When applied to the analysis of batch manufacturing systems discrete simulation is more appropriate. This extends to the modelling of planning and control systems: events occur at discrete points in time, for example the receipt of orders from customers, performing the material planning function, the update of the inventory records and so on. This discussion will concentrate on discrete event simulation, which is identified in the literature as the dominant type.

Discrete event simulation is the focus of several texts, for example Pidd (1992a), Law & Kelton (1991) and Carrie (1988), consequently only a brief discussion will be included here. (The term simulation will be used within the remainder of the text.) Put simply, a simulation model consists of entities, the logic to govern the behaviour of the entities and a mechanism to control the passage of time. Typical entities in a manufacturing model would include machines, operators, parts, trucks, containers, departments and orders. The intention of the simulation model is to imitate what goes on in the 'real' system. In this respect simulation allows the construction of a more comprehensive model of the manufacturing system than can be achieved using queueing theory, IDEF or the GRAI methodology.

There are additional advantages to the use of simulation:

- it offers great flexibility;
- it is the only technique to be able to model complex dynamics;
- its ability to model transient behaviour;

- its ability to model complex material and information flows;
- it facilitates experimentation without disturbing the operations of the actual system;
- the models are more understandable to users and operational personnel.

When all of these characteristics are combined together, simulation offers a more complete modelling environment than that offered by the techniques previously discussed.

Simulation has been applied to different areas of manufacturing, for example manufacturing system design (Shimizu, 1991; Lewis, 1994) and the design and analysis of flexible manufacturing systems (FMSs) (Bilberg & Alting, 1991; Boyle & Kaldos, 1994). Earlier work by Crookall (1985) documents that thorough analysis of both the strategy and manufacturing system is an important initial aspect of determining FMSs. Simulation is considered as an important tool in achieving this, emphasised by the observation that some major companies will only consider schemes for investment if they have first been simulated.

Of course simulation is not without its problems. It is often criticised for being too labour and time intensive. It is perceived that the amount of time required to design and build an adequate model is not insignificant, when project time is limited this can prove to be a decisive factor. The model builder will require appropriate simulation skills in order to build, test and experiment with a simulation model. Although the availability of manufacturing simulators (to be discussed later) should ease some of the problems, the requirements can prove daunting for the non-expert.

Following the observation of Crookall (1985), one would expect the application of simulation in manufacturing industry to be commonplace. This does not appear to be the case. A study of the use of simulation in UK manufacturing industry (Simulation Study Group, 1991) found awareness of simulation as a technique and its application to be relatively low. However, the study did note that those companies which had used simulation had expressed a great deal of satisfaction with the technique. Furthermore the study estimates that there are significant financial benefits to be realised from its application.

5.3 Summary

This chapter has examined some of the different modelling techniques with the intention of identifying that most suitable for modelling planning and control systems. The characteristics of the manufacturing environment are typified by the complex interactions and dynamics and, consequently, the important relationships between the shop floor and the planning and control system should not be overlooked. In this context few modelling techniques are able to provide adequate assessment. Simulation is, however, able to model detailed dynamics and offers the capability sought.

6. The application of simulation to modelling planning and control systems

The previous chapter identified discrete event simulation as the most suitable modelling technique for evaluating manufacturing planning and control systems. This is particularly true when considering the impact of a control system design on the performance of the manufacturing system. However, it cannot be claimed that the application of simulation to planning and control issues is new. This chapter will establish that the current application of simulation to planning and control issues is limited. Moreover, whereas operational issues have been the main focus of attention, the application of simulation to the more fundamental design of manufacturing control systems is rarely considered. The current state of simulation tools does not encourage matters. This chapter will conclude by defining the requirements of a simulation-based tool which will provide the necessary functionality to address control system design issues.

6.1 Manufacturing planning and control systems and simulation - an overview

By way of introduction and before embarking on a more detailed critique of the current literature, an overview will be included here. This overview provides a general picture of the type of research which is taking place: the subject area and the enabling tools and techniques. For the purposes of this overview it was considered pertinent to review a recognised academic journal in the field of manufacturing: the International Journal of Production Research (IJPR) was chosen. This particular journal was chosen because it specifically reports production and manufacturing research and should, therefore, offer a more representative sample of the academic emphasis in this subject.

All volumes of the IJPR for the years 1989-1994 were reviewed and all studies which addressed manufacturing planning and control systems and/or associated issues were noted. These studies were then further divided into three categories:

- those which included some form of simulation;
- those which included some form of modelling (or experimentation) other than simulation;
- those which did not include any discussion of simulation, modelling or experimentation.

The different planning and control issues which were addressed by the studies were noted in order to identify the main focal points of research effort. Furthermore, where possible for those simulation specific studies, the simulation medium was also noted. It should be emphasised that this overview does not include line-based production or flexible manufacturing systems (FMS).

The results from the six volumes of the IJPR are summarised in Table 6.1. It is interesting to note that the total number of planning and control related papers per volume proves to be fairly consistent, perhaps reflecting an equally consistent editorial policy.

Year, volume (number)	Total Papers	CATEGORY				Top 3 issues addressed: Issue (number of papers)
		(i)	(ii)	(iii)	TOTAL	
1994, 32(1-12)	172	11(8)	18	5	34	scheduling (16), kanban/pull (8), order release / sequencing / dispatching (5)
1993, 31(1-12)	175	10(9)	20	6	36	scheduling (15), batch / lot sizing (8), order release / sequencing / dispatching (3)
1992, 30(1-12)	177	11(8)	17	8	36	scheduling (9), batch / lot sizing (8), JIT / kanban (7)
1991*, 29(1-12)	158	7(7)	23	4	34	scheduling (13), JIT / kanban (8), batch / lot sizing (6)
1990, 28(1-12)	156	6(4)	12	7	25	scheduling (9), batch / lot sizing (5), JIT / kanban (5)
1989**, 27(1-12)	126	11(9)	3	17	31	scheduling (15), batch / lot sizing (6), order release / sequencing / dispatching (4)
TOTAL	964	56(45)	93	47	196	

CATEGORIES:

(i) included simulation, figure in brackets confirmed as discrete-event simulation;
(ii) included modelling or experimentation other than simulation;
(iii) no simulation or modelling/experimentation included.

NOTES:

(*) 1991, 29(2) - guest editorial on Robotic Assembly;
(**) 1989, 27(3) - special edition on Robotics.

Table 6.1: A review of the International Journal of Production Research, 1989-1994.

Of the 56 papers which included the use of simulation only 45 could be confirmed as meaning discrete-event simulation. (Consistent with previous and subsequent chapters, 'simulation' will be used to represent discrete-event simulation, unless otherwise stated.) Furthermore, of the studies which employed some form of modelling, the majority (62%) did not use simulation. Although some of these studies used some form of mathematical modelling (for example, queueing theory or dynamic programming), the bulk of the studies evaluated the derived solution using sample, or synthetic, data. This is not the same as an evaluation under dynamic conditions.

Of the studies which were in the area of manufacturing planning and control systems, the three most popular control issues are included in Table 6.1. It is quite evident that scheduling rules, dispatching rules, lot sizing policies as well as JIT/kanban issues receive the bulk of the research attention. Typically the studies evaluate the relative performance of different scheduling rules or lot sizing policies under different conditions. For example, the effects of demand uncertainty on lot sizing policy in a MRP based environment. Alternatively, most of the kanban studies consider the problem of determining an appropriate number of kanbans in order to achieve a certain performance. This does not mean that other issues, such as capacity or inventory issues, are not considered but that they are in a distinct minority in comparison to those listed.

Simulation Medium	YEAR						
	1994	1993	1992	1991	1990	1989	TOTAL
SLAM, SLAM II	3	2	5	2	1	6	19
SIMAN (CINEMA)	3	5	2	1	1		12
Q-GERT					1	1	2
Turbo Pascal	1		1				2
BASIC					1		1
GPSS						1	1
SIMSCRIPT		1					1
GEMS II		1					1
own program				1		1	2
unknown	1			3			4

Table 6.2: Different ways used to build simulation models.

Table 6.2 summarises the different ways in which simulation models have been constructed. What is surprising is that despite the introduction of simulators (Banks et al., 1991), and in particular manufacturing simulators, the dominant method of model building undoubtedly remains with simulation languages. It is fair to say that of the volumes reviewed, SLAM (O'Reilly, 1993) and SIMAN (Kasales & Sturrock, 1991) together represent the most popular medium through which simulation models are developed. It should be noted, however, that many of the studies included specific FORTRAN based subroutines which interfaced with the particular simulation medium.

To summarise, the above overview indicates that scheduling rules, lot sizing policies and determining the number of kanbans are the most popular of the planning and control issues addressed. Where simulation is used, simulation languages appear to be the preferred development medium.

6.2 Manufacturing planning and control systems and simulation - a critique

The above overview has identified some of the major (academic) themes associated with the application of simulation to manufacturing control systems modelling. The following more detailed discussion, based on a much wider range of published papers, will demonstrate the limitations of current approaches to this problem.

6.2.1 Focus of modelling attention

There are two general applications of simulation which are reported in the literature: one addresses policy issues such as lot sizing policies and scheduling rules, the second compares the relative performance of different systems, typically MRP, kanban and ROP.

By far the most popular application is to address the issues involved with lot sizing policies and scheduling rules. The intention of these studies is to evaluate the relative performance of different policies under different operating conditions. Gardiner & Blackstone (1993), for example, compare the performance of six dispatching techniques in a MRP environment; Brennan & Gupta (1993) investigate the effects of demand and lead time uncertainty on different lot sizing policies in a MRP environment; Mittal & Wang (1992) use simulation to determine the number of kanbans in a JIT based production system.

Research into scheduling rules is clearly a favoured topic but, as both Browne et al. (1988) and McKay et al. (1988) point out, much of the developed work is still far from being widely used by practitioners. This is a view reiterated by MacCarthy & Liu (1993). Similarly, the development of sophisticated lot sizing techniques may attract considerable research interest but little industrial application (Browne et al., 1988). Because issues, such as scheduling and lot sizing, are only components of the wider planning and control system, a more holistic approach is recommended.

The second area of application is that of comparing different planning and control systems. Ovrin (1990) reviews a number of studies, both simulation-based and analytical, which evaluate the performance of different planning and control systems in different manufacturing systems. Typically studies compare systems from MRP, ROP and kanban. For example, Rees et al. (1989) compare MRP using lot-for-lot ordering policy with kanban; Sarker & Fitzsimmons (1989) attempt to evaluate the impact of push and pull systems on the efficiency of a production line.

Chu & Shih (1992), in their review of simulation studies in JIT production, make an interesting observation: the emphasis of the JIT research to date is more towards the wider issues than those of system design. The effectiveness of JIT systems is contrasted with other control systems, alternatively the critical factors which lead to successful JIT-based systems are investigated. Their study failed to identify any studies which used simulation to evaluate JIT system designs.

The problem identified by Chu & Shih (1992) is not necessarily restricted to JIT studies. Simulation studies generally appear to be focused on the more operational issues (i.e. scheduling rules, lot sizing policies, etc.), whereas the application of simulation to the more fundamental design of planning and control systems is rarely, if at all, considered.

6.2.2 Scope of simulation models

In the majority of cases, the models built in order to support the studies, particularly those outlined in the previous section, cannot be applied to alternative studies; these models are, therefore, of limited value.

Mittal & Wang (1992) used CADOK (Computer Aided Determination Of Kanbans) which is a specific simulator for modelling kanban environments. Clearly this would not be suitable for modelling push systems or a combination of both approaches. Furthermore one of the limitations of CADOK, acknowledged by the authors, is the lack of material handling capabilities. This restriction not only misses an important element of kanban, i.e. the material handling process, but limits its application to single card (production) kanban systems. Similarly, the models used by Gardiner & Blackstone (1993) and Brennan & Gupta (1993) are only applicable to particular MRP experiments. There have been simulation models constructed to address problems of a particular manufacturing environment. For example, the models used by Hicks & Braiden (1995) and Hendry et al. (1993) are specifically targeted at the make-to-order (MTO) sector.

Simulation models which have wider applicability than those described above are less common. However, a simulation model capable of representing different manufacturing and control environments has been developed by Krajewski et al. (1987). The FORTRAN-based simulator, called MASS (Manufacturing Simulation System), includes four different production and inventory control systems: material requirements planning (MRP), reorder point (ROP) control, MRP/ROP hybrid, and kanban. A model of the shop floor operations is also included.

Although this does provide different production system options within the same environment, there are a number of restrictions. The representations offered are general and are, therefore, limited to more standard interpretations of the respective planning and control systems. Therefore, company specific interpretations of a particular system will be difficult to represent accurately. Secondly, the model construction means that expansion to include new control systems, or variations on a theme, would not be possible. It was pointed out in chapter 3 that a single planning and control solution is unlikely to provide adequate support for an organisation and that hybrid systems should be considered. Browne et al. (1988) have suggested that future production management systems are likely to be based on the hybrid model. For example, the combination of MRP and JIT principles (Parnaby et al., 1987; Flapper et al., 1991) is one alternative hybrid system. MASS, however, can only support MRP/ROP. Expanding the model to include other alternatives is likely to be difficult and thereby exclude it from use in design oriented situations.

6.2.3 Simplifying assumptions

One of the most contentious issues surrounding many of the studies reported in the literature is the use of simplifying assumptions (McKay et al., 1988). Gardiner & Blackstone (1993) summarise the situation:

> *"Often, in an attempt to isolate effects, generalize results, or simplify methodology, the research problem is reduced to the point where results are misleading and not applicable to actual practice."*

In their review of scheduling research MacCarthy & Liu (1993) list some of the simplifying assumptions which frequently appear in the literature:

- machines are always available and never breakdown;
- a job can be processed on at most one machine at any time;
- all jobs are available at the start of processing;
- once an operation is started it is continued until complete.

The review outlined in the previous section noted similar simplifying assumptions as well as, for example, zero machine setting time, zero material transfer times and a zero scrap rate.

The shop floor representations, with respect to the number of machines or work stations and parts, also suffers from similar simplifications. Gardiner & Blackstone (1993) include one of the more substantial models: 15 work stations and a total of 158 parts. At the other end of the spectrum, Cook (1994) utilises a model consisting of a single part and 5 work stations. Wisner (1995) reviews the research in connection with order release policies in job shop scheduling. Included in his review is a summary of the simulation based research in this field. The majority of the simulation studies use hypothetical models to characterise the manufacturing facility. Of the 26 studies only 5 used real data, with, for example, Browne & Davies (1984) including 80 machines and 100 labourers in their model. However, of the remaining 21 studies the average number of machines and operators is 6. These are hardly representative of the typical manufacturing organisation.

There are good reasons, of course, for these and other assumptions. The manufacturing environment is dynamic which therefore increases the complexity of the problem. If certain assumptions can be made then the problem becomes more manageable, but in doing so contradicts the actual production environment and thereby affects the realism of any derived results. This is the crux of the point raised by Minifie & Heard (1985):

> *"Previous research projects have established the effects of a limited number of policy variables on a restricted set of performance measures in highly simplistic environments. In general, these experiments were carried out in simulated environments barely rich enough in realism to permit the manipulation of the experimental variables."*

Based on such experimental environments, Minifie & Heard (1985) question the conclusions which have been drawn. Furthermore, the application of simplifying assumptions leads to the study of hypothetical models and consequently actual company implementations are rarely considered.

6.2.4 Model algorithms rather than systems

The type of models which are constructed concentrate primarily on the underlying algorithm of the respective planning and control system. This is particularly true of MRP and ROP studies. Although this approach may be prudent in terms of simplifying the problem, it does limit the modelling opportunities. Furthermore, it does not recognise that planning and control systems are much more than algorithms or computer systems. Maull et al. (1990) noted that in addition to the planning and control system there are *"policies, procedures and practices"* which together contribute to the infrastructure of an organisation. The operation of the control system is governed by such policies and procedures: examples include the mechanisms to update the inventory system and the shop floor reactions to capacity overloads.

The existence of the policies and procedures is an element of the control system design not often referred to, but which can either enhance or restrict day-to-day operations. Gardiner & Blackstone (1993), in their study of the impact of different lot sizing policies in a MRP environment, acknowledge the role played by management in real operations. They suggest for example, that if a particular lot sizing policy was not performing satisfactorily management would react and replace it with one more suitable. The importance of human involvement in manufacturing operations has also been recognised by MacCarthy & Liu (1993) and Prabhu et al. (1992).

The inclusion of policies and procedures, typically via simulated human intervention, in a simulation model would have important implications. Not only would the richness of a model be improved but a much wider perspective on the various issues would be possible. In the case of lot sizing policies, a more dynamic approach to policy review could be included. For example, as circumstances, such as the demand for a product, varied then alternative policies could be investigated. This is significantly different from the situation whereby a specific policy is fixed for the duration of a simulation experiment.

6.2.5 Coverage of the planning and control hierarchy

Chapter 2 discussed the relationships between the different levels of the planning and control hierarchy and how higher level decisions impose constraints on lower level decisions. These relationships are rarely investigated through simulation-based experiments. Tsubone et al. (1991) discuss the problem of designing a hierarchical production planning system which recognises the inter-relationships between the different levels. They develop an analytical model but provide only a numerical example by way of an illustration; there is no mention of simulation. Although the studies which investigate the effects of different lot sizing policies on shop floor operations do bridge two levels of the planning and control hierarchy, there are few other examples.

MacCarthy & Liu (1993) highlight this problem in relation to the many scheduling studies:

> *"Classical scheduling theory has tended to consider scheduling problems in isolation from the higher levels of the production planning and control function. The interaction between scheduling and higher level decision making is not taken into account. ... scheduling problems do not arise in isolation as discrete activities divorced from any contextual environment."*

If an understanding of the complete planning and control system is to be gained then the important interfaces between the different levels of the planning hierarchy must be included in any model. This is particularly true if the emergent properties are not to be overlooked. In the area of manufacturing systems design, Lewis (1994) describes the severe problems which can occur if the interface between the higher level planning function and the shop floor control system is neither properly designed nor understood.

6.2.6 Integration of real systems within a simulation model

It is possible to overcome some of the inadequacies previously identified by integrating different tools which complement each other. There are a number of examples of integrating simulators or simulation models with other software in order to provide a more realistic environment, for example Umeda(1992), Love et al. (1992) and Clarke (1988).

Love et al. (1992) have integrated a manufacturing simulator with a commercial MRP system and a commercial accounting system to form an extensive simulation model. Earlier, Clarke (1988) linked a commercial MRP system with a detailed model of the shop floor operations. The use of real systems in this ways offers significant benefits: reducing validity problems (it is the real system and so it must be valid), easing data management problems (the data is already present), providing a familiar user interface, and allowing direct translation of model findings into new policies (the model and the real system are identical).

Whilst integration of systems does offer a number of advantages, there are significant drawbacks. Many systems have not been developed to support integration of the nature described above and, consequently, there are difficulties in obtaining the relevant data. Not only is the task of extracting and transferring data from one system to another technically difficult, but there are problems associated with different data formats and consistency. There are also simulation-oriented problems which arise from the timing of events; consider, for example, a net-change MRP system which will require continuous access to the current stock status when this can only be obtained from the simulator at the end of a simulation run or cycle. Although the use of real systems has proved successful when applied to operational policies (Clarke, 1988), the application cannot be extended to the investigation of design issues with respect to control systems. This is because standard, commercial packages will only permit changes to policies and not to the underlying control mechanisms.

6.3 Defining the requirements of a novel simulation tool

6.3.1 Introduction

It has been identified that there are serious limitations to the current application of simulation to planning and control issues, both design and operational. The current state of simulation tools does not encourage matters. It is possible, however, to define the requirements of a novel simulation-based modelling tool which will support the evaluation of both design and operational issues of manufacturing planning and control systems. The following chapter will describe how these requirements will be met.

The requirements can be traced from the characteristics, relationships and demands of control systems described in earlier chapters; additionally those requirements which surround the practicality and usability of such a tool are also included. In other words the requirements can be determined from the application domain.

The intention is to provide the specification of a simulation-based tool capable of modelling, and thereby evaluating, the performance of manufacturing planning and control systems. This is analogous to the simulation of shop floor designs in order to assess their performance under representative dynamic conditions; fine tuning of the final design can then be made prior to implementation. The prerequisites of the planning and control modelling tool are clear: it will need to both map well to the real world counterparts as well as provide the necessary flexibility to support the analysis of both design and operational issues.

This discussion initially considers the wider issues surrounding control system modelling before identifying the detailed requirements.

6.3.2 A suitable analogy

It is often the case that a suitable analogy or illustrative example can provide a helpful guide to understanding. In the context of this work a 'building blocks' analogy will be used in order to support the subsequent discussion. The simulation-based modelling tool can be thought of as a set of planning and control building blocks, very much along the lines of children's LEGO® bricks; because the modelling tool will eventually become a piece of software it can be appropriately considered as 'software lego'.

It is intended that the functionality of the planning and control system domain will be represented by elementary building blocks or bricks. Moreover, the application of these control system building blocks will parallel the way that a set of LEGO® bricks can be used to build a range of different models, from houses and cars to robots. In the same way a particular control system can be constructed from a combination of elementary building blocks: different planning and control systems will require different building block configurations. For example, a selection of the building blocks may be combined to form a MRP based system whereas a different combination of building blocks will represent ROP control. As with the case of LEGO® bricks, different models will consist of common bricks as well as those specific to the type of model being constructed. This means that there will be different types of building blocks.

In software terms, the building blocks will connect together without recourse to any form of programming: the user will be able to construct simulation models from the building blocks alone.

The fundamental requirement of the proposed planning and control building blocks is to support the following modelling requirements.

6.3.3 A user-oriented perspective

A useful starting point from which to identify any requirements is from the perspective of the user. It is important to provide not only the appropriate and necessary functionality but to provide it in such a way that is readily accessible. Consequently, the characteristics of the ultimate user need to be understood. In the context of this work, the user is perceived to be someone who is not a simulation expert and not a computer programmer but someone whose expertise and experience lie with both manufacturing and planning and control systems. These characteristics are important if the modelling tool is to be used by industrial as well as academic personnel.

It should be as straightforward as possible to build models of different planning and control systems. One of the common criticisms of simulation has been the time to build and evaluate simulation models (O'Keefe & Haddock, 1991; Law & Kelton, 1991). This is particularly true when the construction of simulation models requires programming. Furthermore, a user with no programming skills is unlikely to relish the challenge of gaining the necessary programming knowledge (i.e. the language syntax) in addition to developing the logic of a simulation. Christy & Watson (1983) found that among the non-academic community there was a general reluctance to use programming languages for simulation studies. In a more recent survey of simulation users, Hlupic & Paul (1993) found that users (both academic and industrial) preferred to use simulation packages rather than simulation programming languages. It is considered more suitable, therefore, to configure a model and to remove the need for programming altogether.

For a simulation tool to be user-oriented it needs to adopt the same 'world view' as the user; the terminology and concepts should map well to those found in the application domain. In this way the need for difficult translations between the functionality and terminology found in the real world and those offered by the modelling tool can be avoided.

The requirements of the simulation tool from a user's point of view have been identified: to provide the necessary concepts, functionality and terminology in such a way that the user is both familiar and comfortable, and that model building is achieved by configuration and not by programming. In line with the building block analogy, the building blocks must not only represent familiar concepts but combine and operate without the recourse to programming.

6.3.4 Support for proper integration

Chapter 3 established the tightly coupled relationship between the planning and control system and the manufacturing facility; this needs to be reflected in any model which is constructed. Furthermore, since one of the intentions of modelling is to assess the impact of a control system design on the performance of the manufacturing system, the shop floor must be included in the simulation model. The means that the complete modelling environment will include (as a minimum) a model of the planning and control system properly integrated with a model of the shop floor facilities. This provides a *"total systems approach"* (Gooden, 1988). This requirement is essential if the emergent properties between the manufacturing facility and the planning and control system are to be identified and understood.

It is important to define what is meant by 'proper integration'. The complete integration of different systems into a simulation model is not the same as linking systems together for the purpose of automated data transfer. It is not always the case that a simulation model will include both a model of the shop floor as well as a model of the planning and control system. However, of those which do, for example Umeda (1992) and Lilegdon (1992), the level of integration is limited: a set of works orders or a schedule is generated and passed to the shop floor which then drives the production operations. This situation is more akin to an interfacing of the two systems. It is not the same as subjecting both systems to events in the simulation world, in other words integrating the planning and control system in the time-phased feedback mechanisms.

6.3.5 Scope

The modelling tool will require extensive scope; this aspect has several facets and each will be outlined.

To be practical the simulation tool will need to be able to represent the standard, or the more theoretical, forms of the different planning and control systems: for example material requirements planning (MRP), re-order point (ROP), kanban and so on. Moreover, there must be a capability to model both push and pull system philosophies. It will then be possible to construct a model in which the planning system is based on ROP control; alternatively, using the same simulation tool, a MRP driven model may be constructed. Although standard models of the different planning and control system will be configured from the same 'set' of building blocks, not all of the building blocks will necessarily be the same.

Modelling standard forms of control systems does not, however, recognise the fact that the characteristics of organisations vary and these variations are reflected in the implementation of a particular planning and control system. Company implementations often deviate from the more standard models. For instance, although two companies may operate MRP systems their implementations are unlikely to be identical. The proposed tool will need to reflect these differences. This will provide an important opportunity: it will be possible to move away from the more theoretical representations to modelling actual company implementations. This would provide an opportunity to address the lack of attention to practical applications noted by MacCarthy & Liu (1993). The implication is that the results should translate more readily to what might be expected of the actual system. This becomes important when comparing and/or assessing relative performances.

The critique of simulation studies has identified a number of areas which need to be addressed if proper evaluation of control system designs is to be achieved. The modelling capabilities will need to address different hybrid systems, the complete planning and control hierarchy, and the policies and procedures of the planning and control infrastructure.

6.3.6 Flexibility

In order to support both the comprehensive scope outlined previously as well as the range of intended applications, the proposed modelling tool will need to possess extensive flexibility.

It is intended that the same simulation tool will provide the functionality so that the different planning and control solutions can be represented, for example MRP or ROP. This requires a flexible approach to model construction not found in current simulation-based tools. Although Krajewski et al. (1987) developed a modelling environment which supports MRP, ROP, MRP/ROP hybrid and kanban situations, these are in effect only four different models within the same modelling environment. This is not, however, the same as providing the elementary functionality from which MRP or ROP or MRP/ROP hybrid or kanban models can be constructed. In order to do this the structures will need to be extremely flexible. In terms of the above analogy, Krajewski et al. (1987) effectively provide four building blocks; one representing MRP, one representing MRP/ROP, and so on. The intention of this work is to provide many (more elementary) building blocks from which each of the systems can be configured.

The building block approach offers a more open approach to model construction; this is quite apparent when addressing non-standard implementations as well as hybrid systems. Furthermore, flexibility of this kind provides the opportunity to evaluate more imaginative hybrid systems.

Today's manufacturing environment is characterised by change. Pressures from customers and competitors have forced manufacturing organisations to assess their operations which has understandably led to change and has been reflected in manufacturing system designs (or re-designs) as well as business process re-engineering. For example, a move towards cellular manufacture or the introduction of JIT principles.

The planning and control systems of organisations have also been subject to change. This means that the nature of the manufacturing environment is transient. The modelling tool should reflect this and, therefore, will require structural flexibility. The necessary changes can be tested and/or determined by experimentation. The change in product characteristics may lead to the adoption of a different system: for example a product previously considered as a repeater becomes a runner. This may be due to the changing demand requirements for a product as it moves through its life-cycle. With such flexibility the opportunity exists to model the changes without having to configure a completely new model to represent each update: it is the original model which is adjusted. This characteristic would be difficult to achieve practically through the use of programming and simulation languages.

The ability to be able to modify a simulation model in this way increases the 'shelf life' of a model. Mize et al. (1992) point out that developing *"single purpose, throw away models"* is not appropriate for supporting the design or management of manufacturing systems which are subject to almost continuous change.

6.3.7 Extensibility

The development of this simulation tool will be on-going. This follows from two development issues: it is not possible to provide an exhaustive set of the building blocks by the end of the research project, and it is likely that some of the earlier versions of the building blocks will undergo subsequent modification. The structure of the whole modelling environment will need to support both these requirements. If new or modified building blocks are added, then this should be possible with the minimum of disruption to the existing 'modelling set'. The analogy extends to the LEGO® bricks: new versions may (frequently) appear but they do not render the existing sets redundant; the scope of the complete modelling environment is simply extended.

6.4 Summary

This chapter has reviewed the current application of simulation-based modelling to manufacturing planning and control systems. Although many issues are considered, none address the more fundamental design of control systems. The requirements of a modelling tool have been introduced which will support the evaluation of planning and control system design in terms of the manufacturing system. The following chapter will describe how these requirements will be met.

7. The limited provision of planning and control functionality by current manufacturing simulators

This chapter will identify how best to satisfy the requirements derived in the previous chapter. Of the different approaches to building simulation models, the route utilising a manufacturing simulator is considered to be the most appropriate. However, the functionality offered by current manufacturing simulators, as might be expected, concentrates more on the shop floor attributes and does not extend to that found in the planning and control domain. These limitations, which will be discussed, exclude current manufacturing simulators from control system design applications. To remedy this situation, the necessary planning and control functionality will be added to an existing manufacturing simulator.

7.1 Tools for simulation modelling

7.1.1 The choices

There are many ways in which simulation models can be constructed. The different approaches are typically considered along two dimensions: flexibility (i.e. the range of applications which can addressed) and ease of use.

At one extreme lies general purpose programming languages such as FORTRAN, C or Pascal. A simulation model is constructed (i.e. written) from scratch using a particular language. This approach has few, if any, restrictions on the range of problems which can be addressed and thereby offers extensive flexibility. However, writing software, and in particular simulation software, is not a simple task. Using general purpose programming languages to develop models not only means building the entities of the problem domain, for example machines, parts and trucks, but also includes the simulation based features. These features include the simulation executive, event list and time advance mechanism. The process of building a model in this way requires considerable skill and time (Pidd, 1992b). The model builder will need to understand the syntax of the particular language and be able to translate the often complex problem situation into a simulation model. Building models in this way typically involves a repeated cycle of program compilation, testing and debugging (Pidd, 1992b).

Simulation programming languages, for example SLAM II (O'Reilly, 1993), SIMAN (Kasales & Sturrock, 1991) and SIMSCRIPT (CACI, 1993), reduce the amount of programming required to build a simulation model. This is achieved because the standard simulation features are included within the language constructs. Specific, pre-coded routines provide features such as the simulation executive, the time advance mechanism, random number generators, results collection and report generators. Because these features do not have to be coded from scratch, the model development time is reduced. Furthermore, because the syntax of simulation languages is specifically designed for simulation applications, it proves to be more representative, or expressive, of the simulation domain. Although simulation languages ease the problem of model construction there remain drawbacks; the ease of use is provided at a cost. The model building process still requires someone with programming skills and, as Shewchuk & Chang (1991) suggest:

> "... [general-purpose simulation languages] *are difficult and time-consuming to learn and require a good degree of programming skill.*"

Data driven systems, or simulators, remove the need for formal programming: models are constructed using data and are able to run directly (Pidd, 1992b). Because the model logic and the simulation features are in-built (Carrie, 1988), there is no need to develop any program code. Models prove to be easier and quicker to construct, however this is at the cost of a further reduction in flexibility. Shewchuk & Chang (1991), in making the comparison with general purpose simulation languages, suggest of data driven simulators:

> *"... they are very easy to use but are restricted to a narrow range of applications and thus offer limited flexibility."*

However, because simulators prove easier to learn and use, they allow a modeller to concentrate on solving the problem rather than trying to understand the mechanics of the software (Banks et al., 1991). There are two types of simulator: the general purpose and the domain specific.

As might be expected by the name, a general purpose simulator is able to model a wide range of problems and thereby provides additional flexibility over domain specific simulators. The increased flexibility is achieved by utilising generalised concepts to which the modeller maps, or translates, the real world entities. The generalised concepts avoid the terminology of a particular application domain. For example, a 'resource' may be used to represent an aeroplane, a hospital bed or a machine depending upon the type of simulation model being constructed. Although data driven simulators require no formal programming, it is often the case that elements of the model logic need to be described using key words of the package. An example includes the flow of a part through different work centres. The emphasis should, however, be on constructing the models via data.

Manufacturing is perhaps the most dominant form of domain specific simulators. There are several packages available: for example ProModel (Harrell & Leavy, 1993), SIMFACTORY (CACI, 1993) and ATOMS (Bridge, 1990). WITNESS (Thompson, 1994), although a manufacturing simulator, has an in-built language which extends its applicability (O'Keefe & Haddock, 1991): it is more akin to a general purpose simulator. Manufacturing simulators employ concepts and terminology found in the manufacturing domain. These characteristics ease the process and time involved to build models. Users do not need to translate the problem characteristics into the concepts of the simulator.

In a simulator the simulation features are built-in whereas the features of the manufacturing domain are configured via their data attributes. Typical features would include parts, machines, trucks, routings and schedules (Banks et al., 1991). For example, the data associated with a part may include part number, part description and drawing number. ATOMS (Bridge, 1990) is able to automatically load a 'part file' which would then create all the part entities within the model; equally a 'part routing file' can be loaded which will direct each of the parts to the appropriate machines when manufactured. These files are text based and can be created via a text editor; alternatively they may be 'down loaded' from an organisation's own part or routing database. In most cases data can be loaded from a file or entered directly through the user interface. In this way models can be constructed without recourse to detailed programming.

There are, however, drawbacks to manufacturing simulators. It has already been mentioned that they provide limited application. There is a tendency, as manufacturing simulators become more focused, to further limit their applicability or flexibility. For example, RENSAM (O'Keefe & Haddock, 1991) is a simulator designed specifically for modelling flexible manufacturing systems (FMSs) whereas ATOMS, although a discrete part simulator, is unable to represent automated equipment such as FMSs and automated guided vehicles (AGVs).

The reliance on using data to configure models can also prove to be problematic: not only must the data be available but as O'Keefe & Haddock (1993) point out:

> *"Where data are readily available generating the input is easy 'if' the paradigm of the package is shared by the user, but may be difficult otherwise."*

In other words, the data may be available but may not be in the same format expected by the simulator. This places an unwelcome overhead on the model builder; the data has to be manipulated into the format recognised by the simulator. Consequently, users need to understand the limitations of the simulator so that they are able to translate the problem into the appropriate concepts. However, if simulation technology is to be more widely used the tools need to be easier to learn and use (Collins & Watson, 1993). Simulators are a significant step in this direction.

7.1.2 Making a choice

The different overall approaches to building simulation models have been outlined above. The question now becomes one of deciding which approach is most appropriate to satisfy the requirements identified in the previous chapter. Initial consideration will be given to the role of the user.

Although the use of simulation is a popular mode of analysis, it is not as widespread as perhaps it should be. A recent survey of the use of simulation in UK manufacturing (Simulation Study Group, 1991) found awareness of the technique to be low in comparison to other engineering tools such as MRPII and CAD; this has obvious effects on the level of demand. Collins & Watson (1993) blame the high level of effort required to successfully utilise simulation technology for the low levels of actual usage.

The study highlighted several mechanisms by which the awareness of simulation could be extended as well as identifying issues for research and development; included in these issues were ease of use, simulation methodology, data collection and speed of model building. These issues are important if simulation tools are to become accessible. Not only must they have sufficient capability to address real problems but be simple enough for a non-simulation expert to use: resolving both issues is not a straightforward task (Pidd, 1992c).

Both general purpose programming languages and simulation languages are demanding of the model builder; they require simulation model building skills, programming skills and time. It is accepted that as one moves from general purpose programming languages to simulation languages the extent of these requirements is reduced. However, if one considers who may, and should, be constructing the models then such requirements remain a significant obstacle.

Although simulation models are built in order to obtain a prediction of performance of a system, the generated results are not the sole benefits to be gained from using simulation. Ball et al. (1994) discuss the additional benefits to be gained from constructing a simulation model; the process of model building provides important insights into the system under analysis. Paul (1991) goes further by arguing that the emphasis of simulation modelling is more toward problem understanding than problem solving. The model building process highlights, and challenges, the assumptions made. Furthermore, the process provides an insight unlikely to be achieved from examination of simulation results alone.

Additional benefits from the application of simulation would result, therefore, if the user of the simulation model was responsible for the model building process (Ball et al., 1994). In the context of manufacturing systems analysis, manufacturing personnel would appear the obvious choice. They possess the detailed knowledge necessary to build simulation models; this knowledge would need to be elicited by the model builder. It would be quicker and more informative if the model builder could be drawn from the manufacturing personnel, especially since these people would be involved in the decision making. Ülgen & Thomasma (1990) concur:

> *"... simulation should be employed as much as possible by the manufacturing engineers who have the most intimate knowledge of the systems they want to simulate."*

The assumption that the user of the simulation 'technology' will not be a simulation expert emphasises the importance of characteristics such as ease of use and model build time. In this context manufacturing simulators are considered to be the most appropriate simulation technology to be used.

7.2 The provision of planning and control functionality within manufacturing simulators

7.2.1 Introduction

Since the adoption of an approach based on the principles of manufacturing simulators has been recommended, the provision of planning and control functionality by current simulators now needs to be investigated. This section will evaluate current manufacturing simulators from two perspectives: their provision of planning and control functionality and their ability to meet the requirements identified in the previous chapter.

There have been several, more general, reviews of manufacturing simulators. Bridge (1990) assessed manufacturing simulators from the position of supporting the manufacturing systems design process; both Law & Kelton (1991) and Banks (1994) include brief descriptions of the functionality offered and typical applications. More recently Ball (1994) has assessed manufacturing simulators from the perspective of ease of use and scope of application. This study will comment on the ability of current simulators to model manufacturing planning and control systems.

7.2.2 The emphasis of current simulator functionality

It is evident from the range of entities supported by current manufacturing simulators that the emphasis is towards shop floor operations: typical entities include machines, parts, trucks, conveyors. The representation of planning and control elements is not so evident.

Generally the functionality of manufacturing simulators extends to local production control through the inclusion of production schedules and work-to lists. Decision control rules, for example first-in-first-out (FIFO) or earliest due date, may also be included. Different simulators offer different levels of support for shop floor control. MAST (Lenz, 1985) contains six groups of decision rules, ranging from part scheduling and operation sequencing to transport selection and control. SIMFACTORY (CACI, 1993), on the other hand, provides only four operational algorithms as standard.

The emphasis towards local production control and shop floor operations should not be surprising. The review of the International Journal of Production Research (discussed in chapter 6) identified scheduling and dispatching as one of the most popular problem areas addressed by researchers. However, it is interesting to note that of the simulation approaches adopted by researchers, manufacturing simulators did not feature - the emphasis being on simulation languages and particularly SLAM II and SIMAN.

One possible reason for the apparent lack of interest in manufacturing simulators may well be the greater flexibility offered by the simulation languages; researchers are then less restricted by the limitations of manufacturing simulators when constructing shop floor models or to the extent of production control functionality offered. The shop floor models described do not appear, however, to be of the type to cause problems for any manufacturing simulator. It is important to repeat that the review only investigated the application of simulation to planning and control issues; the inclusion of manufacturing simulators is included in other research studies, in particular manufacturing systems design.

Although shop floor controls appear to dominate the available functionality, attempts have been made to extend matters: ATOMS (Bridge, 1990) includes alternative control systems as well as several scheduling algorithms. The control systems include material requirements planning (MRP), kanban and order point. The implementations of these systems are of a more representative nature, for instance MRP concentrates on the order release mechanism. Furthermore, the user is restricted entirely to the options offered; because the construction of the simulator is completely data driven, i.e. no programming required, alternative or modified systems cannot be accommodated.

MRP and kanban have also been included in manufacturing simulators by using 'push' and 'pull' routing constructs, for example WITNESS (Thompson, 1994) or ProModel (Bikos et al., 1994). CADOK (Mittal & Wang, 1992) is a simulator specifically developed to model kanban controlled environments. However, not only is this simulator restricted to 'pull' environments but also to applications typified by single card or production kanbans.

Although the inclusion of alternative planning and control systems is suggested, closer investigation reveals that the emphasis remains with shop floor implementations. The emphasis is only with the lowest level of the planning and control hierarchy: coverage of elements across the planning and control hierarchy is not effectively supported.

7.2.3 Levels of integration

It was emphasised in chapter 6 that, in the context of simulation modelling, integrating and interfacing are not the same, although they are often implicitly considered as such. Interfacing is more akin to linking systems together for the purpose of automatic data transfer, for example if a finite scheduling package is linked to a model of the shop floor, the scheduler merely provides inputs to the model. This is not the same as the situation where a scheduler is integrated within a model and thereby is involved in the time phased feedback mechanisms.

ATOMS (Bridge, 1990) does provide some form of integration of the planning and control activities with shop floor operations. The re-order point system, which is included, takes part in the simulation; orders are generated and released based on the current inventory status.

FACTOR (Lilegdon, 1992) recognises the relationships between the sources of data required to build and run scheduling-based simulation models. Data from the various production data sources (for example, a MRP system, an inventory system, shop floor status and current resource loading) can be utilised to construct a model: this is achieved through data transfer. Simulation models can then be configured in line with the current status of manufacturing operations. Schedules can be generated and the simulation run from this starting point. Alternatively the user can enter all the data required to generate a schedule manually, however, the data transfer mechanisms provided by FACTOR do speed up and simplify this process. Although the FACTOR system does actively support the generation and evaluation of production schedules, the generation of the schedule and the simulation modelling are discrete activities.

7.2.4 The role of 'people' entities

One of the factors which affects the levels of integration is the lack of involvement of 'people' entities in simulation models. MacCarthy & Liu (1993) observed that the potentially significant effects of *"human involvement"* in the scheduling function are overlooked. The inclusion of people entities supports the approach whereby the control system can be modelled rather than simply representing the control algorithm. People are essential components of the 'policies, procedures and practices' advocated by Maull et al. (1990).

Of course including entities such as operators and material planners along with machines, trucks and parts is not an easy modelling task. This is evident from the lack of provision of realistic 'operator type' functionality (Ball, 1994). Operators are often considered as a resource similar to a machine or a piece of equipment, for example FACTOR (Lilegdon, 1992) and ProModel (Harrell & Leavy, 1993). This type of representation minimises the significance of operators. Furthermore by using such an abstraction it would be difficult to accommodate operators taking a more pro-active role within simulation models. The securing of an operator and a machine by a part for the duration of its production does not reflect the reality of the situation. Operators take a more controlling role in activities.

7.2.5 The need to resort to programming

Although manufacturing simulators attempt to eliminate the requirement for any programming in order to build simulation models, it is not guaranteed. In an attempt to increase the flexibility of the simulator some form of programming is supported. This is particularly true when representing planning and control functionality. WITNESS (Thompson, 1994) provides a good example. As Law & Kelton (1991) explain, WITNESS becomes flexible due to its programming-like input/output rules and actions and its ability to include FORTRAN routines. The user must configure control blocks with parameters in order to provide complex decision rules. This is equivalent to programming each rule separately (Bridge, 1990). Conversely, XCELL+ (Conway & Maxwell, 1986) is unable to deal with complex decisions because it does not support programming-like statements (Law & Kelton, 1991).

Similarly, ProModel (Harrell & Leavy, 1993) increases its modelling flexibility due to programming-like constructs and its ability to call C or Pascal routines to model complex decision logic. Bikos et al. (1994) have constructed specific macros to represent MRP-based and JIT-based systems which can then be included in ProModel simulation models. These macros have been tailored specifically to suit the applications studied and would, therefore, be unsuitable for alternative MRP or JIT environments. FACTOR (Lilegdon, 1992) includes a number of the most commonly used scheduling rules as standard and supports an interface for installing user defined rules or logic.

The inclusion of routines written in general purpose languages (for example, FORTRAN, Pascal and C) also extends to models built using simulation languages.

Gardiner & Blackstone (1993), for example, developed a set of FORTRAN subroutines to interface with the simulation package and provide MRP calculations.

Recourse to some form of programming in order to represent planning and control functionality, even using manufacturing simulators, appears to be unavoidable with current systems. Only the most elementary of implementations are supported without having to utilise either the in-built programming-like constructs or by including pre-written routines or macros.

7.2.6 Focus of modelling attention

There are two conclusions which can be drawn from the current state of functionality offered by manufacturing simulators. The first is that the physical shop floor operation is the primary area of concern. The second is that the available planning and control functionality is only suitable for addressing operational issues and not the fundamental design of the complete system.

It is evident from the functionality provided that the focus of attention is primarily aimed at representing the physical shop floor, i.e. the machines, parts, trucks, etc.. The mechanisms for the release of orders provide the triggers for production. The assessment of shop floor designs is, therefore, feasible. However, the lack of attention to detailed production control misses the opportunity to properly assess the broader aspects of shop floor designs. Dale & Russell (1983), Pessler et al. (1983) and Lewis (1994) all describe the problems which can occur if the relationships between the production control system and the manufacturing facility are not properly evaluated.

The lack of coverage across the planning and control hierarchy as well as the limited range of functionality offered by current manufacturing simulators means that only a restricted range of control system issues can be addressed. In essence the application domain focuses on operational issues: for example the evaluation and optimisation of different production schedules. Moreover, the operational issues which can be reasonably addressed prove to be only a subset of those which can be tackled using simulation languages.

Without the programming-like constructs or the inclusion of programmed routines, many simulators would provide only limited support. When standard planning and control like functionality is provided and where no programming is permitted, for example SIMFACTORY (CACI, 1993) and ATOMS (Bridge, 1990), the modeller is clearly restricted to that functionality. This lack of flexibility means that alternative representations or new designs cannot be considered.

The FACTOR system (Lilegdon, 1992) has been specifically designed to meet the needs of manufacturing production planning. It has been designed to support the effective management of the capacity of a manufacturing organisation. The application of FACTOR is aimed at the operational issues of manufacturing; it is used to plan and schedule operations on a day to day basis. The inclusion of simulation means that the effects of a scheduling decision can be predicted, thereby supporting one of the objectives: to develop feasible and achievable production schedules. Again the emphasis is toward operations, extension of FACTOR to address control system design issues would not be practicable.

It is reasonable to conclude that the planning and control functionality offered by current manufacturing simulators is weighted towards operational investigations and analysis; the functionality does not support the evaluation of planning and control system designs.

7.3 Where next?

In order to adequately model manufacturing planning and control systems and be able to assess not only operational issues but design issues, a novel and comprehensive set of simulation requirements have been identified. Current manufacturing simulators are not able to fulfil these requirements.

The data driven characteristic of manufacturing simulators, i.e. the avoidance of any programming, is important. In the context of providing control system functionality it does, however, prove to be the undoing of many of the current batch of simulators. In order to include, or extend, the control functionality some form of programming is necessary. Where the simulators are purely data driven, the modeller is restricted to the limited functionality offered.

The scope of modelling opportunities is limited. This is most clearly demonstrated by the range of functionality. It is difficult to conceive that current functionality could support applications other than those of shop scheduling. The inclusion of a complete planning and control system or even hybrid system is not feasible. Integrating the policies, practices and procedures which themselves underpin most control systems would not be possible. The flexibility requirement, essential to support a simulator which can provide sufficient scope of modelling, is not feasible. When considering planning and control applications, current manufacturing simulators concentrate on limited operational issues; they are not equipped to address design issues.

In order to rectify the situation there are two alternatives: either construct a completely new manufacturing simulator or expand the functionality of a current manufacturing simulator. The first alternative is not considered to be practical. The discussion earlier in this chapter has indicated that a number of manufacturing simulators are already available, either commercially or as research tools. Furthermore the development of a new simulator would incur significant overheads: the simulation oriented features and an adequate set of shop floor functionality would have to be provided in addition to the necessary manufacturing control functionality.

The second alternative provides the opportunity to build on the expertise already invested in current manufacturing simulators. In other words, utilise the shop floor functionality and simulation features of an existing simulator and integrate the appropriate planning and control functionality within it. It is crucial, however, that the underlying functionality offered by the 'host' manufacturing simulator is both acceptable and suitable. The development of the control functionality can then be the primary focus of attention. This approach raises two fundamental questions. Which of the manufacturing simulators currently available is suitable? The simulator must provide adequate 'standard' functionality as well as support the integration of, what will be novel, planning and control functionality. The second question is one of how the planning and control functionality will be developed. Both of these questions will be addressed in the following chapter.

7.4 Summary

This chapter has reviewed the different approaches to the construction of simulation models. When one considers attributes such as ease of use, manufacturing simulators offer the most promising alternative. A review of the planning and control functionality offered by current manufacturing simulators has identified both limited and restricted provision. The emphasis is toward shop floor operations; the investigation of the design of complete planning and control systems is not feasible. The suggested way forward is to extend the functionality of an existing, but suitable, manufacturing simulator.

8. The application of object-oriented techniques

The previous chapter raised two important questions: how will the planning and control functionality be provided and which of the current manufacturing simulators can support the integration of this functionality once developed? Both questions will now be addressed. The development of computer software, and in particular simulation software, is not a simple exercise; typically systems are both large and complicated. The object-oriented paradigm is recognised as one which supports the development of such systems. It will be demonstrated that through the adoption of object-oriented techniques, planning and control functionality can not only be provided but, more importantly, meet the requirements placed upon it. The developed functionality will form the basis of an object library and extend the functionality offered by an existing object-oriented manufacturing simulator, namely the Advanced Factory Simulator (AFS).

8.1 Object-oriented techniques - key concepts

8.1.1 Introduction

Rumbaugh et al. (1991) provide an apt starting point for any discussion about object-oriented techniques:

> *"... object-oriented technology is more than just a way of programming. Most importantly, it is a way of thinking abstractly about a problem using real-world concepts, rather than computer concepts."*

Although the development of computer software underpins object-oriented concepts, it is important to note that there is more to 'object-oriented' than object-oriented programming. The object-oriented paradigm (concepts, principles, methods or techniques) includes object-oriented analysis, object-oriented design and object-oriented programming (Booch, 1994; Graham, 1991).

The problems associated with the development and maintenance of complex software, described in Brooks (1979), led to the development of structured programming techniques (Booch, 1994; Taylor, 1990). The essence of structured programming is a top-down, 'functional decomposition' of the problem domain. A program is systematically broken down into separate components, each of these components is itself decomposed until elementary procedures or functions are reached. This more disciplined and consistent style of programming has led to significant improvements in software development (Taylor, 1990). However, this approach essentially separated the data and functional attributes of a system, consequently problems of software modification and extension have remained.

Object-oriented methods take a different view of the problem domain. The fundamental construct is the 'object'; it encapsulates both data structure and behaviour in a single entity. In essence, object-oriented methods support, and facilitate, a completely different way of understanding, analysing and developing computer-based solutions; the key benefits include reuse and extensibility (Graham, 1991).

The following sections will introduce the key concepts of object-oriented techniques and identify the relationships between object-oriented methods, manufacturing and simulation. More detailed explanations of the key concepts can be found in, for example, Taylor (1990), Rumbaugh et al. (1991), Graham (1991) and Booch (1994).

8.1.2 Objects and classes

Simply by viewing the world around us it is possible to identify different 'objects': for example John Brown, a Volkswagen Golf and a Toshiba T1800 are all objects. John Brown is an object of type person, Volkswagen Golf is an object of type car and Toshiba T1800 is an object of type computer.

The 'object' is the fundamental construct of the object-oriented model and represents a completely different approach to software construction. Essentially an object is a software package which includes both the data structure and behaviour of the real world entity it represents (Yourdon, 1990). This contrasts with the more conventional approaches to developing software where the data and behaviour are only loosely connected (Rumbaugh et al., 1991).

By way of illustration, Figure 8.1 lists four examples of different types of car - these are all objects of type car. In this particular representation the common data attributes are 'Make', 'Registration Number' and 'Colour'. It is possible to represent the behaviour of a car in terms of 'Start', 'Stop', 'Move Forward' and 'Reverse'. Figure 8.2 generalises this definition of a car and provides a template for all future car objects; another car would be defined in terms of those data attributes and behaviour listed. The template is considered to be an 'object class', or 'class'.

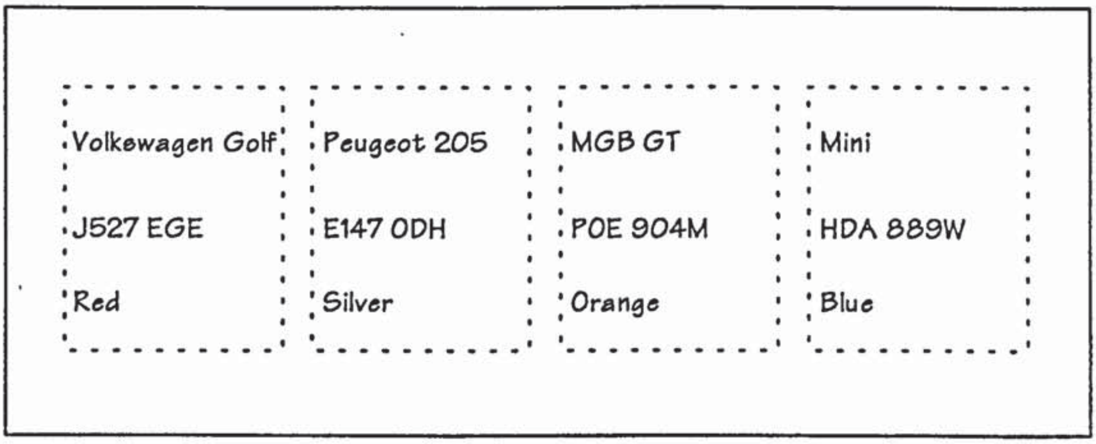


Figure 8.1: Example objects of type 'car'.

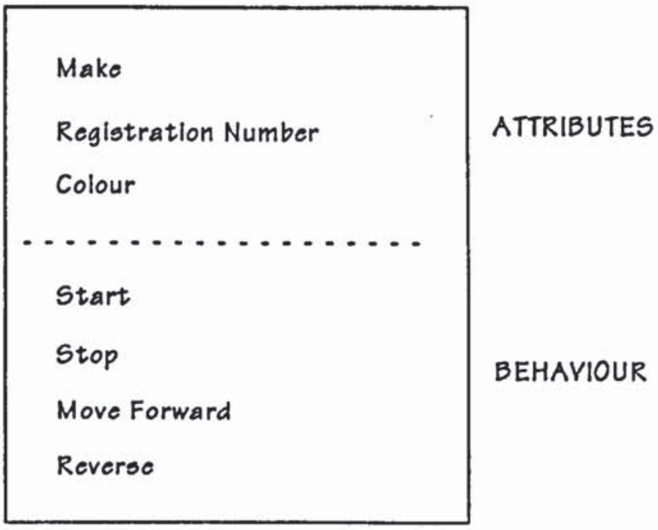


Figure 8.2: A class template.

A class is a template which defines objects with similar data properties and common behaviour. The class is the generalised form and objects are actual 'instances' of the class. When an object is created from a class, values are given to the attributes which define the particular object. In the above example, the objects Volkswagen Golf, Mini, etc. are all instances of the class 'car'. There are other examples of objects and classes: John Brown is an object (i.e. instance) of the class 'person'; Toshiba T1800 is an object of class 'computer'.

In software terms the behaviour is defined in terms of methods (or a set of procedures). In the object-oriented model, methods have an important responsibility: the data within an object can be accessed or modified only by the object's methods. This means that, for example, to change the colour of a car the class would require a method 'Change Colour'.

8.1.3 Encapsulation and information hiding

Encapsulation and information hiding are two closely related terms and support the concept of an object. 'Encapsulation' is the packaging together of both the data and methods which represent an entity; everything one needs to know about an object (its data and workings) is encapsulated together. 'Information hiding' is the technique of making the internal details of an object inaccessible to other objects in the system, except via the interface of the object. It is through this mechanism that the state of an object is protected from outside interference and controls how the object is modified. The (private) data attributes of an object, for example, can only be altered by those methods which are visible (i.e. public) to the other objects. These methods constitute an object's interface to other (different) objects. In the functional approach to software development, data may be accessed from several sources. This makes modification and extension difficult.

Cook & Daniels (1993) provide an interesting analogy of how different objects view one another; they call it the 'boiled egg' model, see Figure 8.3.

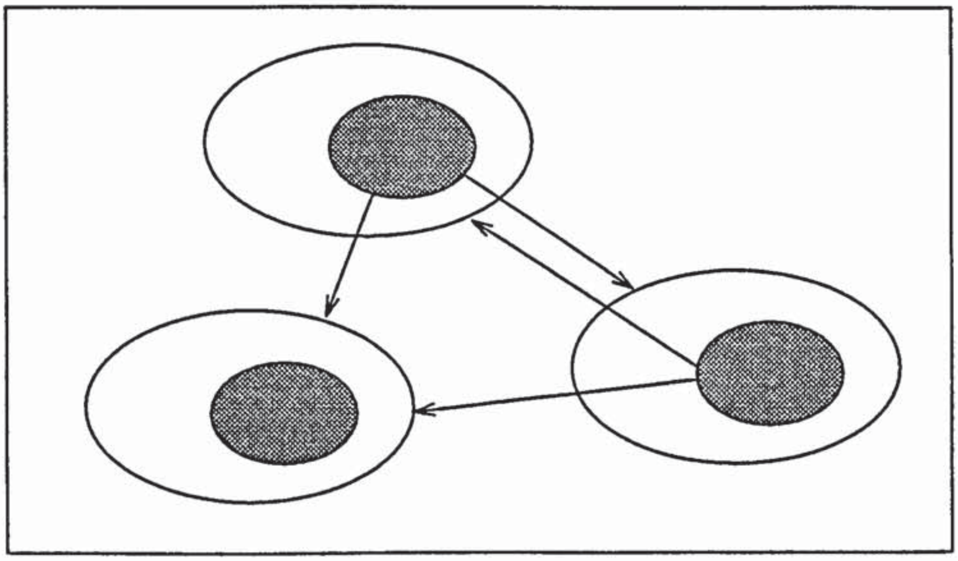


Figure 8.3: Encapsulation - the 'boiled egg' model (after Cook & Daniels, 1993).

An object is represented by a boiled egg: the yolk represents the object's data and the white which surrounds the yolk represents the methods, or operations, of the object. The shell is the only visible part of the object and represents the interface offered by the object to the world. Access to the inside of the egg by a different object is only possible via the interface, i.e. the shell. Although access could be gained by breaking the egg, this would be against the principles of encapsulation and information hiding.

8.1.4 Messaging

Messaging, or message passing between objects, is a direct result of encapsulation. Objects interact, or communicate, with other objects via messages. A message is a request from one object to another to carry out one of its operations (or methods). In essence, the send and receive capabilities of an object represent its interface with the outside world, i.e. the shell in terms of the boiled egg model. Data or methods cannot be accessed or actioned directly.

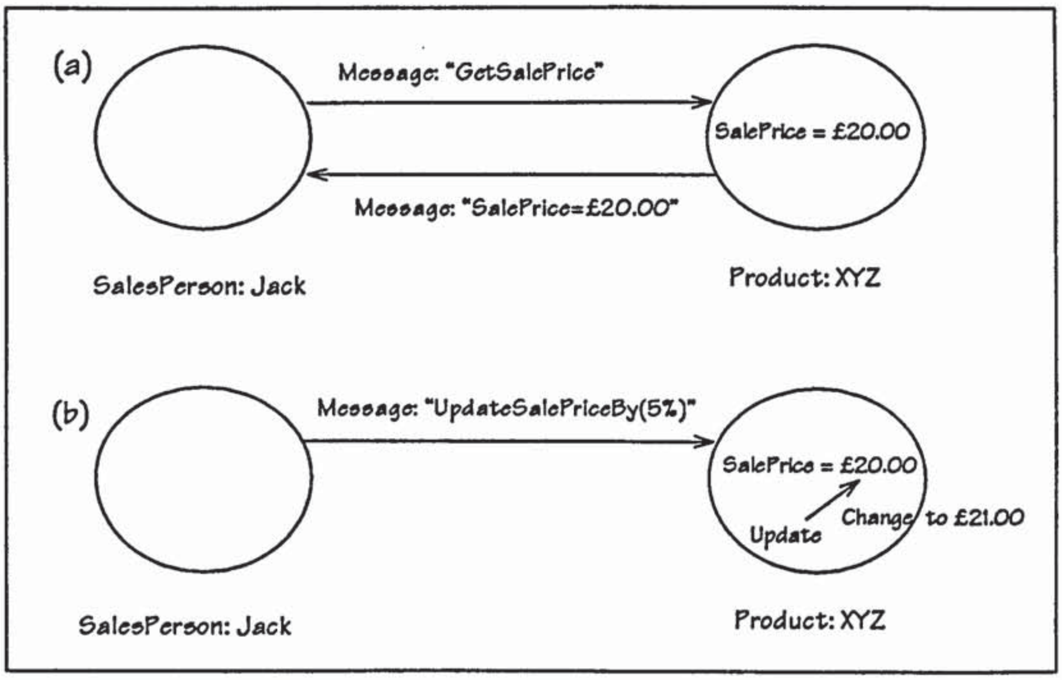


Figure 8.4: Examples of object messaging.

Messages are required either to request information from an object or to change the current state of an object. For example, a *SalesPerson* object may message a particular *Product* object for its current 'sale price', this being one of the attributes of the object *Product*. If the *Product* object can receive this message, i.e. understand it, then it will reply accordingly, otherwise the message will be ignored. In order to reply one of the objects methods will retrieve the current value assigned to 'sale price' and reply, via a message, to the sender object. This process is illustrated in Figure 8.4(a).

Alternatively, the *SalesPerson* object may send a message to the *Product* object to 'raise its sale price by 5%'. In this case a method would exist in *Product* to update the attribute 'sale price'. Figure 8.4(b) illustrates this example.

8.1.5 Polymorphism

Polymorphism is the ability to hide different implementations behind a common interface and thereby simplify the messaging process among objects (Taylor, 1990). In essence, polymorphism is the ability of different objects to respond to the same message, the response being unique to the behaviour of the particular object.

For example, in a graphics package three different shape classes could be *Rectangle*, *Circle* and *Line*, *Rectangle1*, *Circle1* and *Line1* being their respective objects. The same message "draw" can be sent to each of the objects, their response would be appropriate to their particular implementation. Figure 8.5 illustrates the concept of polymorphism.

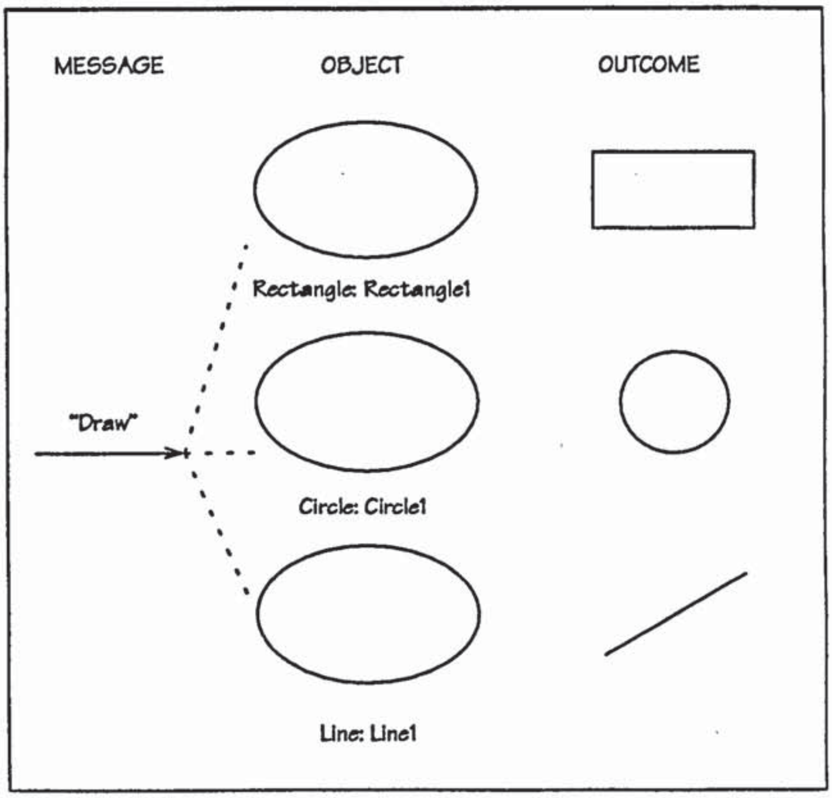


Figure 8.5: An example of polymorphism.

The interface is common but the implementation is specific. Compare the situation if polymorphism were not possible: the sender object would be required to send different messages appropriate to each of the different shape objects, for example "draw rectangle", "draw circle" and "draw line". The sender would need to identify the object with which it is communicating and select the message accordingly.

Polymorphism and information hiding allow objects to be designed which are interchangeable or 'plug-compatible' (Adiga, 1993). It becomes possible to add, or plug in, new classes (and thereby objects) to a system if they respond to the same messages as the present ones. For example, a new class called *Polygon* could be introduced to the graphics system discussed previously. Provided that the *Polygon* class has an implementation of the method 'Draw', then any object, for example *Polygon1*, would be able to respond to the message "draw". The sender object does not have to know anything about the new class or make any assumption on the implementation of the method.

Similarly, providing the class interface remains consistent, old versions of a particular class can be replaced by an updated version without affecting other classes in the system. For example, if a new version of the *Circle* class was developed, perhaps incorporating a more efficient drawing routine (i.e. method), then this could be substituted for the old version. Existing classes could still send the same message, i.e. "draw", and the new version would respond accordingly. To reiterate, the interface remains common whereas the implementation is specific.

8.1.6 Class hierarchy, inheritance and reuse

Classes can be organised into a class hierarchy which results in a family tree like structure. Figure 8.6 suggests a class hierarchy classifying the different forms of vehicle. Car, lorry and bus are all types of 'vehicle'; similarly saloon, estate and sports are all types of 'car'. In an object-oriented class hierarchy all of the data attributes and methods would be organised according to their commonality. All of the data and methods which are common to saloon class, estate class and sports class would be grouped in the class car; the data and methods in each of the three lowest classes (saloon, estate and sports) would be different, and thereby differentiate the three classes. In a similar fashion, the class vehicle would include the data and methods common to all vehicles, in this case car, lorry and bus. For example, the data attributes of the vehicle class could include 'registration number', 'seating capacity' and 'fuel capacity'.

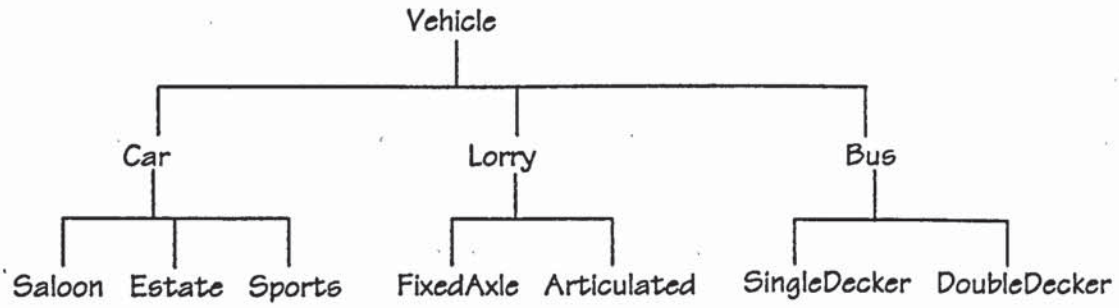


Figure 8.6: An example class hierarchy.

The class hierarchy provides an efficient mechanism in which the data variables and methods of a class can be used by any of its sub-classes. The data and methods of the *Vehicle* class, for example, are accessible to the three sub-classes *Car*, *Lorry* and *Bus*. The terminology often used is that *Vehicle* is the parent (or super) class to the child (or sub) classes *Car*, *Lorry* and *Bus*. *Lorry* is the parent class of the classes *Fixed-Axle* and *Articulated*. The data and methods of *Lorry* are accessible by the two sub-classes, however they are not accessible by the three sub-classes of *Car*. The only data and methods available to these classes is contained in the *Car* and *Vehicle* classes respectively.

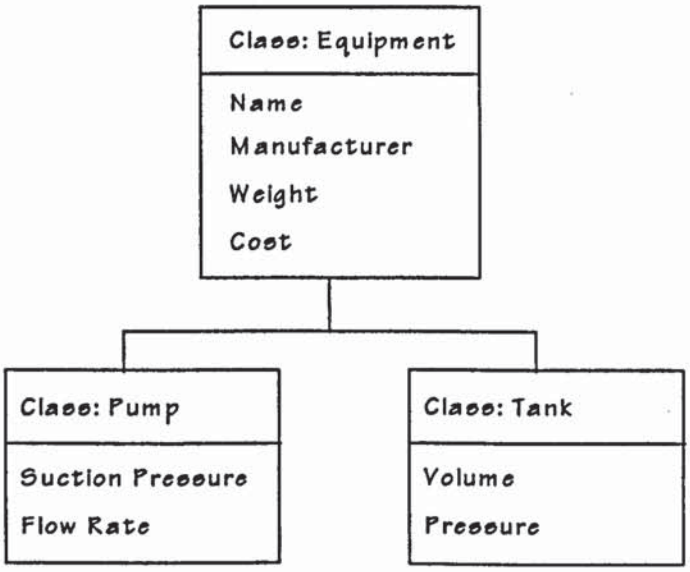


Figure 8.7: A class hierarchy for the classes Pump and Tank (after Rumbaugh et al., 1991).

The availability or accessibility of data and methods from a parent class to a child class is made possible through the 'inheritance' mechanism. In the same way that children can inherit attributes from their parents, a child class can inherit both the data and methods from its parent class; this extends all the way up the particular class hierarchy. Figure 8.7 illustrates a class hierarchy for the classes *Pump* and *Tank*. The inheritance mechanism means that objects of class *Pump* will not only have their own data and methods but will also inherit those of their parent class *Equipment*; Figure 8.8 illustrates two objects. For clarity, the methods have been omitted from both Figure 8.7 and Figure 8.8.

object: Pump101	object: Tank03
CompanyXYZ	ABC-Ltd
200 (kg)	1500 (kg)
300 (£)	1000 (£)
1.1 (atm)	40000 (l)
250 (l/hr)	2.0 (atm)

Figure 8.8: Example objects of class Pump and Tank.

The abstraction of classes to produce a class hierarchy and the inheritance property together support the concept of 'reuse' and thereby economise on the amount of programmed code which is required. The code generated to represent the parent class is written only for that class; subsequent child classes have access to that code through the inheritance mechanism. The only code which needs to be generated is that which represents the child class and differentiates it from both its parent and other child classes.

8.1.7 Object libraries

The concept of an object or class library is particularly relevant to this work. An object library is a coherent collection of complete, tested and reusable objects (Graham, 1991). The objects in a library typically represent a particular application, for example a shop floor library would include different types of machine classes, truck, part and operator classes. Similarly a simulation based library would include those classes supporting the construction of simulation models, for example, event list and time advance mechanism.

The availability of a suitable object library is essential to support object-oriented development. It is through the existence of classes in an object library that the benefits of reuse are made possible. A software developer can use the classes which have already been developed rather than 're-invent' the code. Similarly a user of a simulation object library is able to construct simulation models from the existing classes. The range of functionality offered by an object library is crucial if the concept of reuse is to be properly exploited.

8.2 The benefits of object-oriented techniques

Graham (1991) outlines the benefits of object-oriented techniques through the summation of reusability, extensibility and semantic richness; this is a view shared by Mize et al. (1992). The 'bottom-up' development approach and information hiding maximise the potential for reuse; this is further enhanced through abstraction of classes into a class hierarchy and the inheritance mechanism. The more natural mapping of real world entities to their object counterparts captures the meaning of the application, i.e. its semantics.

Taylor (1990) suggests that the use of object-oriented techniques to develop software offers many of the benefits similar to those sought by manufacturing: improved productivity, higher quality, easier maintenance and reduced development costs. The availability of classes in the form of class libraries will mean that development effort and time does not need to be repeated. In addition, reusing classes from libraries means that there is less code to develop and thereby less scope for the introduction of errors into the software. Furthermore, these classes will have themselves already been developed and tested. If the classes map well to their real world counterparts then identifying where problems are occurring should be eased and be reflected in the maintenance of classes. The encapsulation and information hiding properties should lead to changes being confined to only a small proportion of the software.

It cannot be claimed that the object-oriented paradigm is a panacea; there are limitations which are rightly recognised: for example in Ince (1993), Graham (1991) and Taylor (1990). Clearly the benefits of reuse rely heavily on the availability of suitable object libraries. The development of an object library, as with most forms of software development, is a considerable task. Moreover, it is not until the number of classes in the library reaches some form of *"critical mass"* (Najmi & Stein, 1989) that the benefits can be properly obtained.

It is important to recognise that the adoption of the object-oriented model will not, by itself, guarantee the benefits outlined; strict software development and programming discipline are also required. It is the responsibility of the software developer to enforce issues such as encapsulation, information hiding and messaging; they do not appear implicitly.

The identification of the classes and defining the class hierarchies which represent the problem domain is not straightforward (Shewchuk & Chang, 1991). Ting (1991) suggests that because of the lack of precise algorithms or systems to follow, identifying objects can be a *"creative process"*. Appropriate object-oriented analysis and design (see for example Booch (1994) or Rumbaugh et al. (1991)) are, therefore, important factors in this respect. Both Shewchuk & Chang (1991) and Ball (1994) noted that the development of classes was an iterative process: classes were developed and redefined as the project developed, some classes became redundant as new classes were identified.

In summary, although object-orientation is no panacea it has much to offer software development. Many of the expected benefits are gradually being realised by industry (KPMG Management Consultants, 1994). Moreover, as the following section will demonstrate, object-oriented methods are particularly relevant and applicable to the areas of manufacturing, simulation and manufacturing simulation.

8.2.1 Manufacturing and object-oriented methods

The manufacturing domain proves to be an important area for applying object-oriented methods. In fact Graham (1991) considers manufacturing to be the most important area of application after graphical user interface (GUI) development and the growing interest is clearly evident (Consultants' Conspectus, 1995; KPMG Management Consultants, 1994). An entire issue of the International Journal of Computer Integrated Manufacturing (Rogers, 1994) was dedicated to the application of object-oriented concepts to the design and control of integrated manufacturing systems. In 1992, the Division of Manufacturing Engineering at the University of Calgary hosted the International Conference on Object-Oriented Manufacturing Systems (ICOOMS) (Norrie, 1992). This conference demonstrated interest from both academia and industry.

Glassey & Adiga (1989) make the point that:

> *"There is a natural one-to-one correspondence between physical objects in a factory and instances of software objects that represent them."*

Nof (1994) provides a critique of the potential of object-oriented methods in manufacturing. He suggests that the scope of the methods in manufacturing extends to the development of manufacturing software, models for manufacturing planning and design as well as models for manufacturing control. The study observed several areas of similarity between manufacturing and object-orientation, some of which agree with those of Taylor (1990) mentioned previously. Nof (1994), however, also identifies some limitations of object-oriented methods, for example the need for further development in object libraries. Nevertheless, the overriding conclusion is that they have a great deal to offer to the manufacturing domain.

The interest and potential of object-oriented methods can be assumed from the range of applications being considered. For example, object-oriented techniques have been applied to a variety of manufacturing applications: in manufacturing system specification (Joannis & Krieger, 1992), process planning (Yut & Chang, 1994), material flow design (Rembold & Tanchoco, 1994), computer integrated manufacture (CIM) (McFadden, 1989), flexible manufacturing systems (FMSs) (O'Keefe & Haddock, 1991), information systems design (Wang & Fulton, 1994) and in capacity planning and scheduling (Ulrich & Dürig, 1992).

8.2.2 Manufacturing simulation and object-oriented methods

A significant amount of attention has been paid by the simulation community to the role of object-oriented techniques: see for example Pidd (1995), Bischak & Roberts (1991) and Rothenberg (1986). This is not surprising, the origins of object-oriented techniques can be traced to simulation and to the development of SIMULA (Dahl & Nygaard, 1966), a discrete-event simulation language. One particular focus of attention is that of manufacturing simulation: see for example King & Fisher (1986), Najmi & Stein (1989), Basnet et al. (1990), Shewchuk & Chang (1991) and Ball (1994).

As with manufacturing, object-oriented techniques have a great deal of similarity with simulation. In fact Pidd (1995) suggests that many of the features of object-orientation may already be used by simulation modellers without their explicit knowledge. The features of encapsulation, polymorphism and the class mechanisms being particularly relevant to simulation, a point also noted by Bischak & Roberts (1991).

The advantages of reusable objects when developing simulation models are aptly summarised by Najmi & Stein (1989):

> *"Just as one builds experimental apparatus in chemistry from flasks, tubes, and beakers etc., one builds a simulation experiment from software objects ... as with chemistry experiments, one can dismantle the apparatus for reuse in making the apparatus for other experiments. In the conventional approach, on the other hand, the simulation developer needs to play the role analogous to that of a glass-blower in having to build new components for each experimental mode."*

There have been recent developments in simulation software and, in particular, manufacturing simulation, which have exploited the concepts of object-orientation to varying degrees. Examples include ModSim (Herring, 1990) - an object-oriented simulation language, ARENA (Collins & Watson, 1993) - an object based simulation system, ProModel (Harrell & Leavy, 1993) - an object-oriented manufacturing simulator and BLOCS/M (Glassey & Adiga, 1989) - an object library supporting the simulation of manufacturing systems.

8.3 Object-oriented techniques and manufacturing planning and control

The application of object-oriented techniques to manufacturing planning and control issues has received some attention by researchers: examples include Alasuvanto et al. (1988), Chang et al. (1991), Ulrich & Dürig (1992), Bhuskute et al. (1992), and Persentili & Alptekin (1993).

Of the two applications discussed by Alasuvanto et al. (1988) only one is relevant to the area of production planning; it addresses the development of a decision support tool for the planning and scheduling of dairy production. The application of object-oriented techniques concentrates on the graphical user interface and no discussion of the object classes is included. Ulrich & Dürig (1992) also apply object-orientation to the development of a decision support tool, namely a capacity planning and scheduling tool.

This work concentrates on the development of an operational tool and does not concern itself with either design or simulation modelling.

Persentili & Alptekin (1993) seek to evaluate the impact of planning decisions made by computer-aided production management (CAPM) systems on the manufacturing facility. Again this work adopts more of an operational perspective. The CAPM system discussed is a material requirements planning system (MRP) and includes a master production scheduling (MPS) object, a MRP object and a capacity requirements planning (CRP) object. These objects, which are particularly high level objects, are integrated with a manufacturing cell object which simulates the shop floor operations. Although the authors talk of identifying further sub-classes, it is only the manufacturing cell which is illustrated. The discussed implementation suggests that the material planning system is interfaced but not integrated with the shop floor object.

Chang et al. (1991) detail their work which examines the application of object-oriented principles to the design of *"the manufacturing planning and control system."* However, the focus of this study, and what appears to be future work, is towards shop floor operations: the scheduling of jobs to machines and the allocation of tooling and material handling devices. The work, which incorporates both object-oriented methods and artificial intelligence (AI) techniques, does not extend to other levels of the planning hierarchy.

Bhuskute et al. (1992) emphasise the need to be able to develop simulation models by utilising existing models, rather than *"single purpose, throw away efforts"* (Mize et al., 1992). Their work seeks to exploit the reusability property offered by object-oriented methods. An explanation of the different classes is included in the discussion, they are grouped according to three categories: physical elements, information elements and control/decision elements.

It is interesting to note that one of the information classes, *CustomerOrder*, not only represents an actual customer order, in terms of part number and order quantity, but also triggers the planning logic. The order is exploded, using the bill of materials information and the inventory status for the part, and determines the requirements for lower level components. This logic is embedded in one of the simulation classes. The embedding of functionality in this way will not only prove difficult to modify but does not reflect the reality of the situation. Rather than mixing simulation-based functionality with material planning functionality, it would be more appropriate to encapsulate the planning logic into a separate 'planning' class.

In a related paper, Mize et al. (1992) identify the influence of control policies on manufacturing performance and recommend the inclusion of decision making entities in the simulation environment. However, it is apparent from the classes, which are described in Bhuskute et al. (1992), that operator type objects are not included, although in reality they would be included in various decision making activities.

In summary, the research to-date addresses only limited applications of object-oriented methods to the planning and control domain, concentrating on shop floor operations rather than control system design issues. Coverage of the complete planning and control hierarchy is not considered. This thesis seeks to fully exploit the properties of object-orientation and develop a planning and control object library. This library will support the evaluation of planning and control system designs as well as address operational issues. The classes of the library will integrate fully with classes representing the shop floor functionality to provide a more comprehensive model of a manufacturing system. The required flexibility will be provided through the different classes.

The objects which will populate the library correspond to the building blocks described in the previous chapter, an analogy also used by Rolstadås (1988b) and Mize et al. (1992). Cox & Novobilski (1991) utilise the 'software-IC' analogy; emphasising the similarity with the integrated silicon chip and its associated reusability.

8.4 The Advanced Factory Simulator

The previous section has identified that through the application of object-oriented techniques a planning and control object library can be developed. This library will extend the functionality of an existing manufacturing simulator. It has already been mentioned that rather than develop a completely new manufacturing simulator, the expertise already invested in such development should be exploited. Effort can then be focused on the development of the planning and control functionality. The problem becomes one of identifying a manufacturing simulator which will support the integration of the developed planning and control object library. The Advanced Factory Simulator (Ball, 1994) satisfies the requirements and will be discussed here.

The Advanced Factory Simulator (AFS) is a novel manufacturing simulator, developed at Aston University, which utilises object-oriented principles. AFS provides the necessary simulation features as well as a representative range of manufacturing elements: for example machines, parts, trucks and operators. Because AFS is extensible it supports the integration of new functionality. A complete discussion of AFS can be found in Ball (1994) and Ball & Love (1994), only a summary will be included here.

Ball (1994) assessed manufacturing simulators with respect to their ease of use and scope of application. His study considered these issues from the point of view of the potential user, namely a manufacturing engineer and not a simulation expert. The study concluded that whilst current manufacturing simulators offer ease of use they have limited functionality. The limitations, it is suggested, may preclude the use of the simulators from varied applications. The subsequent research set out to design and develop a manufacturing simulator which would not only retain the important characteristic of being easy to use but would also be extensible. In other words, the simulator architecture should support the inclusion of new functionality and thereby remove the potential for restricted application.

The software architecture of AFS exploits object-oriented principles and has been developed in order to support extension. This is not to claim that AFS is the only application of object-oriented principles to manufacturing simulation, other examples include: Adiga & Glassey (1991), Shewchuk & Chang (1991), Basnet et al. (1990) and King & Fisher (1986). However, the study by Ball identified a number of weaknesses: among them, the need for programming in order to construct models from an object library, a lack of inclusion of operator objects and a failure to demonstrate the inclusion of new functionality. The development of AFS has sought to address the limitations and omissions identified in current approaches. (It is important to note that these observations parallel the author's own observations when considering the application of manufacturing simulators to planning and control problems.)

AFS does not require any programming in order to build, or run, simulation models and can be rightly considered as a data driven simulator. Models are configured either from the user interface or through the application of text files. The functionality in AFS matches the concepts and terminology found in the manufacturing domain and so reduces the abstractions required by the modeller. This has been made possible because AFS has been developed from the point of view of the ultimate user, namely someone from the field of manufacturing and not a simulation expert.

The current functionality of AFS is primarily shop floor oriented and is not comprehensive. However, because the architecture is independent of functionality there is potentially no restriction to the type of functionality which can be added. This means that new or modified functionality can be added, for example a new type of machine or an alternative form of transport device. More appropriate to this discussion is the potential to add planning and control functionality in the form of a planning and control object library.

A user may not possess the necessary skills to include new or modified functionality. Consequently, a second potential user of the AFS architecture is a developer. New or modified functionality may be added to AFS either as a matter of continued development or at the request of a user (i.e. modeller). The use of object-oriented principles both in the design and construction of AFS supports this form of extension. Furthermore, the more natural relationships between the real world entities and their software counterparts enhance this property. Developers will be people with appropriate skills in object-oriented software development. The design of the architecture means that they should be able to extend (or modify) the current functionality without too much difficulty.

There are additional reasons which have also contributed to the choice of AFS as a suitable development system. The availability of the source code and the close proximity of the author of AFS have obvious benefits should there be problems with the software.

It should be noted that the development of AFS and the development of the planning and control object library are separated by approximately 12 months. Furthermore, AFS is still under a limited form of development. However, it is considered that AFS offers the potential to support the inclusion of new functionality which would not be possible from alternative manufacturing simulators.

8.5 Summary

This chapter has introduced the principles of object-orientation; the application of which offer substantial improvements in the development of complex software systems. The manufacturing environment has proved to be a significant application domain of object-oriented methods; this has extended to manufacturing simulation. The previous application of object-oriented simulation to planning and control problems has been limited; there is considerable scope to develop an object library supporting manufacturing planning and control functionality. The Advanced Factory Simulator (AFS) has been identified as a manufacturing simulator which will support the inclusion of new and different functionality.

9. The design of a planning and control object library

This chapter will describe the design of a novel planning and control object library which will support modelling and evaluation of manufacturing planning and control systems. The object library is called WBS/Control. The development of the system, which has followed object-oriented principles, extends the functionality of an existing object-oriented manufacturing simulator, namely the Advanced Factory Simulator (AFS). The chapter begins with an overview of the different types of classes currently available in the library, and how these classes would interact within a typical simulation model. A more detailed discussion of each of the class types then follows.

9.1 WBS/Control

9.1.1 Introduction

WBS/Control is an object-oriented class library providing manufacturing planning and control functionality for use in simulation models. The name WBS/Control is derived from the origins of this research: the development of planning and control functionality forms part of a wider research agenda, namely the development of a business-wide, or enterprise, simulator which has been termed the 'Whole Business Simulator' (WBS). This will be discussed subsequently in chapter 12.

It is intended that WBS/Control will be used to model different planning and control systems and thereby support the evaluation of such systems. The structure of the object library is such that both design and operational issues associated with manufacturing control systems can be readily investigated. WBS/Control extends the functionality of an existing object-oriented manufacturing simulator, namely the Advanced Factory Simulator (AFS). The inclusion of planning and control functionality within a manufacturing simulator provides a more comprehensive simulation environment and reflects the activities of actual manufacturing systems; the organisation of the production is managed by a planning and control system. Only through such modelling capabilities can the important emergent properties between the manufacturing facility and its control system be understood.

9.1.2 An illustration of the modelling potential of WBS/Control

Before embarking on a detailed discussion of the design of the classes which will populate the class library WBS/Control, an overview will be presented. This will introduce the scope of classes currently available in WBS/Control and the range of simulation activity possible; in order to do this a model of a typical manufacturing situation will be discussed. Figure 9.1 illustrates the scenario; the model, which has been configured from the functionality offered by AFS and WBS/Control, is actually made up of a collection of the respective objects, i.e. instances of the classes.

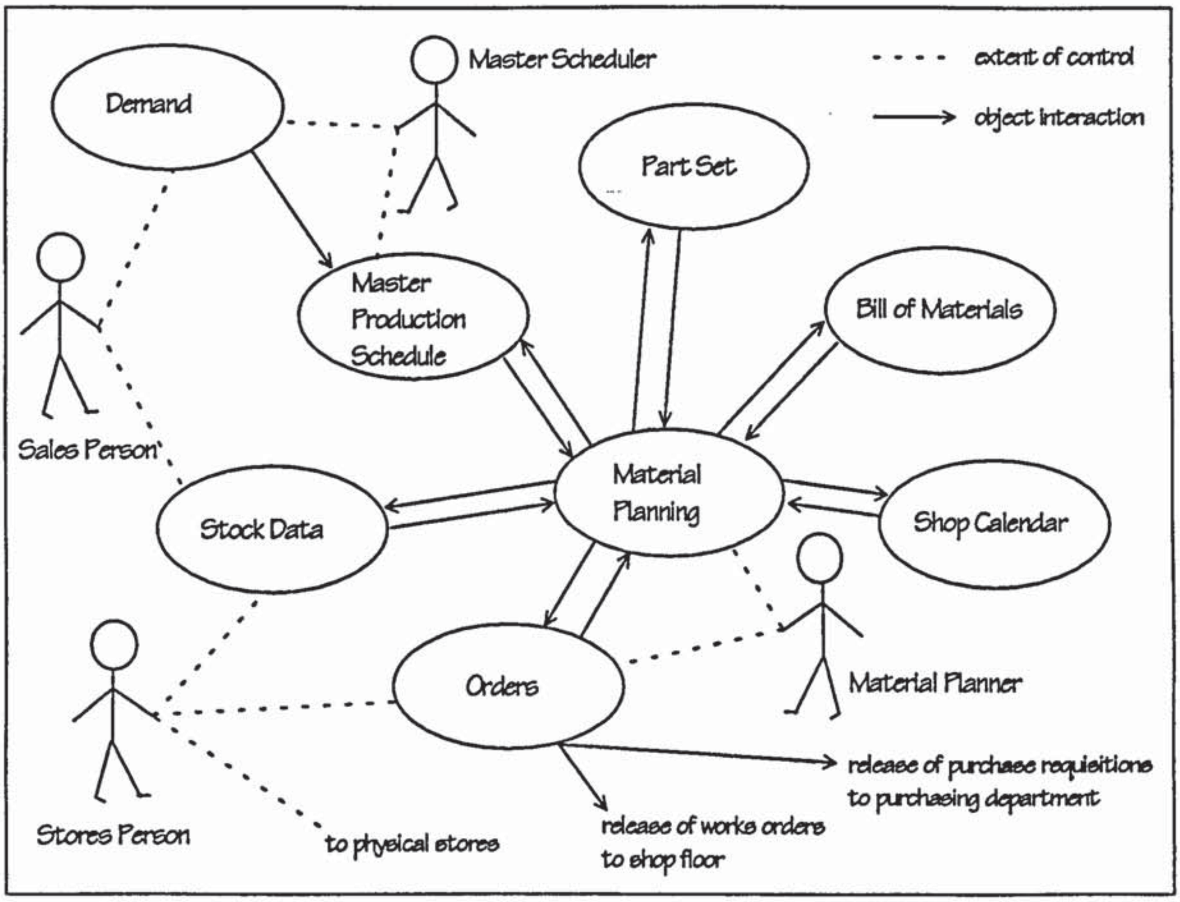


Figure 9.1: An example manufacturing scenario.

The model consists of several different classes: those representing the planning and control functionality and those representing the shop floor. The latter consists of objects representing machines, parts, operators and material handling capabilities (for example, hand trucks): these objects are not shown in Figure 9.1. The supporting simulation features of the Advanced Factory Simulator (AFS), for example the executive, time advancement mechanism and event list, will not be discussed here. Similarly, discussion on how to configure models from the user-interface will be covered in chapter 11.

The focus of this discussion will be the objects which provide the planning and control functionality. To aid identification, *italics* will be used throughout to represent class (or object) names. The objects in this particular model are instances of the following classes: a *Demand* class, a *MasterProductionSchedule* class, a *ShopCalendar* class, a *BillOfMaterials* class, a *PartSet* class, a *StockData* class, an *Orders* class and a *MaterialPlanning* class. These objects integrate with the shop floor objects and thereby provide a complete simulation model. Each of the classes will be briefly discussed in the context of the different activities.

The *Demand* class represents the future demand for the products which are manufactured and, as would be expected, is derived from the customers and represents the long-term requirements. This demand is translated into a future production programme covering the defined planning horizon by the *MasterScheduler* and is represented in the *MasterProductionSchedule* class in terms of end item, due date and order quantity. The *MasterScheduler* is a 'person' entity which can perform master production scheduling activities.

The *PartSet* class holds the information referring to all of the manufactured and purchased parts in the model: information such as 'part number', 'lead time', 'lot sizing policy' and 'planning horizon'. This class represents what would be termed the parts database in a typical manufacturing organisation. *PartSet* not only provides the informational aspects relating to parts, i.e. the data attributes, but also the mechanisms through which the attributes can be accessed and modified; the mechanisms are implemented as methods. Similarly, the basic stock records and transactions of an inventory system are provided by the *StockData* class and include, for example, booking in/out of stock items and the associated update of due-in/allocated parts. The *BillOfMaterials* class represents a manufacturing bill of materials and identifies the parent-child structural relationships. The *ShopCalendar* represents the manufacturing time available over a specified period (for example, one year); it identifies weekends, holidays and shutdowns, as well as the daily hours available. An object of this class is utilised during the material (and capacity) planning processes in order to plan production in the time and days which are available.

The master production schedule (MPS), i.e. the production programme covering a defined planning horizon, provides the initial input to most material planning activities. The same is true within the modelling framework provided by WBS/Control. Within a model the material planning activities are achieved by the *MaterialPlanner*, a 'person' entity with material planning skills, in conjunction with *MaterialPlanning* objects. Based on the current MPS, the current inventory status, the product structures and the shop calendar information, a set of suggested works and purchase orders are derived.

How the suggested orders are determined is based on the type of material planning performed. The *MaterialPlanning* object could, for example, represent a material requirements planning (MRP) based calculation or, alternatively, be based on re-order point (ROP) control. Alternatively, due to the flexibility offered through the *MaterialPlanning* class structure, unique control approaches may be implemented. The detail of how the material planning activities are included will be discussed later.

The derived works and purchase orders are only suggested by the material planning system: decisions need to be taken about which orders to confirm. In the real world, decisions about which of the suggested orders to release may be based on quite different factors, for example the immediacy of an order or whether there is a potential capacity over-load at a particular work centre. The latter may result in a modification of the MPS and further material planning 'runs'. The decisions are determined by the operational procedures of an organisation. Situations such as those described can be supported in the modelling environment described here. The process is either left with the *MaterialPlanner* to decide which of these orders to confirm or release or, alternatively, it can be carried out through manual intervention by the modeller, via the user-interface of the simulator. The released works orders, purchase requisitions and the current suggested orders are all held in their respective *Orders* objects.

As would be expected the works orders are released to the appropriate shop floor departments. The typical recipients are cell leaders, or shop foremen, who are 'people' entities with cell management skills. The cell leader may either distribute a works order directly to available operators or, for example, based on the current set of orders perform some form of scheduling or sequencing before releasing them to the appropriate operators. It should be noted that the orders for what would be in reality bought-in raw materials or components, i.e. purchased items, would be passed on to the modelled 'purchasing department'. Addressing the problems of modelling the support departments of a manufacturing organisation, for example the purchasing department, and integrating the appropriate functionality in AFS is the subject of a separate research project (see Jackson & Love, 1994). The focus of this illustration will be the planning and control system and the shop floor.

It would be wrong to assume that the simulated events can only occur sequentially. As in the real world, the modelled manufacturing and planning activities take place concurrently. Manufacture will be taking place continually during the daily shift, albeit subject to interruptions such as machine breakdown or material unavailability. On the other hand, the production planning activities may take place at predetermined times and frequencies. The development of the MPS could, for example, take place every Monday at 11.00 hrs whereas the detailed material planning activity could take place daily at 09.00 hrs.

Once a particular order has been completed on the shop floor, the operator responsible advises the cell leader accordingly; transport is arranged to move the completed batch to a subsequent operation (determined by the part routing) or, if complete, to the stores. On arrival at the stores an operator with stores management skills, i.e. the *StoresPerson*, will receive the physical stock and update the inventory system (represented by the *StockData* object). If configured to do so, the *StoresPerson* could also book in the completed order and thereby update the *Orders* object.

The entity defined by *SalesPerson* assesses the demand for the 'saleable' products and at the respective order due date looks to satisfy the demand, provided the finished goods are available. In essence, this aspect of the simulation closes the loop between an original demand, planning for production, manufacture and finally making a sale. However, both manufacture and planning activities continue throughout the duration of the simulation 'run'.

9.1.3 Discussion

The above illustration has provided an opportunity to summarise not only the range of functionality which is available in WBS/Control but also the scope of the models which may be configured. The range of classes represents those typically found in most manufacturing organisations and utilised in the material planning and control activities. Models consist of appropriate 'people' entities who perform their respective tasks. Just as a machine operator would participate in a simulation model, so too may a material planner or shop scheduler. The appropriate links to the shop floor are evident as well as the feedback necessary in order to maintain the different information 'systems'.

The combined functionality of WBS/Control and AFS provides a richer modelling environment and one which more closely represents the operations of the 'real world'. The integration of a planning and control system into the manufacturing system provides a completely different modelling capability: this is not the same as releasing a prescribed set of orders into a simulation model of the shop floor and running the model. WBS/Control allows the simulated planning and control procedures to be continually carried out in parallel with the simulation of the manufacturing facility. Consequently it can continually effect what happens on the shop floor.

The above illustration, as well as providing an introduction to the functionality of WBS/Control, has identified several important aspects which now require more detailed explanation. These aspects may be summarised as:

- information-type classes (chapter 9);
- material planning classes (chapter 9);
- object communication (chapter 10);
- 'people' entities (chapter 10);
- integration of WBS/Control and AFS (chapter 10).

The classes such as *PartSet*, *StockData* and *Orders* can be described as information-type classes. Each one represents its respective areas of information, in terms of data and behaviour, through appropriate attributes and methods. Information objects take a more passive role in a simulation model whereas the classes representing the material planning activities adopt a more active role.

The modelling of manufacturing planning and control systems through the *MaterialPlanning* class(es) provides an interesting application of the object-oriented paradigm. Different material planning approaches were mentioned in the above example, i.e. MRP and ROP, but the modelling capability of WBS/Control is not just confined to these. The way in which either of these approaches and alternative approaches can be modelled through the *MaterialPlanning* class(es) needs to be examined. The specification derived in chapter 6 placed considerable requirements on WBS/Control and set it apart from other approaches to this problem. The ability to integrate material planning activities into the simulation environment is one significant development.

There is a substantial amount of communication taking place between the different objects in the model described: not just between shop floor objects and planning and control objects but also between the different control objects themselves. The elements of communication, included through the object-oriented concept of 'messaging', are an important attribute of the system.

The role of the 'people' entities and their associated skills is one of the novel features of the AFS architecture which sets it apart from other manufacturing simulators. The exploitation of this functionality in WBS/Control not only adds richness and realism to the modelling environment but also means that the modelling capability is not restricted to the computer-based aspects of control systems. It is possible to include the 'policies, procedures and practices' (Maull et al., 1990) which are intended to support the computer-based processes: for example, a model may be configured so that goods received notes are batched together by the *StoresPerson* and only then are the inventory records and outstanding orders updated.

The objects are all part of the simulation model and, furthermore, because of their class structures they include all of the necessary mechanisms to participate and exist in the simulation world. Consequently, the natural timing and delays of the real system can be represented in the object(s). The feedback between the shop floor and the planning and control functions does not have to be immediate. A simple example would be an event which sets a MRP 'run' to take place over a weekend and last for several hours. Alternatively, the *MasterScheduler* can be set to perform any master production scheduling activities every Monday at 11.00 hrs.

The planning and control class library, WBS/Control, has been developed and integrated within an existing manufacturing simulator, AFS. It is important to recognise that it is through the open object-oriented architecture of AFS that WBS/Control can be integrated. Although this architecture will be briefly discussed in this thesis, a more complete discussion can be found in Ball (1994) and Ball & Love (1994).

9.2 Information classes

One of the opportunities offered by the adoption of object-oriented methods is that the elements which populate the developed systems can more closely reflect those found in the real world. Object-oriented methods support a closer correlation between the system under investigation and the software solution. This is achieved through the fundamental component of object-orientation, the object; objects capture the essence of the problem domain. If one considers the manufacturing facility, i.e. the shop floor, certain classes of object readily spring to mind: machine, part, truck, container. The classes encapsulate the data and behaviour of the respective entities. The application of object-oriented methods to the domain of manufacturing planning and control offers equivalent opportunities.

It is important that the designed planning and control classes map well to their real world counterparts. However, it will become apparent from this and subsequent discussion that that there are elements of the planning and control domain which map more readily than others to object classes.

The classes included in the above illustration such as *PartSet*, *StockData* and *Orders*, intuitively represent the entities to be found in the real world. These classes will be described as 'information' classes: each class represents their respective areas of information. The information classes provide a close correspondence to the entities that they represent, both in terms of the data attributes and the behaviour which is implemented as methods.

The current information-type classes present in WBS/Control include *PartSet*, *StockData*, *BillOfMaterials*, *ShopCalendar*, *RouteData*, *WorkCentreData* and *Demand*. The class *Orders* is itself made up of three different order classes: *SuggestedOrders*, *WorksAndPurchaseOrders* and those represented by the *MasterProductionSchedule* class. These information classes are described in Table 9.1 and their implementation will be discussed in chapter 10: summary documentation is provided in Appendix 1.

The identification of what have been termed information classes has also been considered by other researchers. Bhuskute et al. (1992), in their development of a simulation environment, visualise the 'world' from three different planes: a physical plane, a control plane and an information plane. They suggest that this results in a structured approach to the abstraction of the objects found in the problem domain. Furthermore, the classes included in the information plane collectively model the behaviour of the informational aspects of a manufacturing system. The range of classes is however restricted: offering only *CustomerOrder*, *ShopOrder*, *Operation*, *Routing*, *BOM* and *ItemMaster*. Furthermore, close inspection of the *ItemMaster* class reveals a certain amount of overloading: it includes attributes associated with alternative classes. It is reasonable to suggest that this class is a combination of part data and inventory data.

CLASS NAME	CLASS DESCRIPTION	ASSOCIATED CLASSES
PartSet	Equivalent to a parts database: location of all information relating to parts.	PartList, PartData.
StockData	Equivalent to an inventory system: includes stock status for all parts and monitors all stock transactions via a stock audit trail.	StkList, StkItem, TransactionList, TransactionData.
BillOfMaterials	A manufacturing bill of materials: identifies the parent-child structural relationships.	ParentList, Parent, ChildList, Child.
ShopCalendar	Configures the available manufacturing time over a defined period: identifies weekends and shut-downs, available daily hours and overtime.	PeriodList, PeriodData.
RouteData	Defines the sequence of operations for all manufactured parts: identifying the work centres involved.	Parts. PartRoute, OpList, OpData.
WorkCentreData	location of all information relating to the different work centres: includes the calculated capacity loading (based on the production programme).	WcList, WorkCentre, WcLoading, LoadingData, McList, McData.
Demand	Defines the product demand over the duration of the simulated time.	DemandList, DemandData.
MasterProduction Schedule	Defines the master production schedule (MPS).	MpsOrder.
SuggestedOrders	These orders are suggested (generated) by the material planning process and can be based on the MPS.	PlannedOrders.
WorksAndPurchase Orders	These are confirmed orders, released to the shop floor (or purchasing dept.), the inventory records identify the respective due-in and allocated quantities.	WPOrders.

Table 9.1: Current 'information' classes present in WBS/Control.

The information classes provided by WBS/Control appropriately encapsulate the data and functionality of their real world counterparts as well as supporting a wider range of different classes. The information classes, as with all objects, are represented by a combination of data attributes and methods; the methods support the behaviour of the class which includes the interactions with other classes as well as the management of the attributes. The concepts of information hiding and encapsulation mean that access to the data by different classes is restricted to the interface of the class: direct manipulation of attributes is not, therefore, allowed.

Figure 9.2 represents a selection of the attributes and methods of the implemented class *PartData*. (The relationship between the classes *PartData* and *PartSet* will be explained in the subsequent implementation discussion.) Consider 'part0001', an object of class *PartData*. Since an object is an instance of a class, the class attributes are assigned the actual values which differentiate the object from other objects of the same class. The object 'part0001' will be defined by the values assigned to the attributes nLeadTime, nLowLevelCode, pLotSizingPolicy and nMinOrderQty: for example '2', '0', 'FixedOrderQty' and '25' respectively.

```
class        : tPartData

parent class :

attributes   : cPartNumber
               nLeadTime
               nLowlevelCode
               pLotSizingPolicy
               nMinOrderQty
               . . .

methods      : GetPartNumber
               ModifyPartLeadTime
               SetLowLevelCode
               SetLotSizingPolicyTo
               GetMinOrderQty
               . . .
```

Figure 9.2: Selected attributes and methods from *PartData*.

There are two points to notice if another object, of a different class, requested the current minimum order quantity for 'part0001'. The first is that the request could only be made via the interface of the object and secondly, in response, this value would be obtained by the part object's own method 'GetMinOrderQty'. The value of '25' would be returned.

The information classes described are in essence 'passive' classes in the simulation environment: they store data which is utilised and/or modified (indirectly) by other classes. It is these 'other' classes which take an 'active' role in the simulation. Booch

(1994) defines objects which can operate upon other objects but are not themselves operated on by other objects as 'actors'. In contrast, those objects which can both operate on other objects and be operated on by other objects are defined as 'agents'. In the example described above (see also Figure 9.1) it is the *MaterialPlanning* object and the 'people' entities which adopt roles similar to Booch's 'actors' and 'agents' respectively.

9.3 Material planning classes

9.3.1 Introduction

The identification and design of material planning classes represents a significantly different application of object-oriented principles to this particular problem than has been identified in the literature to date. Figure 9.1 represents the material planning activities by the class *MaterialPlanning*. It is the purpose of this section to explain 'how' these activities are achieved. Chapter 6 defined a specification of a simulation-based modelling tool which would support the evaluation of both design and operational issues surrounding manufacturing planning and control systems. The specification, in the form of a set of requirements that the WBS/Control must satisfy, is summarised here:

- to adopt a user-oriented perspective, or 'world view';
- to configure models without recourse to programming;
- to properly integrate with shop floor operations;
- to be able to model different planning and control systems;
- to be able to model hybrid control systems;
- to include the elements of the planning and control infrastructure;
- to be able to extend the current functionality.

It was argued in chapter 8 that the application of object-oriented methods is the most appropriate mechanism by which the above requirements can be satisfied. The relevance of object-orientation both to manufacturing and simulation has been established. Furthermore, the object-oriented property of classes supports the 'building block' approach advocated in chapter 6.

The problem now becomes one of identifying the material planning classes. This is not an insignificant problem. In the case of the information classes described above, it is considered that they have been designed to satisfy the requirements placed upon them. Furthermore, the definition of these classes matches what one would be expect to find as elements of the real planning and control system. The situation is similar to that of identifying shop floor classes: based on an analysis of the problem area, deriving a range of classes such as machines, parts, trucks, etc. is to be expected. However, identifying classes which represent planning and control systems, as well as satisfying the outlined requirements, is not so clear. Not only must the classes represent conventional, or standard, systems such as MRP, ROP and kanban, but also hybrids and potentially new variations. Additionally, there is an opportunity to support the modelling of more company specific implementations. To satisfy such scope requires representative classes which are extremely flexible in their configuration.

In order to identify the potential material planning classes, an analysis of the problem domain was necessary. Because of many different planning and control systems which warrant investigation, initial analysis has concentrated on a subset, namely MRP and ROP. These provide an appropriate starting point since MRP and ROP are perhaps the most widely implemented manufacturing control systems (Newman & Sridharan, 1995; Lopez & Haughton, 1993; Little & Johnson, 1990). The understanding and insight gained from analysing these systems can then be extended to other systems.

9.3.2 A macro approach to class definition

An initial approach to defining the material planning classes, though not subsequently implemented, was the macro approach. By considering its design here, useful insight into the approach finally adopted will be achieved.

The range of classes which might represent a MRP system are illustrated in Figure 9.3. There are some similarities between the *BOM* and *Inventory* classes defined here and those later adopted as information classes, i.e. *BOM* and *StockData*. Additionally, the separation of the different classes in the way illustrated would appear appropriate: a comparison with Figure 9.1 emphasises this. However, the way in which the MRP and MPS class have since been implemented is significantly different from this original design.

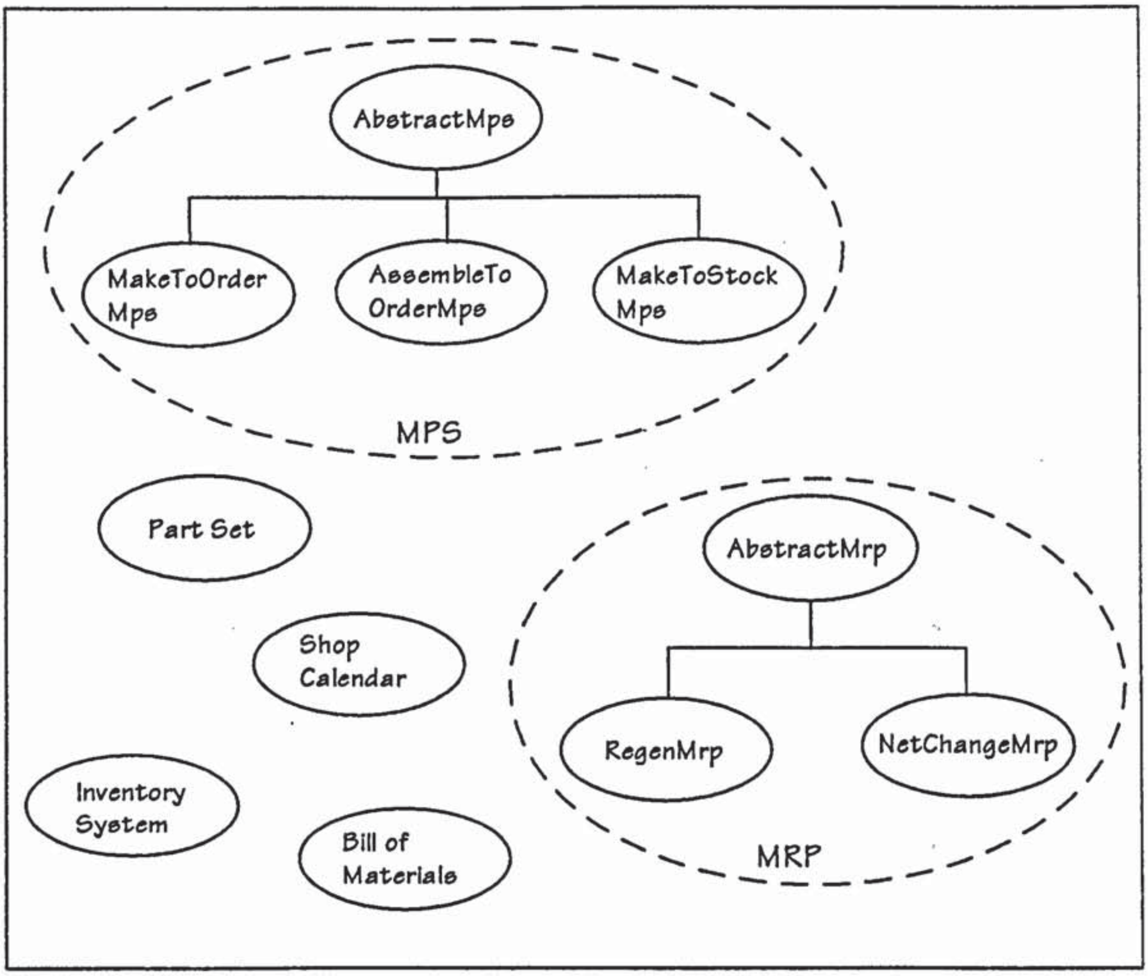


Figure 9.3: Some initial MRP-type classes.

The MPS and MRP classes suggested by Figure 9.3 offer what is considered to be a macro representation of the respective systems. Although a similar approach has been taken by Persentili & Alptekin (1993), they only offer a single (and 'standard') MRP class. The intention here is less restrictive: the MRP system could either be a 'regenerative' system or a 'net change' system. A class hierarchy developed along these lines would have a parent class, for example, *AbstractMrp*, which would contain all of the data attributes and methods common to both a regenerative system and one of net change. The lower level classes, for example *RegenMrp* and *NetChangeMrp*, would represent the data and methods specific to their respective approaches. The common data and functionality would be inherited from the parent class *AbstractMrp*.

Similarly, the different approaches to master production scheduling could be implemented by classes representing the different functionality. For example, a make-to-order (MTO) approach would be represented by the class *MTOMps* whereas a make-to-stock (MTS) approach would be represented by *MTSMps* and so on. A class hierarchy would naturally develop: a parent class, for example *AstractMps*, would contain the data and functionality common to MTO, MTS, etc. and would be inherited by the respective child classes.

There are, however, several problems with this approach. This analysis has suggested classes which take a more macro, or conventional, view of the problem domain, resulting in classes such as *NetChangeMrp* or *MTSMps*. Adopting this approach would populate the class library (i.e. WBS/Control) with classes suitable for modelling standard systems but it offers little flexibility. It would be difficult, for example, to represent a new control system design which, although based on regenerative MRP, is a significant variant. This would require the creation of a new class.

This problem is emphasised further if a more radical design is suggested or if more company specific planning and control systems are to be modelled. The latter case provides an interesting modelling problem: an organisation may have implemented a conventional control system but made one or two modifications in an attempt to make the system more closely represent the characteristics of the company. For example, the author is aware of an organisation which has a MRP system but, because little or no finished goods are stocked, does not need to net the order quantity against the inventory status. This situation would be difficult to model using the type of material planning classes proposed so far.

The adoption of this more macro view is at variance with the intended application of WBS/Control identified through the above requirements specification. The problem can be illustrated through use of the building block, or LEGO® brick, analogy introduced in chapter 6. The macro view parallels a situation where there exist 'house' building blocks, 'tree' building blocks, 'car' building blocks and so on. This is fundamentally different from the actual implementation where the building blocks, or bricks, are much more elementary. It is from these elementary building blocks that a house, or a tree or a car, can be constructed. Furthermore it will be likely that many building blocks will be used in any one of the models: in terms of object-orientation this equates to the reuse property. Clearly this cannot be achieved through the macro approach described; a more elementary view is required.

It should not be assumed, however, that a more macro interpretation of classes does not have a potential role in simulation modelling. Boughton & Love (1994) discuss a planning and control class library populated by two types of classes: high and low level classes. The high level classes provide less detailed representations of the planning and control functions. These classes take a more aggregate view of the problem domain and would replace the detailed interpretations achieved through a configuration of low level classes. Figure 9.4 represents a range of different planning and control classes which would eventually populate WBS/Control.

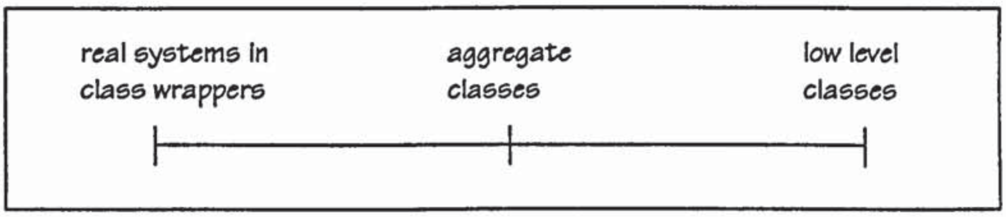


Figure 9.4: A range of planning and control class types.

This range of classes will be addressed in chapter 12. There are however two points to note here: the first is that the different classes provide different modelling alternatives and thereby extend the modelling scope of the class library. The architecture of WBS/Control must, therefore, be able to support these different types of classes. The design of low level classes will now be introduced.

9.3.3 Low level classes

The previous discussion has emphasised the need to take a more elementary, or low level, view of the planning and control domain. This approach concurs with that of Rolstatås (1988b) who had earlier suggested:

> *"One of the possibilities for better understanding of production planning goes through defining the fundamental operations of a planning system. Different systems could then be synthesized by different combinations of these basic operations."*

This approach is particularly relevant if control systems such as MRP, ROP or kanban are to be constructed from the elementary classes rather than classes representing systems *per se*. The classes which have since evolved take a 'micro' view of the situation and thereby provide the necessary flexibility.

Different planning and control systems can be configured from these classes, some of which will be common to different systems. The classes defined follow more accurately the building block analogy and will be described here. Although the following description concentrates on the medium-term planning activity, the concepts apply equally across the planning and control hierarchy.

Close inspection of the MRP calculation identifies exactly the processes which determine the material requirements. Based on the standard inputs to the MRP calculation, the following activities can be identified. The original order quantity for a part (defined in the MPS) is first of all netted against the current level of stock; the revised order quantity is then batched according to the particular lot sizing policy for the part; in order to identify when the manufacture should begin, the order due date is then offset according to the product lead time. This determines the 'planned' or 'suggested' order for the demand level item. This order then places a demand on the components of the part, which are identified by the next level in the bill of material; the process then continues for these component parts.

This process is depicted in Figure 9.5. The different steps are organised sequentially along with the required information and its source. The gradual development of the suggested order for 'part0001' is illustrated: the MPS based order for 25 required by 20/07/95 is translated into a suggested order for 50 to be launched on 18/07/95. This order then triggers a demand for the component parts of 'part0001', i.e. 'part0002' and 'part0003'; they then follow the same process to determine the respective suggested order quantities and launch dates.

Similarly, Figure 9.6 illustrates the activities involved with ROP control. The current inventory status for each of the parts is compared with the defined reorder level; if the current stock value is below the reorder level, an order is generated for the difference; the order quantity is revised according to the lot sizing policy for the part; the order due date is then determined by 'onsetting' the due date by the product lead time.

The analysis of these two systems has provided a completely different view of the problem and suggested classes which are more elementary in their structure. Returning to the MRP example, it is considered that the processes 'gross to net', 'batch order quantity', 'offset by lead time' and 'explode demand' can be encapsulated in classes; these classes will be akin to 'actor' classes (Booch, 1994) whereby objects of actor classes can operate upon other objects. This leads to the following potential classes: *GrossToNet*, *Batching*, *OffSet*, *ExplodeDemand*.

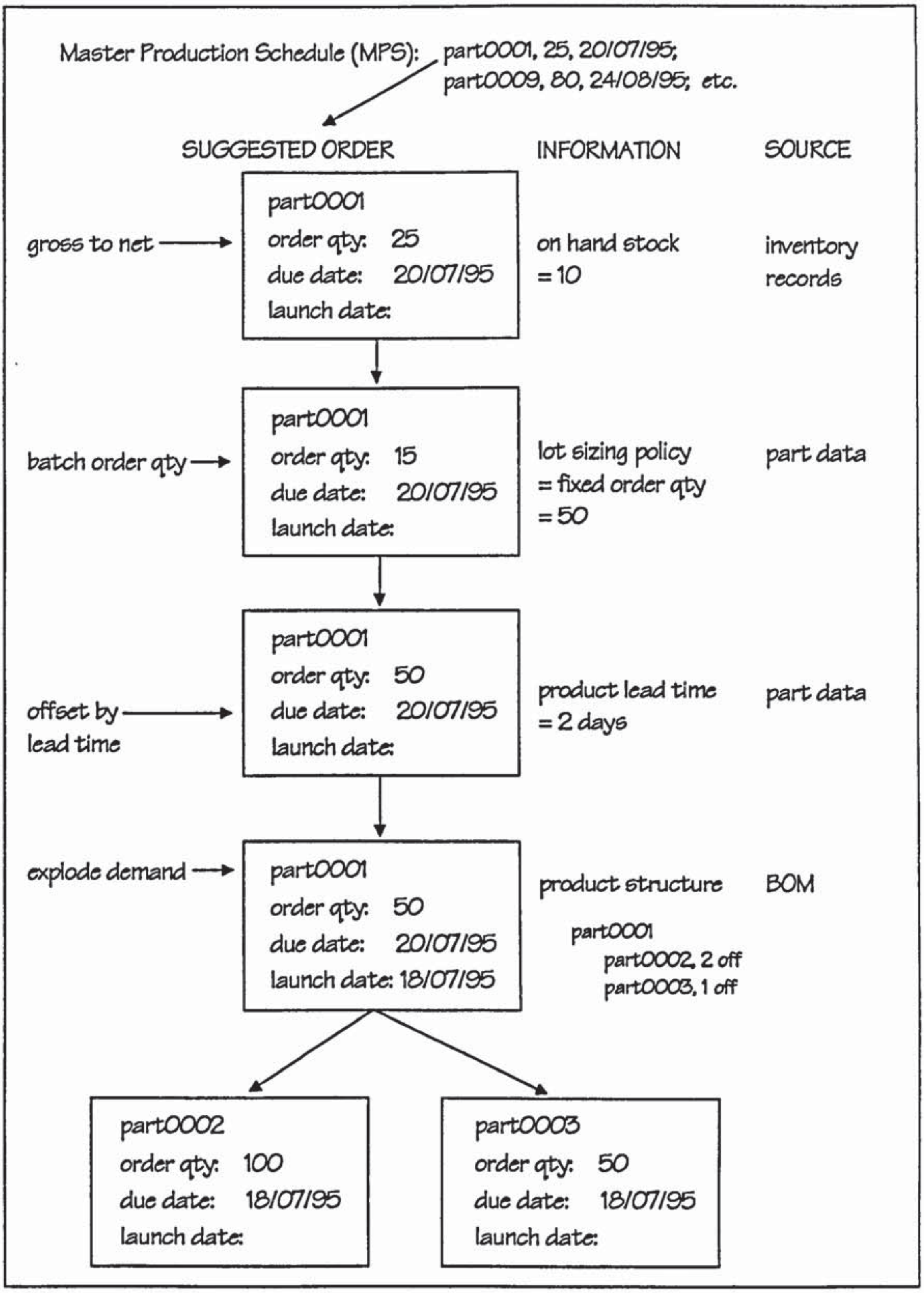


Figure 9.5: The core MRP calculation.

An object of the *GrossToNet* class would operate upon a suggested order (itself an object) and determine the net order quantity. In order to perform this process certain information is required: the original order quantity and the current stock level. As Figure 9.5 indicates, the 'on hand' stock can be obtained from the inventory records, whereas the original order quantity will be held with the suggested order. Following the object-oriented principles, the *GrossToNet* object will obtain this information by messaging the appropriate objects; 'messaging' will be discussed in the following chapter. Similarly, an object of class *OffSet* would determine a suitable 'launch date' to begin production. The 'due date' would be offset by the part lead time, the latter being obtained from an object of class *PartSet*. The determined 'launch date' would then be checked, via a *ShopCalendar* object, to make sure it does not fall on a non-working day.

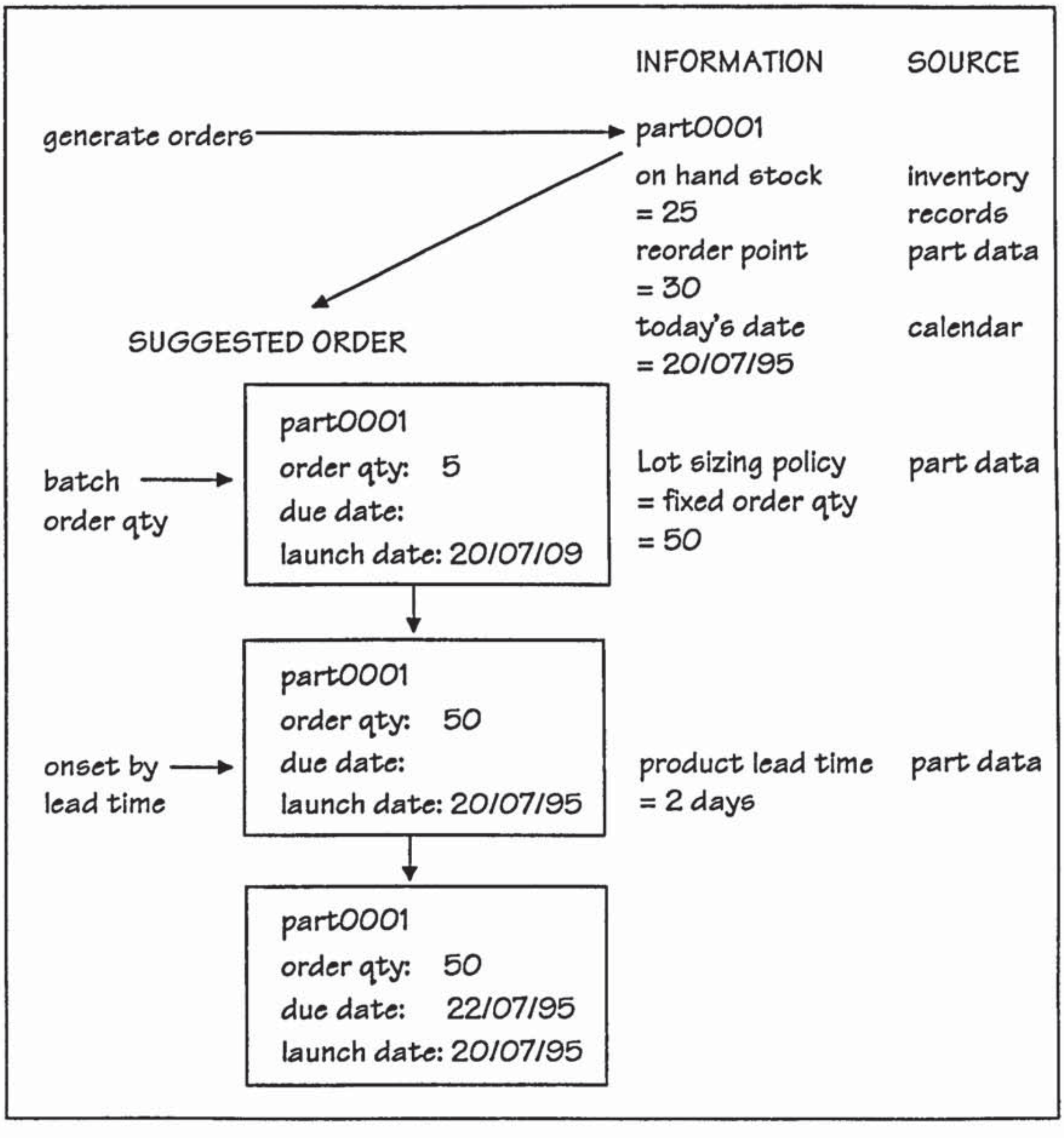


Figure 9.6: The activities associated with the ROP system.

Figure 9.7 parallels the MRP process described in Figure 9.5 and represents the object interactions which would take place. In essence, the *MaterialPlanning* object shown in Figure 9.1 is implemented by the shaded material planning objects.

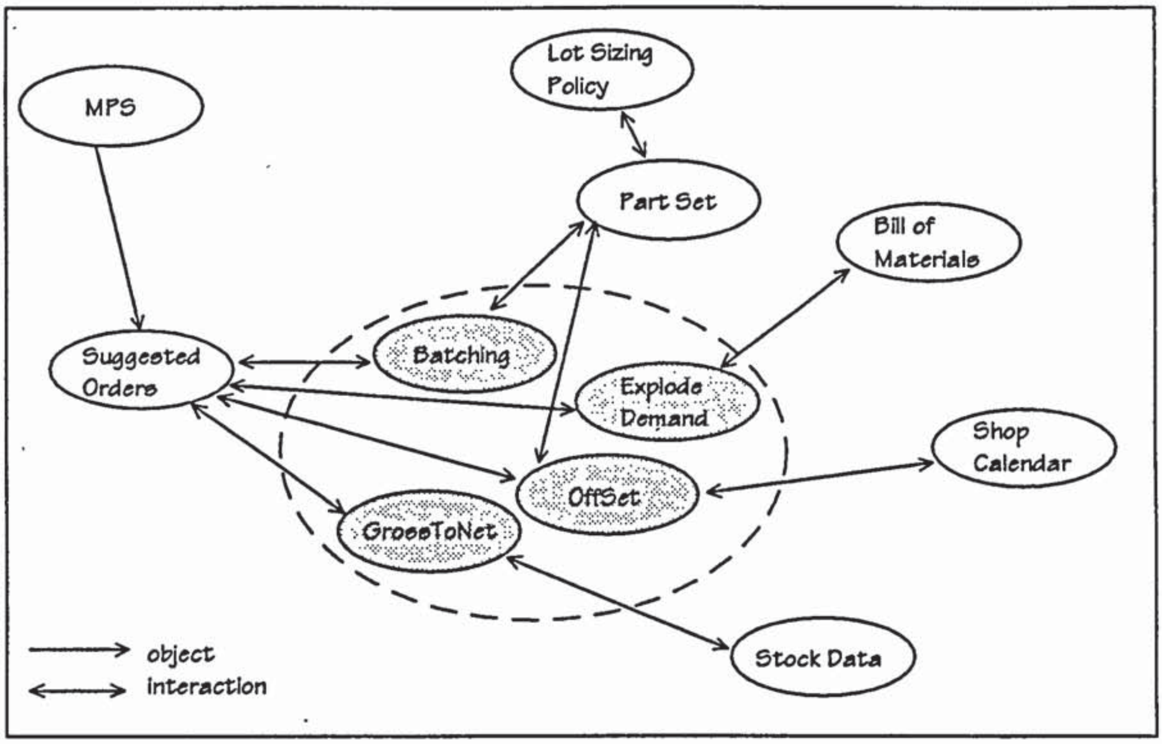


Figure 9.7: A low level object representation of MRP.

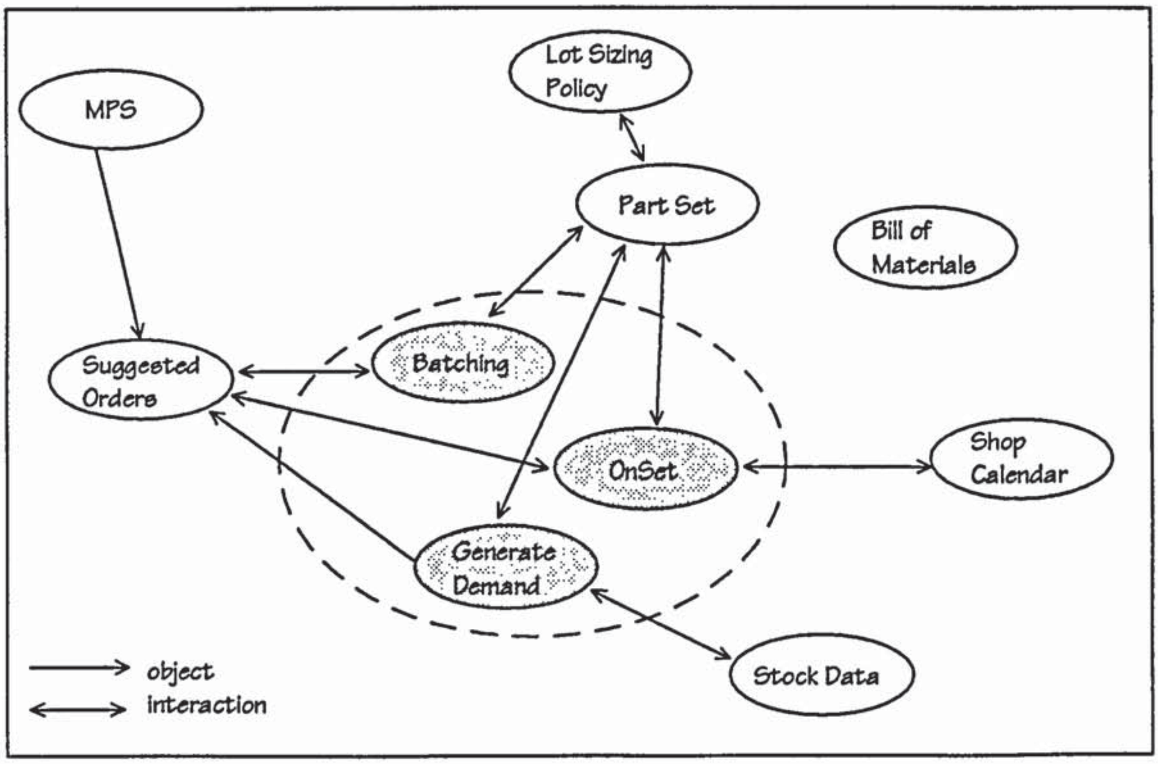


Figure 9.8: A low level object representation of ROP.

In the case of a ROP system, the analysis illustrated by Figure 9.6 has identified the following potential classes: *GenerateOrders*, *Batching* and *OnSet*. The object interactions are illustrated in Figure 9.8 which parallels the processes illustrated by Figure 9.6. It will be apparent that the only new classes required for this configuration are *GenerateOrders* and *OnSet*: the class *Batching* is the same as that derived for a MRP configuration.

Developing classes in this way offers a flexible approach to the construction of models of planning and control systems which can be demonstrated by a simple example. The specific company MRP implementation described previously, which does not include the 'netting' process, can be modelled using the classes so far described. The material planning configuration illustrated in Figure 9.7 would be replaced by *Batching*, *OffSet* and *ExplodeDemand*. The flexibility is further emphasised by the use of the *Batching* class in either the MRP or ROP implementations; this is equivalent to common building blocks being used across different models and represents a form of reuse.

This section has introduced 'low level' classes which provide the material planning functionality of WBS/Control. The low level classes discussed so far cover some of the material planning aspects of a planning and control system. A complete description of the low level material planning classes can be found in Appendix 1. It is believed that further low level classes can be derived to perform capacity analysis, the development of a master production schedule and shop floor control. The flexibility of this approach means that models of a range of possible systems can be configured: from standard implementations of a particular control system to more radical designs.

In order not to detract from the design of the low level classes, several issues have been deliberately omitted from this chapter. These issues, to be discussed in the following chapters, include:

- the implementation of the designed classes;
- how a control system model is configured so that the selected processes are actioned in the correct sequence;
- modelling hybrid planning and control systems;
- the role and potential of messaging between objects;
- the inclusion of 'people' entities;
- the inclusion of 'policies, procedures and practices';
- the relationships between the classes of WBS/Control and AFS.

9.4 Summary

This chapter introduced the different types of classes which will populate a novel planning and control class library called WBS/Control. These classes have been divided into information classes and material planning classes; the flexibility of the latter offers an opportunity to represent different manufacturing control systems. The functionality of WBS/Control can be combined with that provided by AFS, an object-oriented manufacturing simulator, to provide a comprehensive modelling environment.

10. The implementation of WBS/Control

This chapter details the implementation of WBS/Control. The design of some of the key class types was discussed in the previous chapter; it is the intention here to explain how these designs have been transformed into a usable system. The relationships between WBS/Control and the Advanced Factory Simulator (AFS) will be described in the appropriate context.

10.1 Overview

WBS/Control is an object-oriented class library which provides manufacturing planning and control functionality for use in simulation models. The design and implementation of the class library means that WBS/Control can be used to support the evaluation of manufacturing control systems from both design and operational perspectives. WBS/Control is to be used in conjunction with the Advanced Factory Simulator (AFS) (Ball, 1994).

WBS/Control has been implemented using an object-oriented programming language, namely Borland Pascal with Objects, and Microsoft Visual Basic. (Full reference to all commercial software used in the development can be found in Appendix 3.) The core functionality of the class library has been developed in Borland Pascal; the user-interface has been developed in Visual Basic and provides a Microsoft Windows based application. The complete simulation environment, i.e. AFS and WBS/Control, has been developed for use on PC-based systems; the hardware and software requirements are detailed in Appendix 4.

The software which has been developed is fully documented within the source code itself and can be found in the Borland Pascal files. However, for completeness, summary documentation has been included in Appendix 1.

10.2 The architecture and functionality of AFS - an introduction

AFS is a manufacturing simulator which not only provides shop floor modelling functionality but also a 'host' simulation architecture. A brief discussion of the main aspects of both the architecture and functionality will be included here; a more complete discussion of AFS can be found in Ball (1994) or Ball & Love (1994).

Further reference to either the architecture or the functionality will be made in the context of the subsequent WBS/Control discussion.

It is important to clearly distinguish between the functionality of WBS/Control (developed by the author) and that already provided by AFS. The reader should readily distinguish:

- the functionality of WBS/Control;
- the functionality of AFS used 'as-is'.

10.2.1 The macro architecture

There are several requirements made of any manufacturing simulator: to be able to build and modify models, to perform simulation experiments and to provide a means of viewing simulation models while they are running (i.e. animation). These requirements are fulfilled by AFS and are represented in the 'macro architecture' (see Figure 10.1).

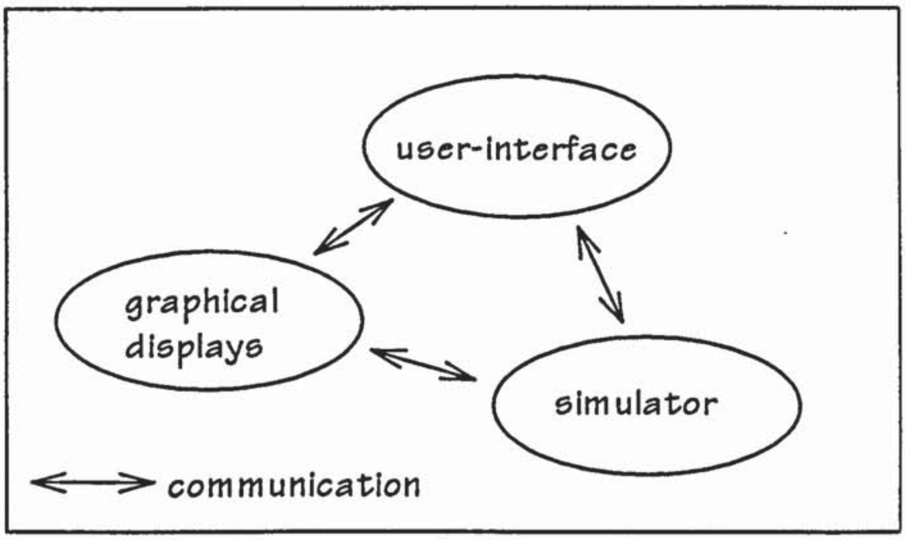


Figure 10.1: The macro architecture of AFS (after Ball et al., 1994).

The 'simulator' is responsible for the model representation and simulation experiments; it can be divided into two parts: application and support classes. The application classes are those which provide the modelling functionality: for example, machines, parts, operators and trucks. The simulation-based features, for example the event list and clock, are provided by the support classes. It is through the 'user-interface' that a user (or modeller) communicates with the simulator to configure, edit and run simulation models; the experimental results are also available through the user-interface. The 'graphical display' provides a means by which a simulation model and its 'runs' can be viewed. This aspect supports model verification during the model building stages as well as animation of the simulation 'runs'.

The elements of the simulation environment are provided through the applications of this macro architecture. An important point to note is that the architecture makes no assumptions about the functionality which it supports (Ball et al., 1994). This property means that any functionality can be supported by the architecture, provided that it conforms to certain required standards. This has particular relevance to the introduction of planning and control functionality via WBS/Control.

10.2.2 AFS core and functionality

In order for new classes to be supported by the architecture of AFS, they must inherit one of the core, or abstract, classes of AFS; this class hierarchy is illustrated in Figure 10.2.

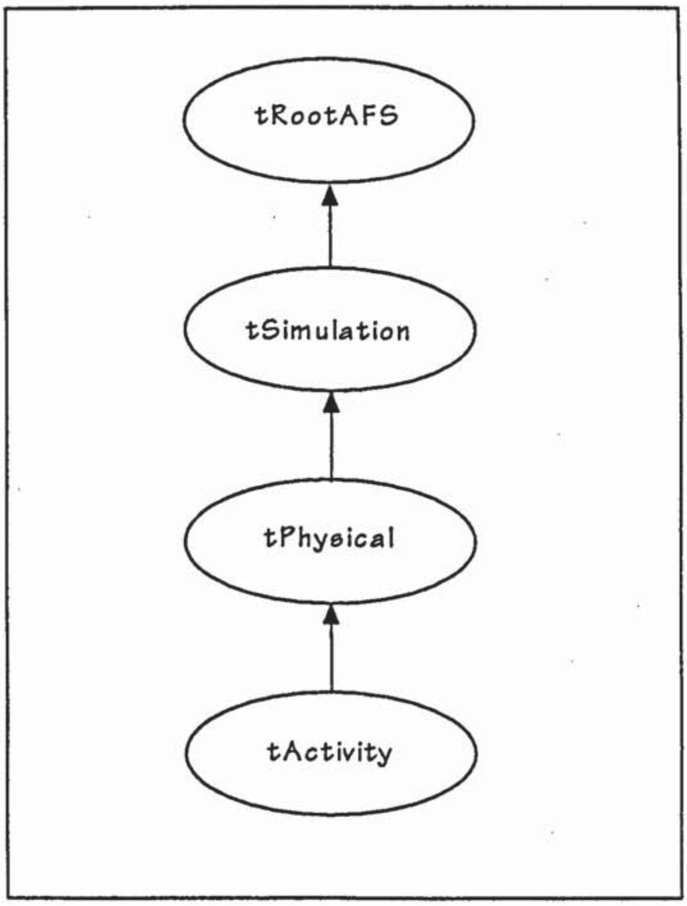


Figure 10.2: The core (abstract) class hierarchy of AFS.

All classes of the AFS architecture are descendants of class *tRootAFS*. This class provides data and methods by which all objects can be differentiated and monitored within a model. The class *tSimulation* provides the simulation capabilities: to possess and move between different states, for example 'idle', 'loading' and 'running'. It also provides access to the simulation clock and event list as well as collating results; *tSimulation* is a descendant of *tRootAFS*. The class *tPhysical* provides the ability for objects to 'physically' exist in the simulation world; the properties of position, size, icon, etc. are provided. Inherited from *tPhysical* is the class *tActivity*: this class provides the messaging mechanisms through which objects can communicate.

There are two points to notice from the class hierarchy described here: the abstract nature of the classes and the consequences of the inheritance mechanism. The classes which have so far been described are all 'abstract' classes which means that no instances of these classes , i.e. objects, will ever exist. For example, there will not be *Physical-1* or *RootAFS-4*, i.e. objects of *tPhysical* and *tRootAFS*. No object of an abstract class exists because it does not have a real world equivalent.

It is through the inheritance mechanism that descendant (or child) classes include the properties of their parent within their own functionality. In the classes so far described, this means that *tActivity* inherits the properties (i.e. data and methods) of *tPhysical* and thereby gains the properties of *tSimulation* and *tRootAFS* as one moves up the class hierarchy. The benefits of software reuse are possible through appropriate class hierarchies and inheritance; AFS exploits this object-oriented property.

Figure 10.3 illustrates a wider selection of AFS classes in the context of their class hierarchies. Classes which are at the lowest level of a hierarchy have real world equivalents and thereby support the existence of objects. For example, *ManualMachine01* is an object of class *ManualMachine*. This object is able to send and receive messages because it inherits these properties from *tActivity*; the machine-like properties are inherited from *AbsMachine* and *OperationStage*. The class *BatchStore* inherits general 'storage' properties directly from its parent class *Storage*; it can physically exist due to the fact that its parent inherits those attributes from *tPhysical*.

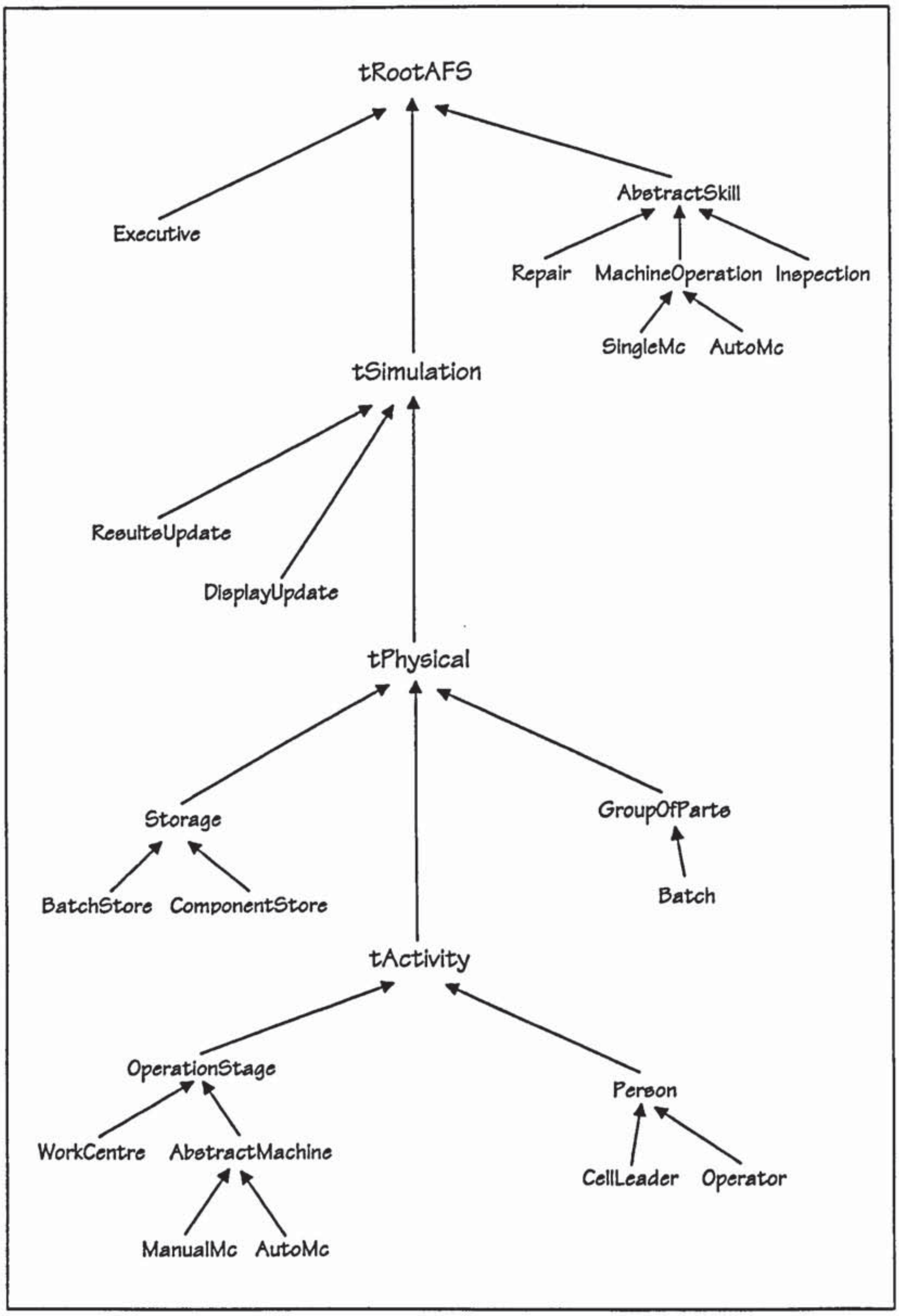


Figure 10.3: Selected AFS class hierarchy (after Ball, 1994).

10.3 Class structures and hierarchies from WBS/Control

This section will explain how the information-oriented classes of WBS/Control, identified in the previous chapter, have been implemented; in other words how the design has been translated into 'coded' classes which can be integrated into the AFS architecture. In order to retain clarity, the key aspects of the implementation will be drawn from a subset of WBS/Control classes; the class structures, relationships and hierarchies for all current classes are illustrated in Appendix 2.

Figure 10.4 illustrates the implementation of the classes *PartSet*, *PartList* and *PartData*. This 'set' of classes is equivalent to a parts database and provides all of the information relating to the manufactured or purchased 'parts' involved in a particular model. These classes are information-type classes (described in the previous chapter).

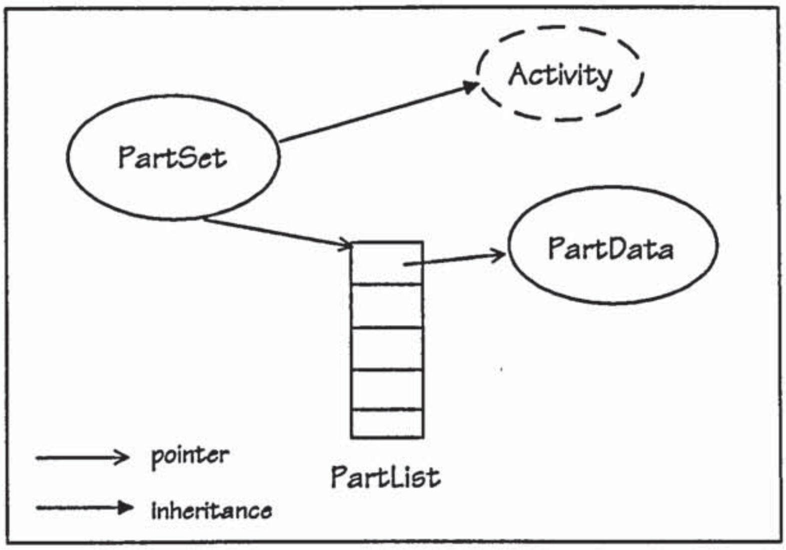


Figure 10.4: The class representation of a parts database.

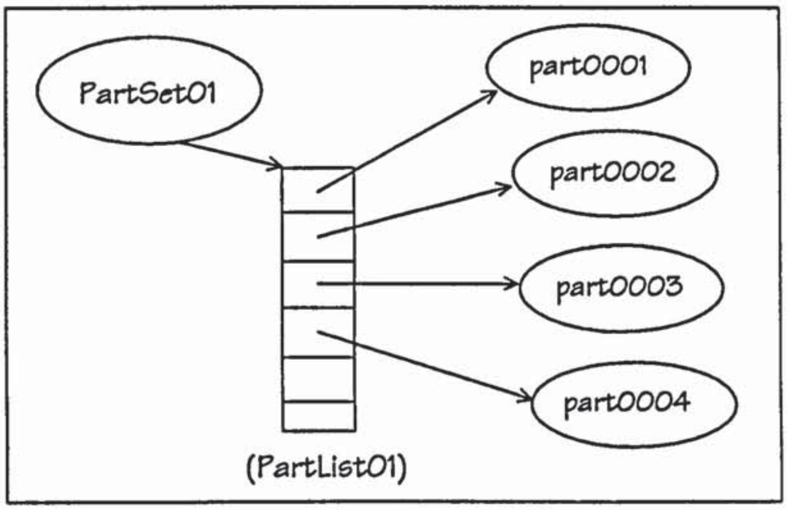
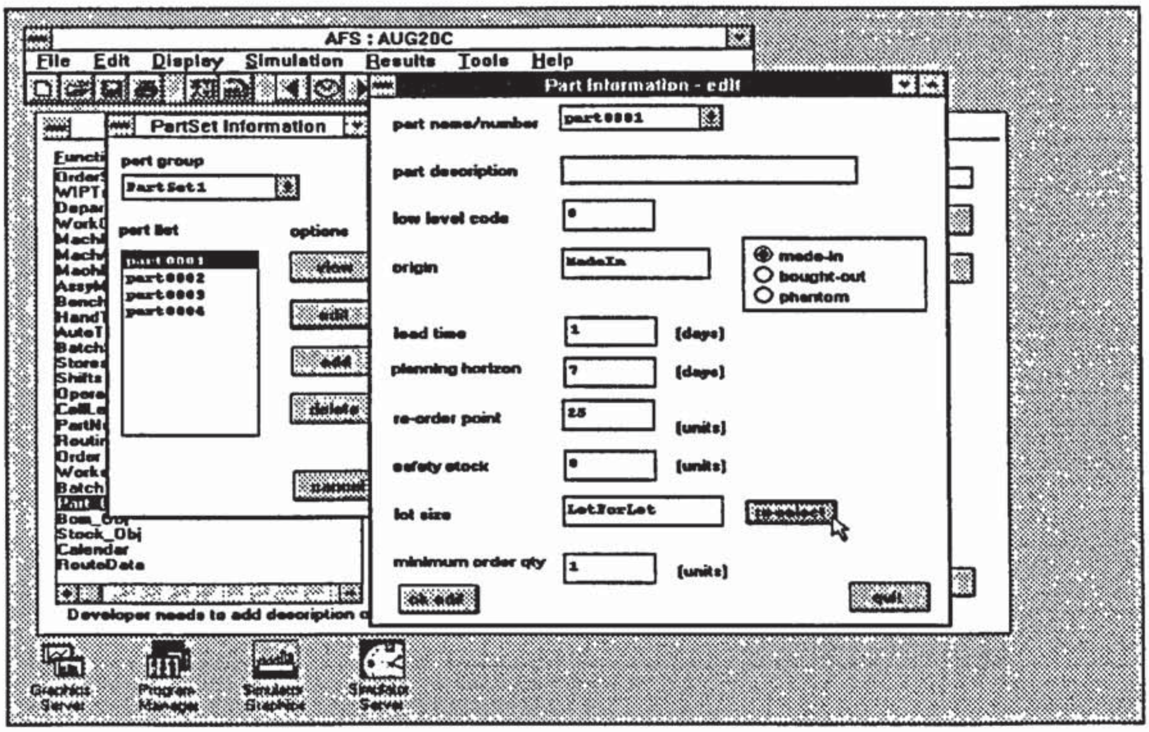


Figure 10.5: Example part objects.

The class *PartSet* provides all of the data and methods necessary to develop and maintain the (model) parts database; the class *PartData* defines all of the data and methods associated with an individual 'part record'. For example, if a model included four parts, 'part0001', 'part0002', etc., each part object would be an instance of *PartData* and would represent the associated part information as illustrated in Figure 10.5. All of the part objects are identifiable with *PartSet01* through the use of a 'collection' of 'reference pointers' (or pointers) which are contained in the class *PartList*; the collection of pointers is a Borland Pascal mechanism. The part information is made available to the modeller via the view/edit dialog of the user-interface; an example dialog is illustrated in Figure 10.6. (A more detailed discussion of the user-interface is included in chapter 11.)



part group: PartSet1

part list:
part0001
part0002
part0003
part0004

part name/number: part0001

part description:

low level code: 0

origin: MadeIn
● made-in
○ bought-out
○ phantom

lead time: 1 (days)

planning horizon: 7 (days)

re-order point: 25 (units)

safety stock: 0 (units)

lot size: LotForLot

minimum order qty: 1 (units)

Figure 10.6: The part class view/edit dialog.

The class *PartSet* is a descendant of *tActivity* and can, therefore, send and receive messages to other classes whose parent is *tActivity*. The classes *PartList* and *PartData* are related to *PartSet* but do not inherit any properties from AFS. This means that access to individual objects of type *PartData*, by other objects involved in the model, can only be made via a related object of type *PartSet*, i.e. *PartSet01* in the above example. This implementation has been chosen for three reasons. The first is one of representation: it is considered that this implementation offers a form of database representation to be expected of such an information system; in essence the representation provides an object-oriented parts database. Secondly, this approach emphasises the encapsulation property. The final reason is to ensure that the internal class and object lists provided by AFS are not populated by the individual part objects: only an object of type *PartSet* will appear on these lists.

The implementation of other information classes, for example *StockData* and *ShopCalendar*, has been achieved along similar lines to those described above.

Figure 10.7 illustrates the class hierarchy associated with 'lot sizing' functionality; different lot sizing policies are supported through these classes.

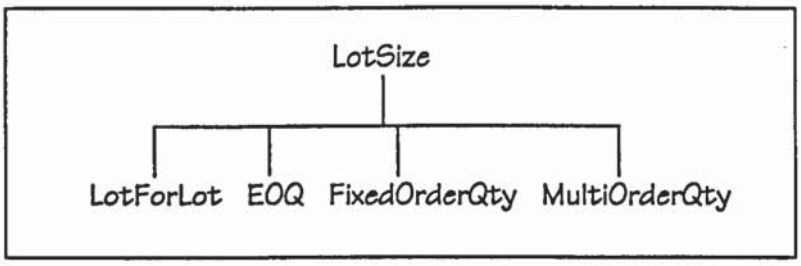


Figure 10.7: The lot sizing class hierarchy.

The parent class *LotSize* is an abstract class and a descendant of *tActivity*. The current descendant classes of *LotSize* include *LotForLot*, *EOQ*, *FixedOrderQty* and *MultiOrderQty* and provide the associated lot sizing policy. For example, an instance of the class *EOQ* will determine the batch size according to the economic order quantity. Similarly, a batch size determined according to a *FixedOrderQty* object will either be at the value of the fixed order quantity or the quantity suggested by the material planning process, whichever is greatest.

The properties which are common to all of the child classes are abstracted into the class *LotSize*; these properties are then inherited. This abstraction facilitates the expansion of the system and supports the reuse of code. For material planning purposes, all parts are prescribed a lot sizing policy; for example 'part0001' may be batched according to a lot-for-lot policy whereas 'part0002' may be based on an EOQ. This represents the situations found in actual companies. In WBS/Control, each *PartData* object includes a reference to its lot sizing policy; this is illustrated in Figure 10.6.

Figure 10.8 illustrates the 'orders' class hierarchy. There are currently three types of order available in WBS/Control: *MasterProductionSchedule*, *SuggestedOrder* and *WorksAndPurchaseOrder*. These classes were described in chapter 9 (Table 9.1). The common properties of these three classes are found in the parent class *AbstractOrder*.

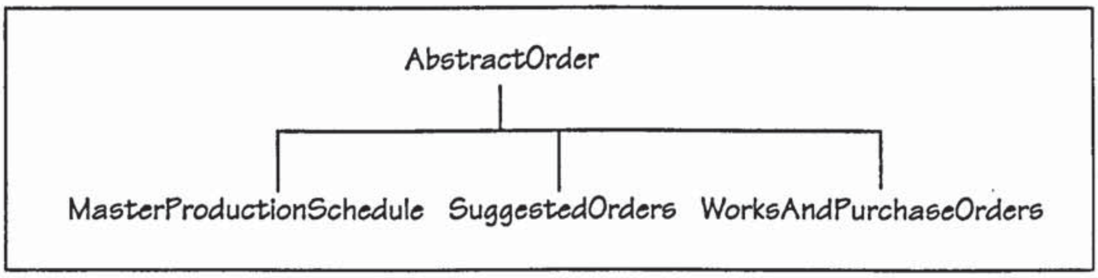


Figure 10.8: The 'orders' class hierarchy.

The implementation of each of the three types of order is similar to that of the *PartSet*, *PartList* and *PartData*. Figure 10.9 illustrates the situation for the future production programme, i.e. the MPS, represented by *Mps01*. The objects *MpsOrder01*, *MpsOrder02*, etc. represent the individual entries of the production programme (*Mps01*): recording the (MPS) order reference, end item number, (MPS) order quantity and due date. Access to the individual MPS entries is only possible via the interface of *Mps01*.

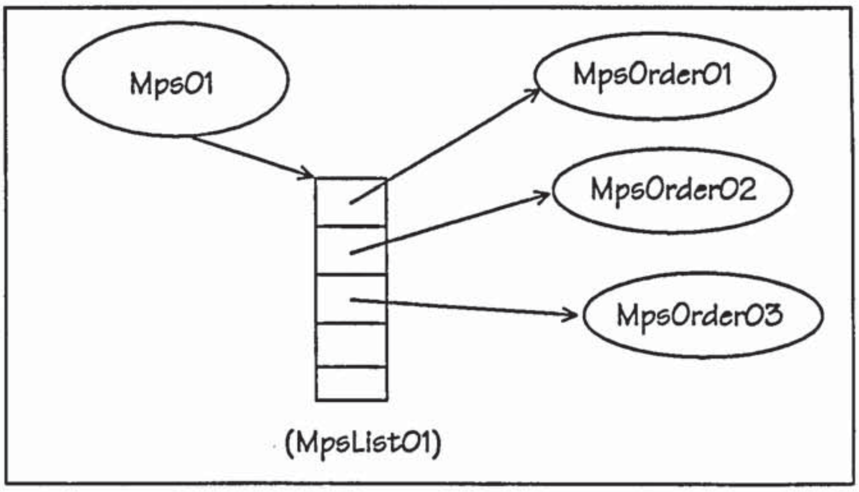


Figure 10.9: An MPS object.

10.4 Material planning activities

This section will describe how different manufacturing planning and control systems have been implemented in WBS/Control.

10.4.1 Low level classes

The concept of 'low level' classes was introduced in the previous chapter as the preferred approach to designing material planning classes. The application of low level classes supports the necessary flexibility in order to model different control systems; in essence different control systems are configured from (appropriate) low level classes. Figure 10.10 illustrates the material planning hierarchy and identifies a selection of the low level classes currently available. All of the low level classes are descendant from the class *AbstractMtlPlanning*. All of the classes are described in Appendix 1.

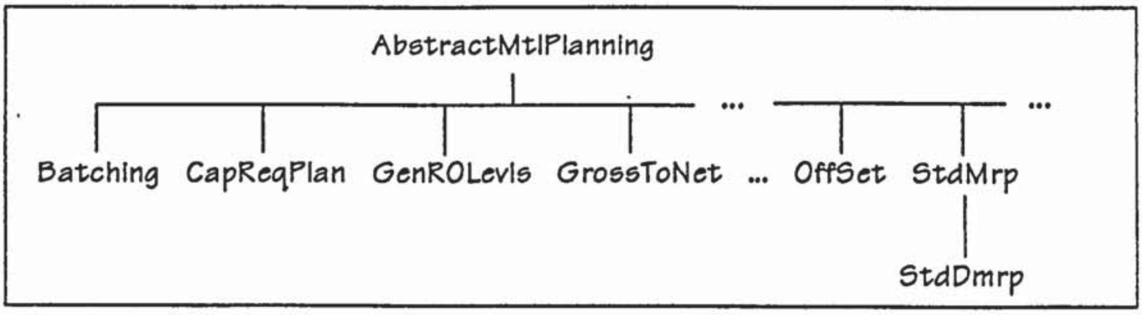


Figure 10.10: Selected material planning classes.

Each of the classes is self-contained and thereby independent of other low level classes. For example an object of the *GrossToNet* class would possess all of the necessary functionality to 'net' the current order quantity against the current on hand stock and update the particular 'suggested order' accordingly. Similarly an object of the class *InclDueIn* (i.e. include due in orders) increases the value of the on hand stock by the due in quantity on the date the order is expected. The information necessary to perform a particular process is obtained by messaging; this will be explained subsequently.

The encapsulation of the material planning activities in self-contained (independent) classes of this nature provides the important flexibility property through which different planning and control systems can be represented. Chapter 9 focused on the representation of MRP and ROP systems to illustrate this principle. The model of the MRP system was configured from the low level classes *GrossToNet, Batching, OffSet* and *IncDepDem* (i.e. explode demand for child parts); the ROP model was configured from *GenROLevls* (i.e. generate orders based on the reorder level value), *Batching* and *OnSet*. It will be evident that *Batching* is common to both interpretations.

The range of different low level class implementations has so far been discussed; the following section will explain 'how' different material planning systems can be configured from the low level classes.

10.4.2 The concept of a planning and control 'routing'

The availability of appropriate low level classes provides potential to represent different planning and control systems. However, this is only possible if a suitable mechanism exists by which the low level objects can be 'knitted' together to perform the planning and control activities.

The concept of a planning and control 'routing' has been applied to this problem. A direct analogy can be drawn with a manufacturing routing, i.e. the definition of the sequence of operations required to make the part along with the work centres involved in the operations. In the same way, a planning and control routing, or pc-routing, defines the sequence in which the planning and control processes are carried out as well as the information requirements.

In the previous chapter, Figure 9.3 illustrated the core MRP calculation whereas Figure 9.5 illustrated a low level object configuration of the MRP calculation. What was missing from Figure 9.5 is the mechanism by which the low level objects are configured; Figure 10.11 illustrates the implementation of the pc-routing mechanism for this particular example.

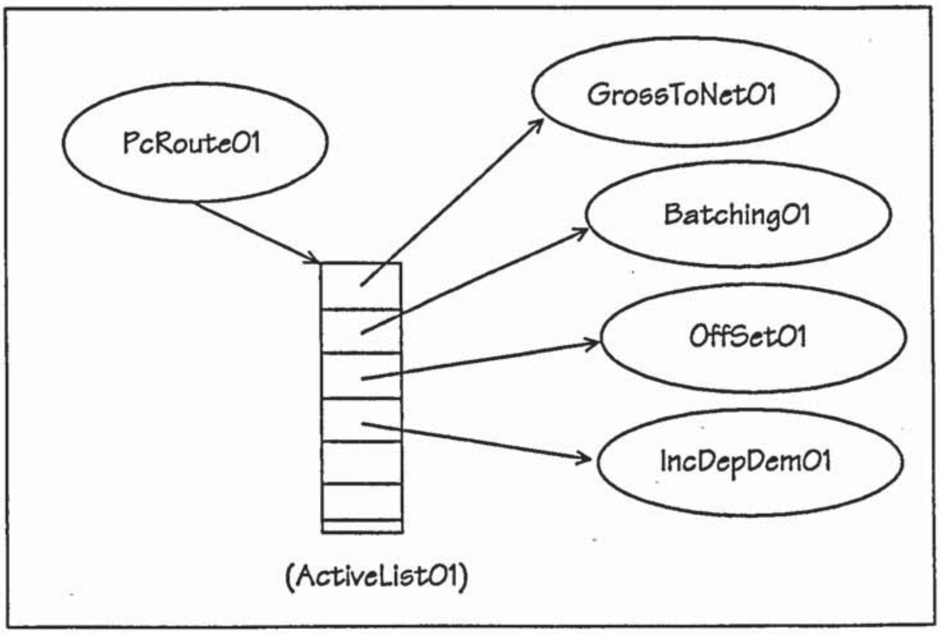


Figure 10.11: An example of a pc-routing implementation.

The object *PcRoute01* defines the pc-routing and is an instance of the class *PcRoute*. Each of the low level objects are attached to *PcRoute01* by a collection of pointers held on *ActiveList01*. When the material planning activity takes place within a simulation 'run', each of the low level objects of *PcRoute01* 'perform' their material planning activity in turn. The sequence is defined by the position of the object on the *ActiveList*, i.e. *GrossToNet01*, *Batching01*, *OffSet01* and *IncDepDem01*.

A modeller will construct a pc-routing, such as *PcRoute01*, through a dialog in the user-interface (building models will be discussed in more detail in the following chapter). In this way a modeller can configure different planning and control systems from the low level classes: no programming is required. The duration of the material planning activity is also set via this dialog. For example, *PcRoute01* could be set to take 3 hours.

The modeller is not restricted to one pc-routing per simulation model: any number of different planning and control systems are possible. The user is only restricted by the range of low level classes (currently) available. For example, *PcRoute01* may define a MRP-based system whereas *PcRoute02* could define a ROP system. In this way hybrid systems can be easily accommodated. Building a model which can represent different control systems is achieved through the application of different pc-routings.

In a model which includes a hybrid planning and control system, each pc-routing is attached to a group of parts (i.e. those which are defined in *PartSet* and *PartData*). For example, *PcRoute01* could be 'attached' to parts 'part0001' and 'part0002' whereas *PcRoute02* could be 'attached' to 'part0003' and 'part0004'. Consequently, the material planning activities for 'part0001' and 'part0002' are based on the low level objects defined in *PcRoute01*. Figure 10.12 illustrates this concept. This is how a 'runners, repeaters and strangers' model or, alternatively, an 'ABC' model can be constructed.

Furthermore, the flexibility of this approach means that a new control system design can be configured: providing the low level classes exist, they are simply added to a pc-routing in the defined sequence. This may be a single control system applied across all of the parts or a combination of two or more pc-routings.

Not only can hybrids be modelled but also transient behaviour. If, in the above example, changes in circumstances meant that 'part0003' should now be planned according to *PcRoute01*, this could be achieved by 'detaching' it from *PcRoute02* and 'attaching' it to *PcRoute01*. Equally, if a new control system has been devised for a company which will perform the material planning for all parts then this too can be accommodated without the need to build a completely new simulation model. The new pc-routing (e.g. *PcRoute03*) would be configured, all parts 'detached' from their current pc-routing and then 'attached' to *PcRoute03*. Consequently, the effects of transferring from one control system to another can be evaluated.

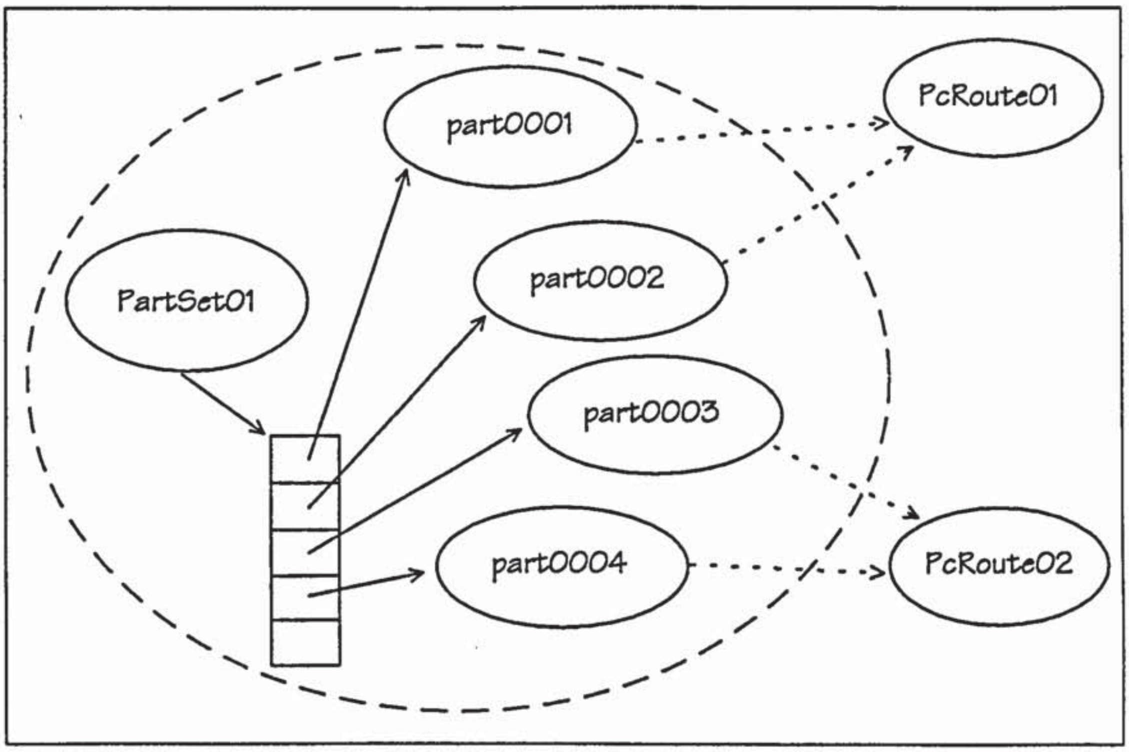


Figure 10.12: Linking parts to pc-routings.

The development of low level classes and the application of the pc-routing offers a completely different approach to modelling planning and control systems. Furthermore, the flexibility which is implicit in this approach supports a variety of modelling applications. The combination of low level classes and the flexibility of the pc-routing means that it is now possible to address the more fundamental design of planning and control systems.

10.5 'People' entities and skills

One of the aspects of AFS which sets it apart from other manufacturing simulators is the way in which 'people' entities are included in the simulation architecture. Ball (1994) suggests that a closer correspondence between the concepts found in reality and those found within the simulation software aids user understandability. Furthermore, the development of an object-oriented simulator is itself supported by this attribute.

This has led to the inclusion of operators, material handlers and cell leaders which take a more realistic role in the simulation model: for example, operators load, run and unload machines, material handlers use trucks to transport material and cell leaders pass production and transportation instructions to the respective operators. This implementation contrasts with other simulation tools whereby 'parts' take the active role and use machines and operators as resources.

Figure 10.3 included some of the classes which are used to represent 'people' entities in AFS. The class *Person*, a direct descendant of *tActivity*, has two child classes: *Operator* and *CellLeader*. These classes support the basic functionality associated with manufacturing personnel. For example, an object of class *Operator* can perform functions such as work according to a shift pattern, load, run and unload a machine. In order to perform more specific tasks operator objects are assigned 'skills'. For example, for operators to use a truck and move parts around the shop floor, they need a 'transportation skill'; similarly, an operator can only manage several machines if possessing the 'multi machine skill'. An operator can be assigned one or more skills. Put simply, operators and cell leaders are 'people' classes assigned with appropriate 'skills', this is illustrated in Figure 10.13. A selection of 'skills' classes are included in Figure 10.3: for example, the class *SingleMc* skill inherits the behaviour of *MachineOperation* skill which is a child of the class *AbstractSkill.*

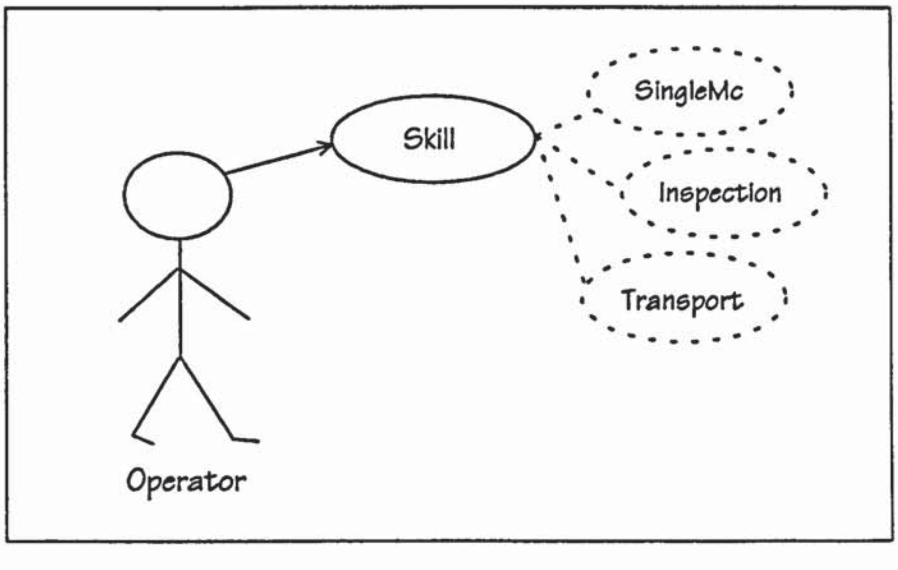


Figure 10.13: Representing 'people' in AFS.

10.5.1 'People' entities, skills and WBS/Control

The functionality included in AFS with respect to 'people' entities has been exploited by WBS/Control to extend the planning and control modelling capabilities. It is important to differentiate between the functionality included in AFS and its extension to the planning and control domain through WBS/Control. The new 'skills' which have been developed as part of WBS/Control mean that it is now possible to model far more than the material planning algorithms; the inclusion of 'policies, procedures and practices', i.e. elements of the planning and control infrastructure, is now feasible. It has been argued throughout this thesis that there is more to planning and control systems than their respective algorithms, whether or not they are computerised. In the past the inclusion of people and their role within the planning and control system(s) has rarely, if at all, been considered. WBS/Control is able to address this important omission.

The illustration which was used at the beginning of chapter 9 included 'people' entities such as a 'material planner', a 'master scheduler' and a 'stores person'. In essence, the inclusion of this functionality has been achieved by extending the 'skill' class hierarchy. The effort and programmed code to achieve this has been considerably reduced because of the development of the AFS architecture along object-oriented principles. It will be apparent from Figure 10.14 that the bulk of the new skills' functionality has been inherited; the development has concentrated upon what differentiates the planning and control skills from those already present.

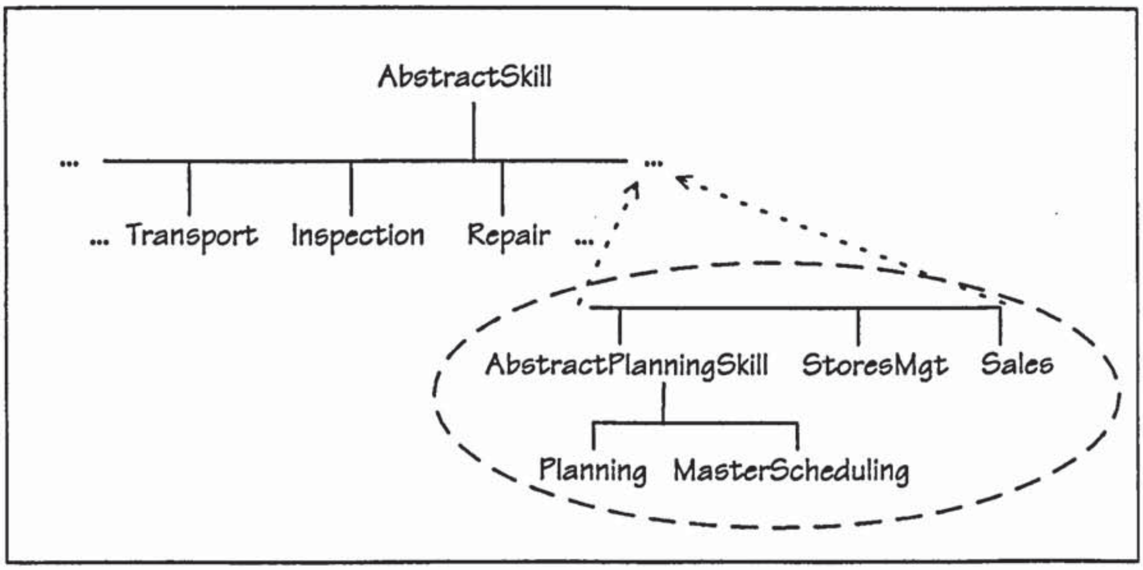


Figure 10.14: The inclusion of planning and control skills in an existing hierarchy.

The skills which have been developed enable an operator to perform planning and control activities; the activities themselves are defined in pc-routings. This combination has led to a more simplified arrangement: the skill classes developed for WBS/Control encapsulate the infrastructure-like functionality whereas the pc-routing encapsulates the planning and control functionality. The user configures a skill to an operator through a dialog in the user-interface; this will be discussed in the following chapter.

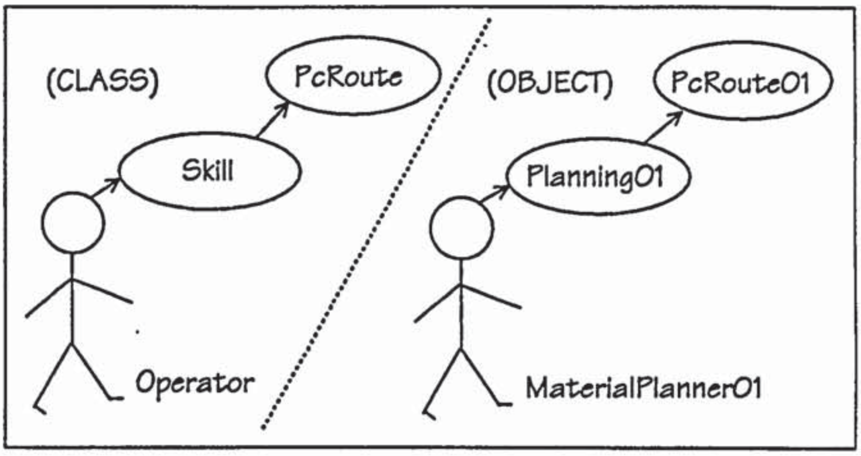


Figure 10.15: The relationships between operator, skill and pc-routing.

Figure 10.15 illustrates the relationships between operator, skill and pc-routing. Both the classes and an object implementation are included. The pc-routing (i.e. *PcRoute01*) is configured to perform the appropriate material planning tasks, for example ROP control. An object of class *Planning* skill (i.e. *Planning01*) is configured to an instance of *Operator* (i.e. *MaterialPlanner01*); *PcRoute01* is then attached to Planning01. This enables *MaterialPlanner01* to perform a material planning activity based on ROP control.

Within the implementation of the (*AbstractPlanning*) skill, the timing and frequency of the planning activity is set: for example, *MaterialPlanner01* could be set to determine the material requirements (based on ROP control) every day at 09.00 hrs or, alternatively, weekly at 08.30 hrs. This planning would be based on the state of the factory and information systems at the prescribed time. Also included within this implementation is the capability to confirm suggested orders and release works orders to the shop floor. This functionality has been implemented through simple control logic: if the launch date falls between 'today's' date and the date when the material planning next takes place, the order is confirmed and released. More complicated procedures can be included in a new skill.

The *StoresMgt* skill further demonstrates the potential to include elements of the planning and control infrastructure. Essentially, this skill books-in completed works orders and updates the inventory system appropriately; this is based upon the receipt of parts at the physical stores. The current implementation determines when the event takes place as well as the frequency. The duration of the activity is based upon the number of items to book-in; this is similar to the situation of an operator processing a batch of parts on a machine. A policy-like decision can be included in the configuration of the skill: whether or not the *StoresPerson* books in all of the 'goods received notes' in one go or in batches of, for example, 10.

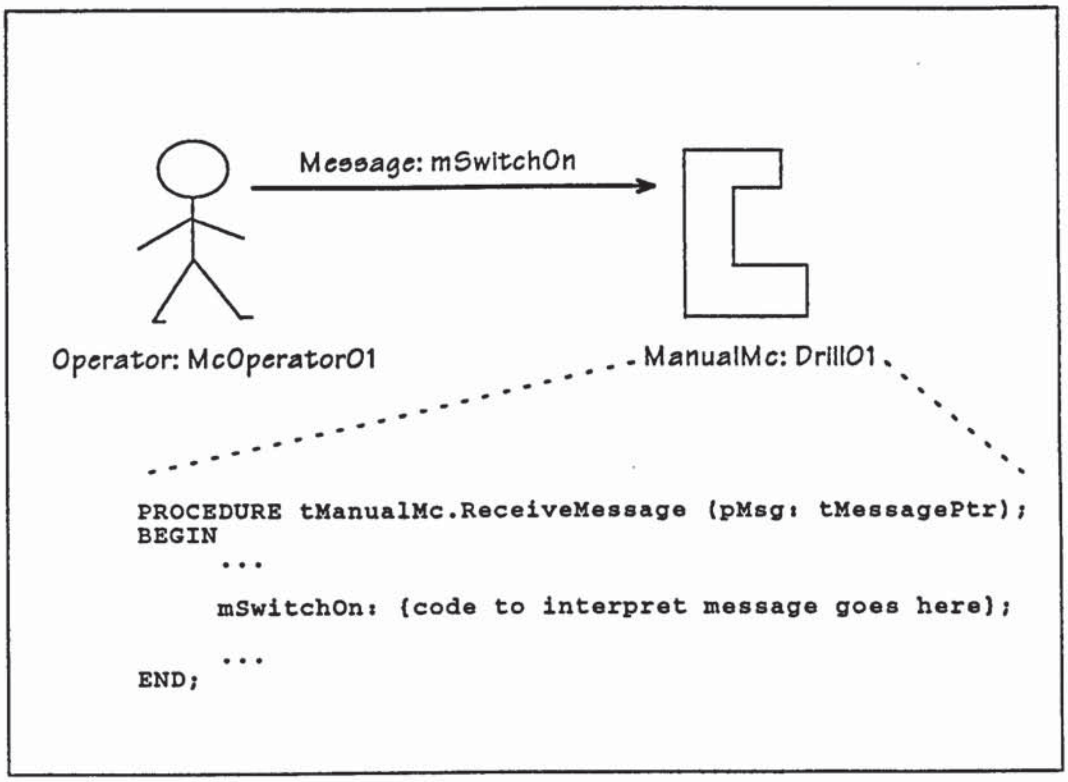
The remainder of skills currently available within WBS/Control are described in Appendix 1 and documented within the Borland Pascal code of the software file.

10.6 Messaging

WBS/Control fully exploits the message passing, or messaging, functionality provided through the AFS architecture: communication between objects is based on messaging. The implementation of this functionality enforces the object-oriented property of polymorphism; in other words objects can communicate with minimal knowledge of one another.

The mechanism through which this is achieved is the *tMessage* class: every message is encapsulated in a message object, i.e. the message is passed as an object. A generic messaging mechanism supports the passing of messages between different objects. This functionality is provided through the core class *tActivity*: every class, therefore, which needs to be able to both 'send' and 'receive' messages must be a child of *tActivity* and thereby inherit the functionality. For example, because *Person* and *OperationStage* are both descendant of *tActivity*, *Operator* and *ManualMc* are able to communicate via message passing (see Figure 10.3).

The sending and receiving of messages adheres to a common pattern. The 'sender' object first of all 'bundles' the message together in the form of a message object and then passes this object to the 'receiver' object(s). In fact a (reference) pointer to the message object is passed and thereby allows access to the message. All objects capable of receiving messages possess a method called 'ReceiveMessage'; this method is responsible for receiving all the messages of a class and contains the code necessary to interpret the message and take the appropriate action. This process is illustrated in Figure 10.16 .



```
PROCEDURE tManualMc.ReceiveMessage (pMsg: tMessagePtr);
BEGIN
     ...
     mSwitchOn: {code to interpret message goes here};

     ...
END;
```

Figure 10.16: Messaging between objects.

Messaging is crucial to support polymorphism: provided objects can send and receive a particular message they can communicate without formal knowledge of one another. For example, the operator (*McOperator01*, above) could send the same 'switch on' message to a manual machine or an automatic machine; the interpretation of the message is left to the receiving object. Similarly, a *StoresPerson* could send the same 'RequestStockData' message directly to a *StockData* object or to a 'computer' object within which the inventory system resided. The *StoresPerson* does not need to know anything about the receiver object, not even 'who' or 'what' they are sending the message to. This property offers great flexibility in model construction. (The process of configuring messages and objects will be discussed in chapter 11.)

The benefits of polymorphism are also realised within the implementation of the pc-routing. As Figure 10.17 illustrates, the same 'PerformMaterialPlanning' message is sent to all of the low level objects configured in the pc-routing: each of the objects reacts to, or interprets, the message accordingly. This becomes especially relevant when new classes are added to the set of low level classes: the same message can also be sent to the new object without modification to existing object code. This property supports the extension of the object library: to use the 'building block' analogy, new 'building blocks' can be added with the minimum of disruption to the existing set.

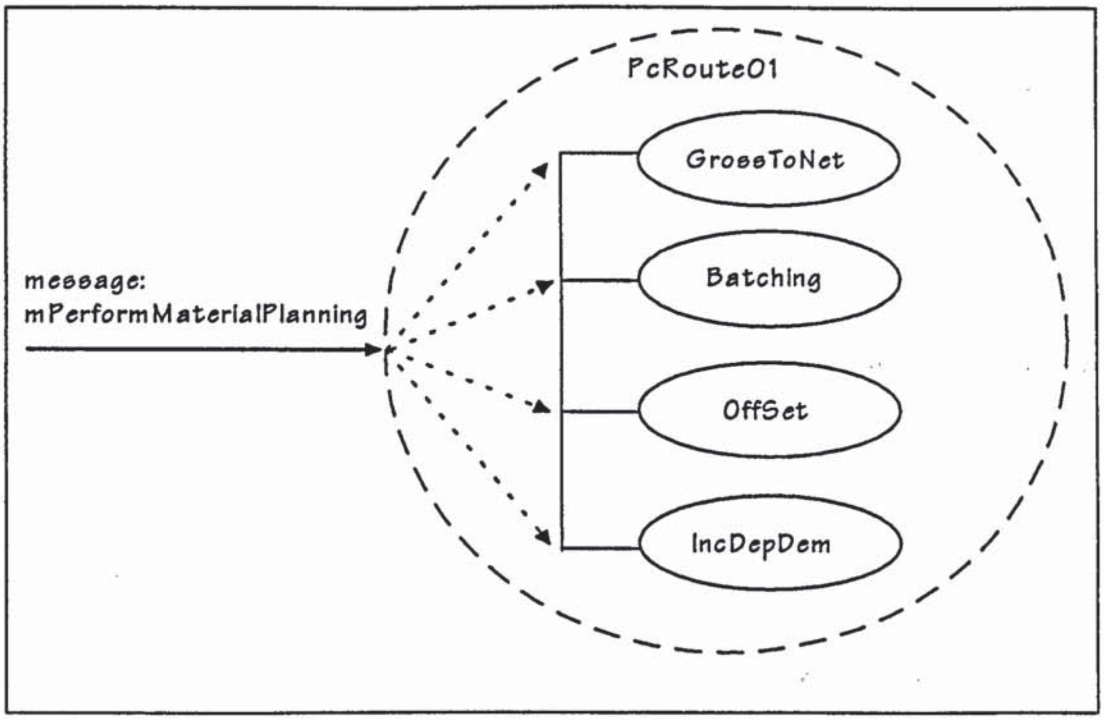


Figure 10.17: Polymorphism and the pc-routing.

10.7 Summary

This chapter has discussed the implementation of the planning and control class library WBS/Control. The relationship between AFS, the host simulation architecture, and WBS/Control has been established. This implementation has introduced a novel approach to the representation of manufacturing control systems, namely the low level class architecture and the concept of a pc-routing. The flexibility which results means that it is now possible to model planning and control systems from a more fundamental design perspective. The inclusion of 'people' entities not only extends the richness of the modelling environment but allows a more complete analysis of the designed system to be made.

11. The application of WBS/Control

This chapter describes the application of WBS/Control from two perspectives: building simulation models and extending the current functionality. The first perspective is that of the user, or modeller, whilst the second is one of development; the emphasis is on model configuration. The configuration of models through the different dialogs of the user-interface will be explained: the data driven attributes of manufacturing simulators are fully exploited by the Advanced Factory Simulator (AFS) and WBS/Control - no programming is required to build models. This chapter will demonstrate that it is possible to model properly planning and control systems through the application of an object library. The functionality offered by WBS/Control will increase as new and modified classes are developed. The extensibility of the class library will also be demonstrated.

11.1 The user-interface

In many ways the user-interface provides the key to the successful use of any software tool and the situation is no different in the case of an object-oriented manufacturing simulator. It is through the user-interface that a user (or modeller) configures, edits and runs a simulation model; the experimental results are also available through the user-interface. In order to reduce the time, effort and skill required to build simulation models, the design of the AFS user-interface has employed terminology and concepts found in the real world (Ball, 1994); this approach has been continued in the development of WBS/Control related dialogs.

In terms of interaction between user and software, graphical user-interfaces can be considered to be the norm; the growth in Microsoft Windows and Windows based applications are typical examples. The development of the AFS user-interface provides a Windows-like environment, through the application of Microsoft Visual Basic. (All commercial software used in the development has been referenced in Appendix 3.) This discussion begins with the dialogs specific to the AFS architecture and functionality.

11.1.1 AFS specific dialogs

The development of the user-interface has enforced the data driven principle of manufacturing simulators: the modeller does not have to resort to any form of programming in order to build, run and analyse simulation models. Dialogs provide the means to enter and edit data. The user-interface has been divided into two types of dialog: those which are specific to some form of functionality, for example a machine or operator dialog, and those which are common across simulation models, for example results display and simulation set-up and run. Figure 11.1 illustrates the structure of the user-interface.

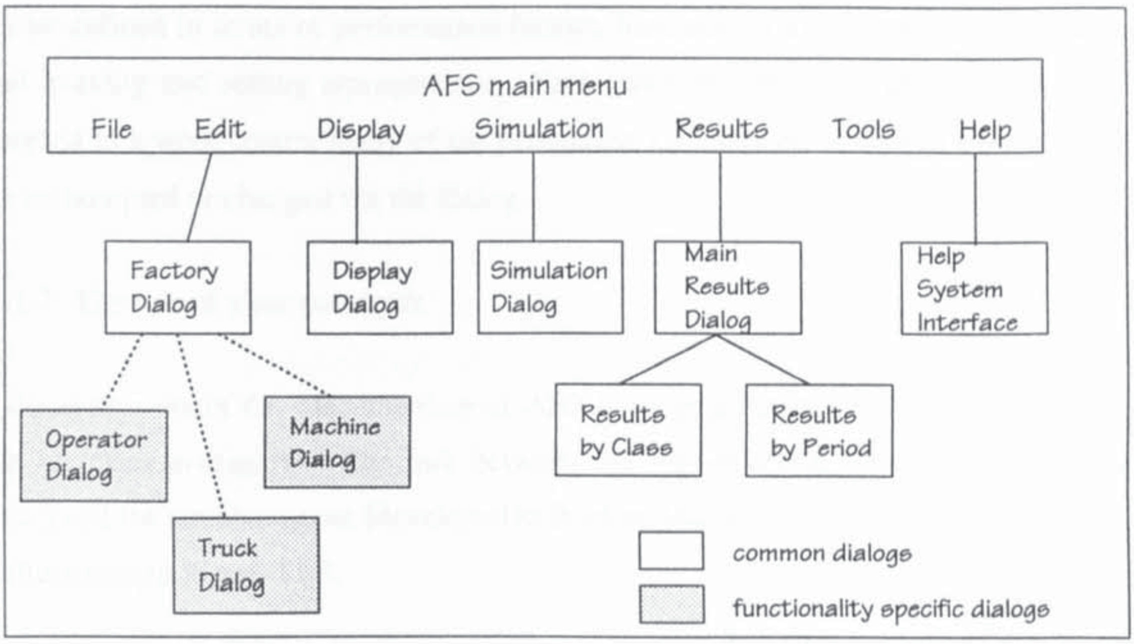
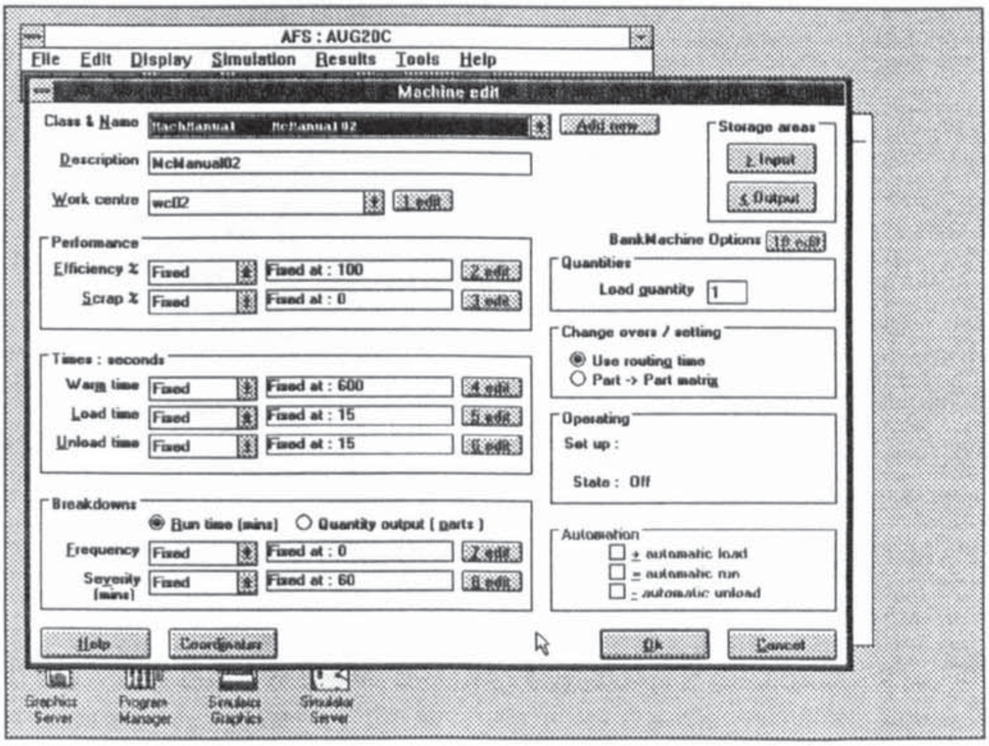


Figure 11.1: The structure of the user-interface of AFS (after Ball, 1994).



Class & Name: MachManual McManual02

Description: McManual02

Work centre: wc02

Performance
Efficiency %: Fixed — Fixed at : 100
Scrap %: Fixed — Fixed at : 0

Times : seconds
Warm time: Fixed — Fixed at : 600
Load time: Fixed — Fixed at : 15
Unload time: Fixed — Fixed at : 15

Breakdowns
● Run time (mins) ○ Quantity output (parts)
Frequency: Fixed — Fixed at : 0
Severity (mins): Fixed — Fixed at : 60

BankMachine Options

Quantities
Load quantity: 1

Change overs / setting
● Use routing time
○ Part -> Part matrix

Operating
Set up :
State : Off

Automation
☐ + automatic load
☐ = automatic run
☐ - automatic unload

Figure 11.2: The manual machine dialog.

The development of dialogs in this way supports an object-oriented view: the data relating to the functionality is 'encapsulated' in the associated dialog. For example, the data relating to setting the simulation time, period length, run length and so on are all found within the simulation dialog(s). This minimises the scope of changes to the user-interface if changes have been made to the core.

An example AFS dialog is illustrated in Figure 11.2. This shows that a manual machine can be defined in terms of performance factors, load/unload times, breakdown pattern, load quantity and setting arrangements. Each machine (object) is given a name and assigned to a work centre; many of the parameters are supplied as default values which can be accepted or changed via the dialog.

11.1.2 The role of class managers

A discussion about the user-interface of AFS is an appropriate point to introduce the role of 'class managers'. The link between the user-interface (developed in Visual Basic) and the simulator core (developed in Borland Pascal) is made via class managers, as illustrated in Figure 11.3.

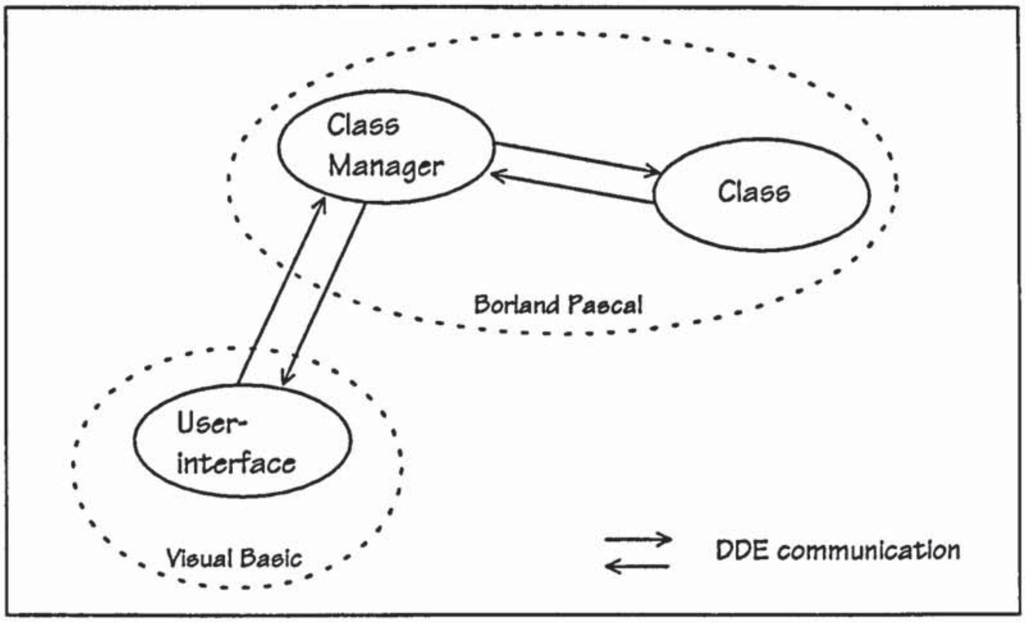


Figure 11.3: The relationship between the user-interface, class managers and classes.

Consider, for example, the manual machine dialog illustrated in Figure 11.2. When the dialog is accessed, the values of the data attributes for the object *McManual01* are displayed; these values are retrieved from the object. In other words, the user-interface (i.e. the dialog) requests the information from the class manager which then 'gets' the information and passes it on to the dialog. Similarly, to change (or edit) the value of a data attribute, the new value, entered via the dialog, is passed on and updated in the object via the class manager. This communication between the applications is made possible by the DDE (Dynamic Data Exchange) messaging mechanism. DDE is a Windows standard inter-application communications protocol which allows applications to communicate using predefined standards.

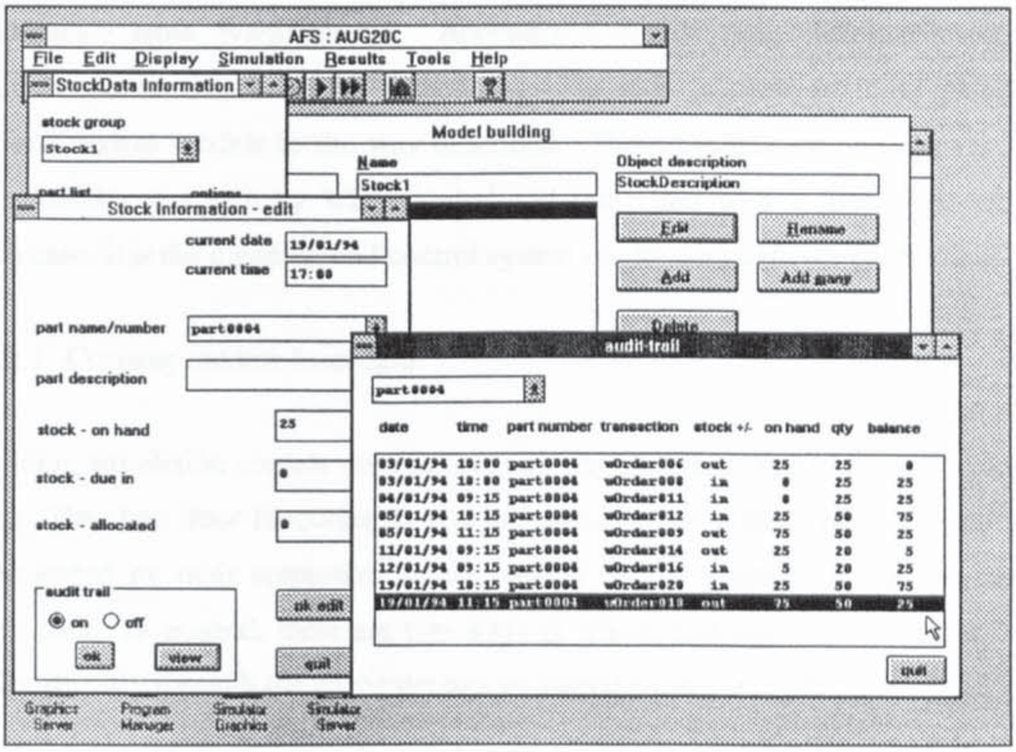
All classes, therefore, have a class manager. In addition to data retrieval and modification, a class manager also provides the mechanisms to create and destroy objects, for loading data from text files, for managing the load and save process and for the preparation of the results data. The provision of class managers by the AFS architecture means that there is a distinct split between the data management mechanisms and the simulation functionality; the class manager, itself a class, encapsulates the data management functionality associated with a class. For example, the *ManualMc* class manager provides all of the data management mechanisms for the class *ManualMc*, whereas *ManualMc* encapsulates all the functionality associated with a manual machine entity.

11.1.3 WBS/Control related dialogs

The style of the AFS user-interface has been continued by WBS/Control. The user builds, or configures, models of a planning and control systems through a Windows-based interface. The dialogs have been developed so that the user is able to build and edit models with relative ease. Whilst many of the dialogs of WBS/Control will be introduced in the context of the subsequent discussion, two examples will be included here. With respect to the structure of the AFS user-interface (see Figure 11.1), the WBS/Control dialogs are to be found within the 'Edit | Factory' option.

Figure 11.4 illustrates the edit dialog for the *StockData* information class; the expected data is included, i.e. on hand stock, due in orders, as well as stock allocated to orders. This dialog also provides access to the stock audit trail for the part number.

Figure 11.5 illustrates the dialog associated with works orders and purchase requisitions. The overview dialog (in the background) lists all orders along with their current status; the foreground 'order edit' dialog is for the highlighted order number. Orders will be generated through the material planning option(s) (see pc-routings) or, alternatively, a set of orders can be loaded into the model from a text file. In both cases, an order can be modified through the order edit dialog.



stock group: Stock1

Name: Stock1

Object description: StockDescription

current date: 19/01/94

current time: 17:00

part name/number: part0004

part description:

stock - on hand: 25

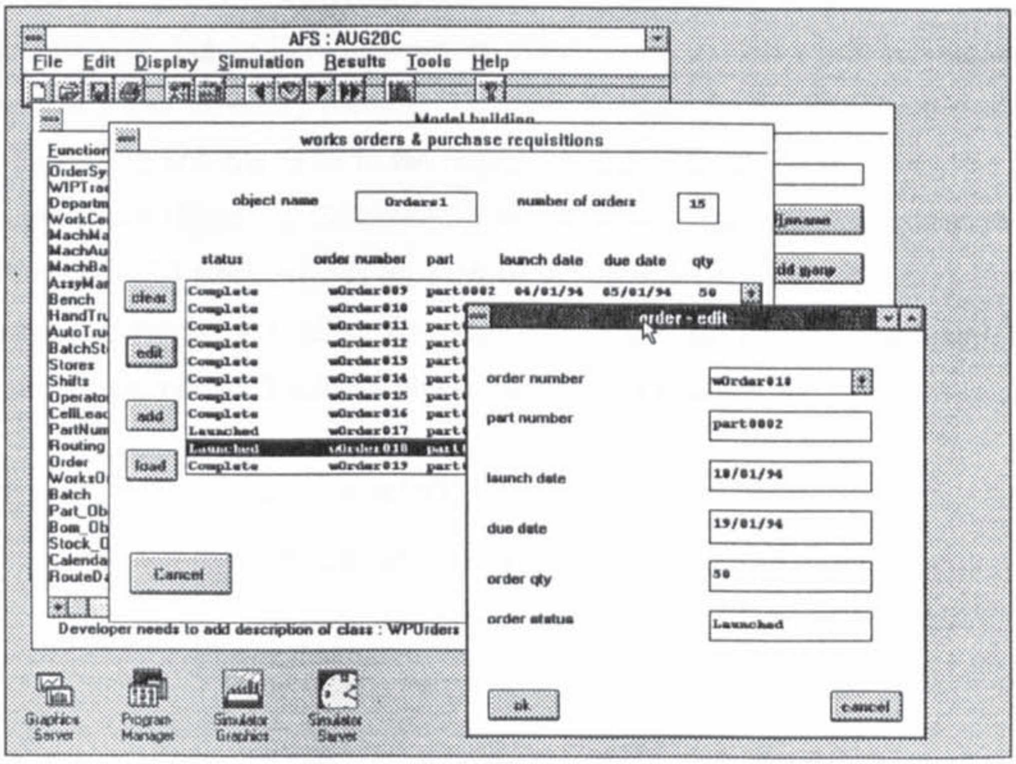
stock - due in: 0

stock - allocated: 0

part0004

date	time	part number	transaction	stock +/-	on hand	qty	balance
03/01/94	10:00	part0004	wOrder006	out	25	25	0
03/01/94	10:00	part0004	wOrder008	in	0	25	25
04/01/94	09:15	part0004	wOrder011	in	0	25	25
05/01/94	10:15	part0004	wOrder012	in	25	50	75
05/01/94	11:15	part0004	wOrder009	out	75	50	25
11/01/94	09:15	part0004	wOrder014	out	25	20	5
12/01/94	09:15	part0004	wOrder016	in	5	20	25
19/01/94	10:15	part0004	wOrder020	in	25	50	75
19/01/94	11:15	part0004	wOrder018	out	75	50	25

Figure 11.4: The edit dialog of *StockData* class.



object name: Orders1

number of orders: 15

status	order number	part	launch date	due date	qty
Complete	wOrder009	part0002	04/01/94	05/01/94	50
Complete	wOrder010	part			
Complete	wOrder011	part			
Complete	wOrder012	part			
Complete	wOrder013	part			
Complete	wOrder014	part			
Complete	wOrder015	part			
Complete	wOrder016	part			
Launched	wOrder017	part			
Launched	wOrder018	part			
Complete	wOrder019	part			

Developer needs to add description of class : WPOrders

order number: wOrder018

part number: part0002

launch date: 18/01/94

due date: 19/01/94

order qty: 50

order status: Launched

Figure 11.5: The works and purchase orders dialog.

11.2 Model Construction

The following sections will discuss how planning and control system models can be constructed from WBS/Control. Appendix 5 details three different planning and control system models which demonstrate that it is possible to build planning and control system models in the way described. These models are constructed from the functionality provided by WBS/Control and integrated with a shop floor model - in each case, it is the planning and control system model which drives production.

11.2.1 Creating models from data

Building simulation models does not require any programming to be performed by the user. The shop floor functionality and the planning and control system functionality is represented by their respective object counterparts and these objects are configured from data. In general, there are two ways in which data can be entered into a model: either directly through the user-interface or loaded from a text file.

If data is loaded from a text file, the user must organise the required data into the required format. Although it is the responsibility of the class manager to organise the 'text upload' from a text file, the data must be organised so that it can be 'read' by the class manager. Figure 11.6 illustrates an example of a text file which will create a stock system and four inventory records. The class manager for the *StockData* class expects the data in the text file to be in the format defined. The first line defines the name of the *StockData* object (i.e. *StockData1*) and an object description; the subsequent lines define the initial stock records for each of the parts defined in *StockData1* in terms of 'on hand', 'due in' and 'allocated' quantities. The class manager reads each line and creates an appropriate *StockItem* object for each part number.

```
StockData1 StockDescription
StockData1 DATA part0001 25 0 15
StockData1 DATA part0002 30 10 5
StockData1 DATA part0003 0 0 0
StockData1 DATA part0004 12 25 15
```

Figure 11.6: Creating objects from data files - an example text file.

If objects are created from text files, the data attributes can be modified directly through the dialog. This is particularly relevant if objects are created with default values which may be the case if the data is unavailable at the time.

11.2.2 Graphical displays

The AFS architecture provides the opportunity to create a graphical display of a model. The display is responsible for showing the physical position and size of objects through their icon representation and can, therefore, be arranged to represent the layout of the 'real' system. Graphical displays are useful to support the verification of models as well as providing the ability to view a simulation 'run'. A graphical display of an example model is shown in Figure 11.7 and illustrates the different shop floor functionality available through AFS.

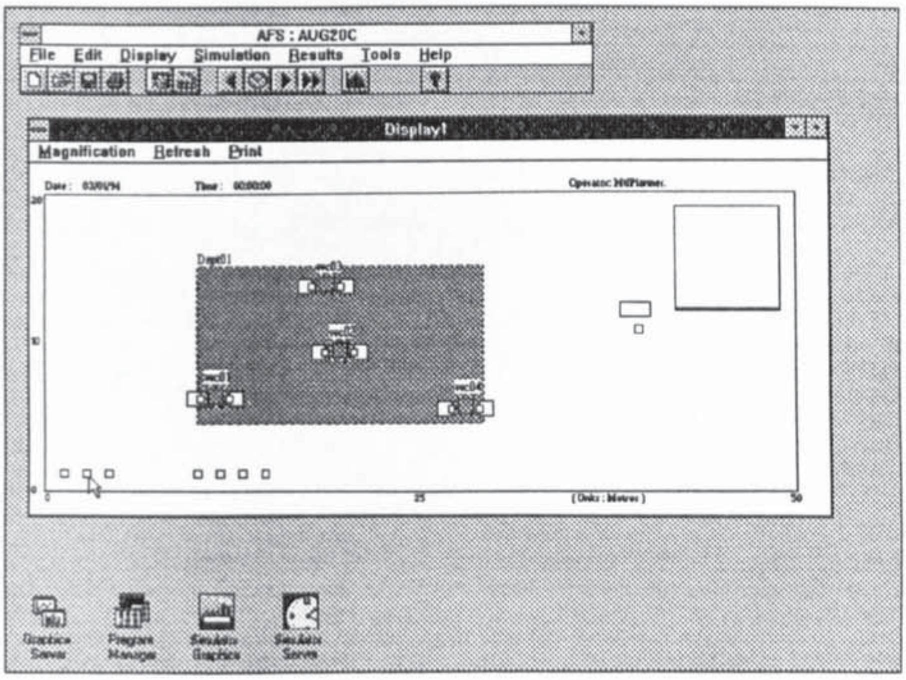


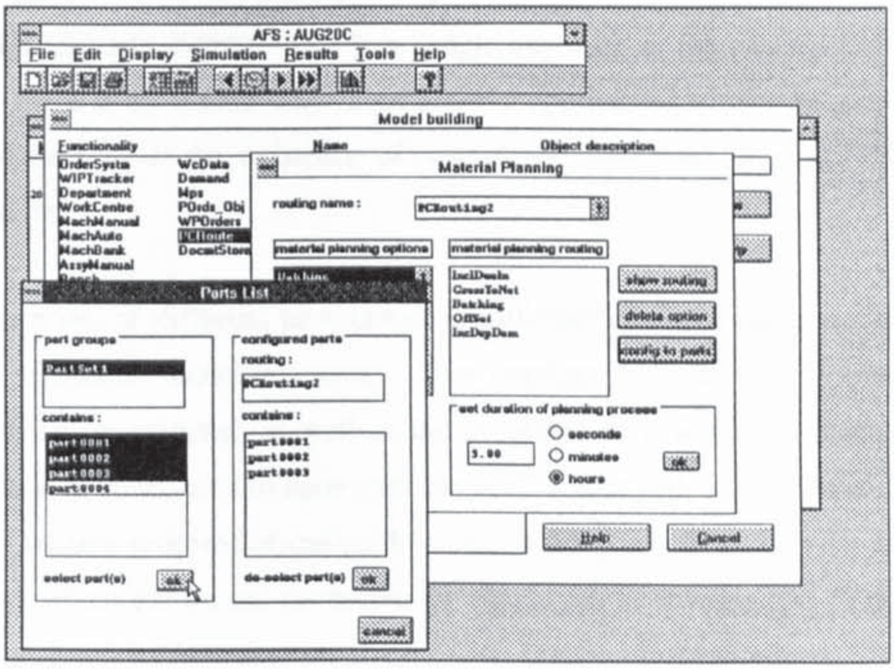
Figure 11.7: An example graphical display.

The department (*Dept01*) contains four work centres, *wc01*, *wc02*, *wc03* and *wc04*, each containing a single work station: *wc01*, *wc02* and *wc03* containing manual machines and *wc04* containing a manual assembly machine. Operators have been assigned to each work station; each operator will be assigned the appropriate skill, for example *Op01* can operate *McManual01* and is assigned a 'single machine' skill, whereas *Op04* has been assigned the 'assembly' skill in order to operate the *AssyManual01* machine. Also included in the model are central stores, a material handler, a hand truck, a cell leader, a material planner, a stores person and a sales person.

The dialogs of the displayed objects can be accessed through the graphical display by 'double clicking' on the object icon. For example, double clicking on the central stores icon opens the *Stores* dialog which identifies the current status, i.e. what parts are currently in store and in what quantity.

11.2.3 Configuring a pc-routing

The previous chapter introduced the concept of a pc-routing as the means by which low level objects can be 'knitted' together to perform material planning and control activities. This section will describe how the shop floor model can be extended through the integration of a planning and control model. Figure 11.8 illustrates the associated dialogs.



routing name : PCRouting2

material planning routing:
IncDueIn
GrossToNet
Batching
OffSet
IncDepDem

set duration of planning process: 3.00 ● hours

part groups: PartSet1
contains : part0001, part0002, part0003, part0004

configured parts
routing : PCRouting2
contains : part0001, part0002, part0003

Figure 11.8: Configuring a pc-routing, *PcRouting2*.

The pc-routing dialog(s) are accessed from the 'Edit | Factory' option. The opening dialog lists all the available low level classes from which a pc-routing can be configured. If a low level class is selected, an object of this class is created and attached to the routing. In the above example the low level objects of *PcRouting2* include *IncDueIn*, *GrossToNet*, etc. Also set within this dialog is the activity duration, i.e. how long the material planning activity will take. The pc-routing illustrated represents a MRP approach and has been set to take 3 hours.

The parts whose planning and control is determined by a pc-routing are configured in the (nested) dialog: this is also shown in Figure 11.8. In the example, the material requirements for 'part0001', 'part0002' and 'part0003' are determined by *PcRouting2*; the requirements for 'part0004' will either be derived by another pc-routing or not included in the planning and control processes. The modelling of hybrid systems is not difficult to achieve: a pc-routing is configured and linked to the parts involved. Similarly, it is easy to move a part from the control of one pc-routing to another: the part is de-selected from the original pc-routing and configured to either an existing or completely new pc-routing. The implementation of the pc-routing concept has provided an extremely flexible approach to modelling planning and control systems.

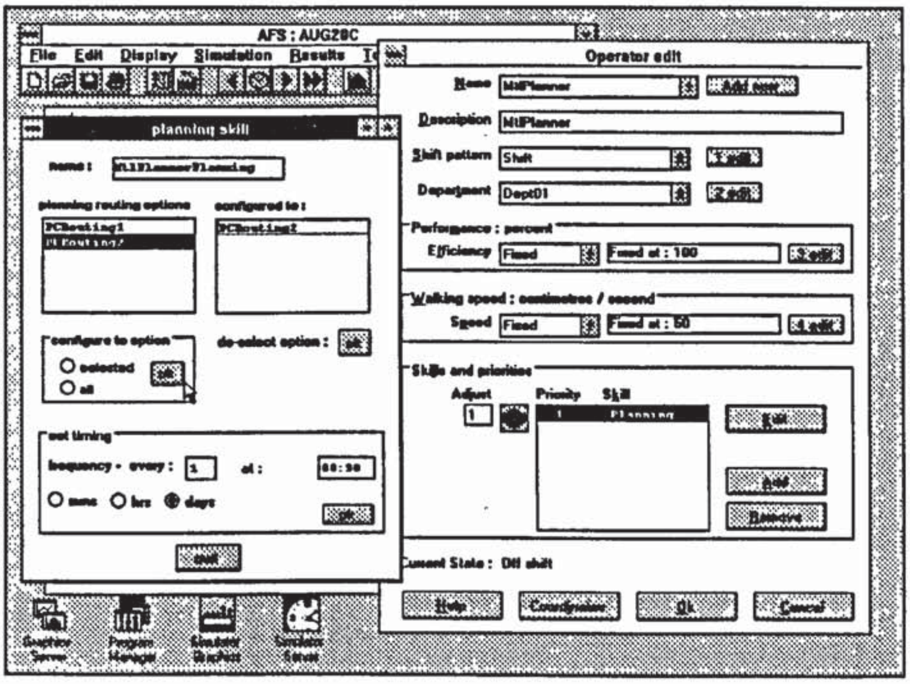
Different pc-routings can be developed for different levels in the planning hierarchy. For example the low level class *DemToMps* translates the product demand into a master production schedule (MPS). This is a simple approach to the construction of a MPS, however there is no reason why more specific approaches cannot be modelled. The main requirement is the existence of appropriate low level classes. This will be discussed later.

The application of different pc-routings at different levels in the planning hierarchy offers considerable modelling scope. The interface between the different planning levels can be investigated, as well as the processes as a whole. For example a pc-routing may be configured to determine the MPS whose orders are then included in the material planning process determined by one or more pc-routings; because of the timing and delays which can be set, the knock-on effects can be investigated. The role of the pc-routing also extends to determining the shop floor control procedures.

11.2.4 The inclusion of 'people' and skills

The introduction of 'people' entities within a model follows the same principles as those required to include shop floor operators, for example a material handler. In fact, planning and control personnel are simply operators assigned with the appropriate 'skill'. It is the skill class which encapsulates the functionality associated with the respective task. In the case of WBS/Control, the skills extend the modelling capability beyond that of representing a material planning algorithm.

Figure 11.9 illustrates how an operator is assigned a 'planning' skill. The object *MtlPlanner* is an instance of the class *Operator*. The default skill attached to an operator is *SingleMach* (i.e. single machine); this is removed via the dialog and the *Planning* skill is added. This skill enables *MtlPlanner* to perform the material planning activities defined on an attached pc-routing; in the example given, *MtlPlanner* will action *PcRouting2*. The timing and frequency of when this activity takes place is also set within this dialog; for example daily at 08.30 hrs. It was also explained in chapter 10 that this skill supports the release of orders to the shop floor. The combination of timing, frequency and duration (set with the pc-routing) provides the facility to properly integrate the material planning (and control) processes within the simulation. This facility is more realistic than, for example, releasing a work-to-list or schedule to the shop floor at the start of the simulation 'run'.



name : MtlPlannerPlanning

planning routing options: PCRouting1, PCRouting2

configured to : PCRouting2

configure to option: ○ selected ○ all

de-select option :

set timing

frequency - every : 1 at : 08:30

○ mins ○ hrs ● days

Name: MtlPlanner

Description: MtlPlanner

Shift pattern: Shift

Department: Dept01

Performance : percent

Efficiency: Fixed Fixed at : 100

Walking speed : centimetres / second

Speed: Fixed Fixed at : 50

Skills and priorities

Adjust	Priority	Skill
1	1	Planning

Current State : Off shift

Figure 11.9: Assigning a 'planning' skill to an *Operator*.

11.2.5 Message configuration

Objects communicate by message passing, consequently the desired 'message links' need to be configured at the model build stage: for example, a machine will need to be assigned to an operator. The message configuration is achieved through a mechanism provided by the AFS architecture. This is a powerful mechanism which allows classes (and objects) to be linked without formal knowledge of one another, the only constraint being that the 'receiving' class must be able to understand (i.e. receive) the message. The relationships between different classes are based, therefore, on their associated 'send' and 'receive' messages; these messages are authorised at class level. Objects of classes can only be configured if the links have been authorised at class level. Figure 11.10 illustrates the relationships between the *Operator* class and the WBS/Control classes *WorksAndPurchaseOrders* and *StockData.*

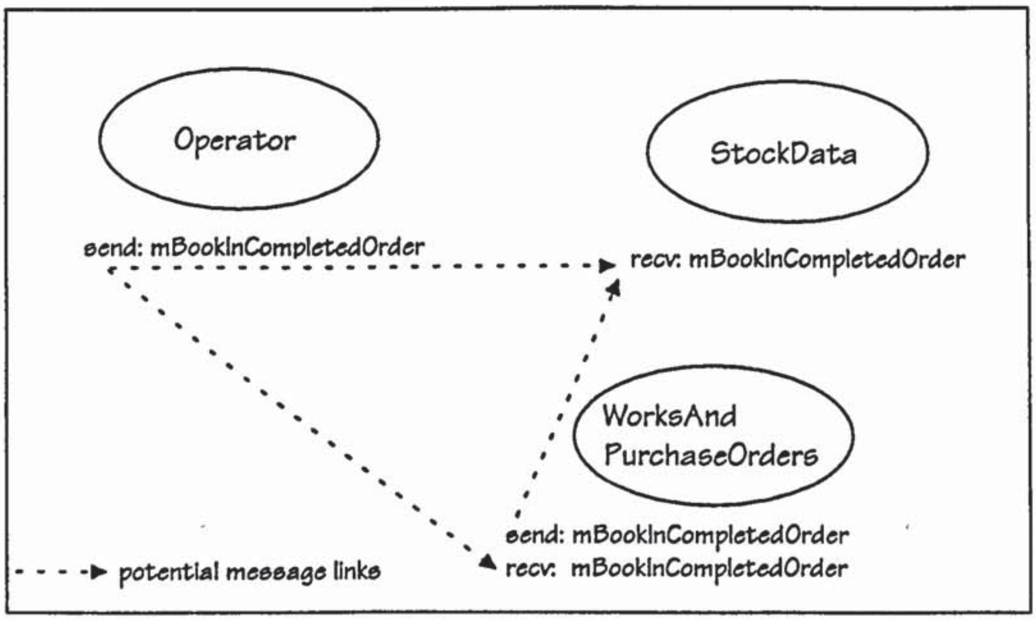


Figure 11.10: Identifying class relationships through their 'send' and 'receive' messages.

One of the messages that the class *Operator* can send is 'mBookInCompletedOrder'; this message can be received by the classes *WorksAndPurchaseOrders* and *StockData.* If these classes are linked (i.e. authorised) then communication can take place between them, provided two criteria are satisfied: objects of these classes exist and the objects themselves have been authorised to communicate. The linking of both classes and objects is achieved through the dialog found within the 'Edit | Configuration' option of the user-interface.

Figure 11.10 further indicates that the class *WorksAndPurchaseOrders* can also send the message 'mBookInCompletedOrder'. The potential is, therefore, for two types of communication: *Operator* to *WorksAndPurchaseOrders* and *StockData* or, alternatively, *Operator* to *WorksAndPurchaseOrders* and from *WorksAndPurchaseOrders* to *StockData.* This represents two potential modelling scenarios. The first is where an operator books in a completed order and then updates the inventory system accordingly. The second scenario is where the operator books in a completed order which 'automatically' updates the inventory system. The configuration of classes and objects through the user-interface is illustrated in Figure 11.11.

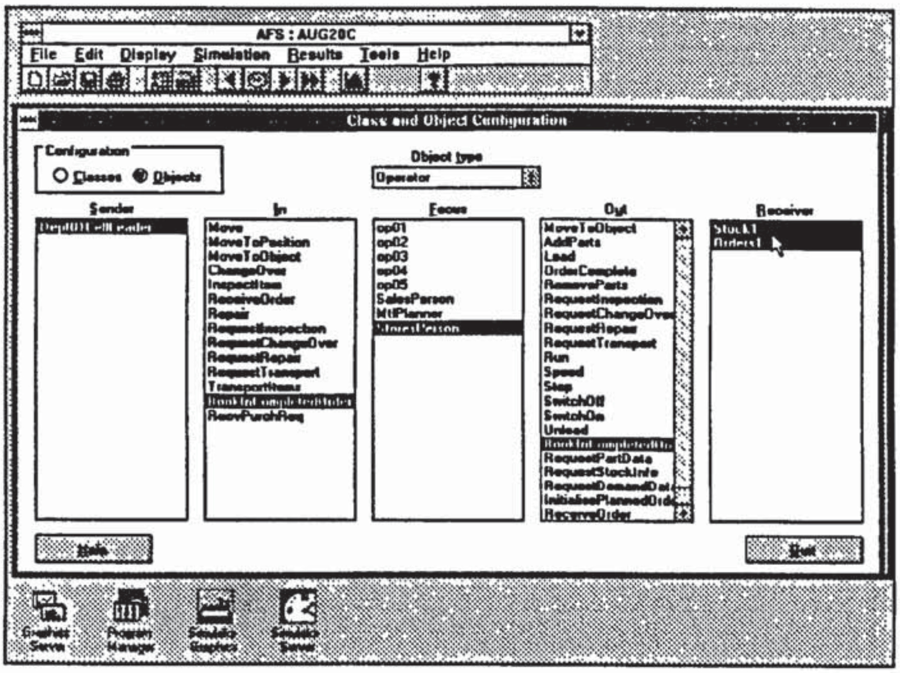
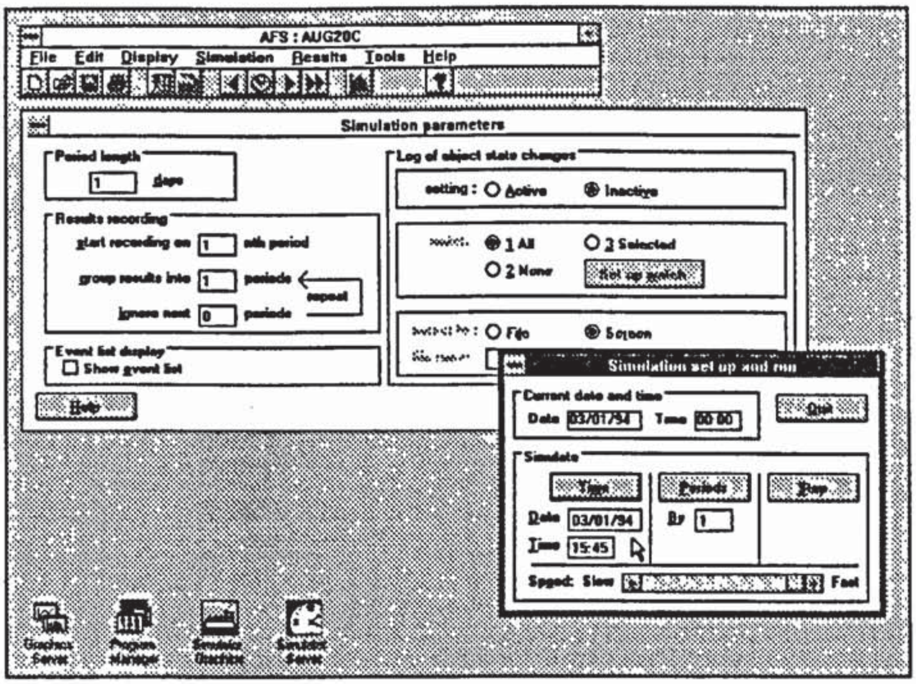


Figure 11.11: The class/object configuration dialog.

The above example illustrates the power of messaging. It is important to emphasise that the only formal link between classes (and thereby objects) is their 'send' and 'receive' messages; these links, however, must be configured if any communication is to take place. Prior to configuration, classes do not have any knowledge of one another; following configuration, class knowledge remains minimal and is based on respective messages. For example, the *Operator* class only knows that the *StockData* can receive the message but knows nothing of what the class does with it on receipt, i.e. the sender class does not know how the receiver class interprets and implements the message. In essence, messaging allows classes and objects to be decoupled and thereby independent of one another; this property is supported by the encapsulation of data and behaviour within the class. The independence and decoupling of classes supports the extension of class libraries, which will be discussed in section 11.3.

11.2.6 Setting the simulation parameters

The simulation parameters and results recording are set within their respective 'simulation' dialogs: these are illustrated in Figure 11.12.



Period length: 1 days

Results recording
start recording on 1 nth period
group results into 1 periods
ignore next 0 periods
repeat

Event list display
☐ Show event list

Log of object state changes
setting : ○ Active ● Inactive
objects: ● 1 All ○ 3 Selected
○ 2 None
output to : ○ File ● Screen

Current date and time
Date 03/01/94 Time 00:00

Simulate
Date 03/01/94 By 1
Time 15:45
Speed: Slow Fast

Figure 11.12: Setting the simulation and results parameters.

The user has control over the simulation 'run' length: whether to simulate for a set length of time, a period or a single 'event', as well as the starting date and time. The length of the period can be adjusted as well as how the results are to be recorded. There are several ways in which the simulation results can be viewed: for example by class for a single period or by object for all periods.

11.3 Adding new classes to WBS/Control

One of the requirements placed upon WBS/Control is that of the ability to support expansion: it must be possible to increase the functionality of the class library through the introduction of new or modified classes. In terms of the 'building block' analogy, it must be possible to add new building blocks with the minimum disruption to those which already exist. The ability to add new classes to a library has distinct advantages:

- the range of functionality is not static;
- the development of classes can be an on-going process;
- modified classes can be added to the library.

In the context of WBS/Control, the functionality so far has concentrated on information classes and low level classes associated more with MRP and ROP systems. The inclusion of different planning and control functionality is necessary if more types of systems are to be modelled: for example kanban, period batch control (PBC) or shop floor scheduling and dispatching policies.

The inclusion of new functionality includes new or modified dialogs as well as new or modified classes. There is an important consideration when developing extensible class libraries: the software architecture must be able to support the process. The AFS architecture has been developed to support the addition of new classes. This has been made possible because the architecture itself makes no assumptions about the functionality which it supports (Ball, 1994). Consequently, any functionality can be supported provided that it conforms to certain requirements. WBS/Control has exploited the extensibility property inherent within the AFS architecture.

11.3.1 The roles of user and developer

The addition of new functionality to a class library raises an interesting question: who is to provide the new or modified functionality?

This thesis has argued throughout that the user of WBS/Control should be able to construct models without recourse to programming. The development of WBS/Control has supported this approach: the user builds models through the user-interface. The only restriction placed on the user is the range of functionality available; it will be evident that the user can only construct models from the available classes. It is inevitable that during the life-time of a class library, a potential user will desire new or modified functionality. In fact the ability to add new or modified classes is likely to extend the life-time of a class library.

The limits of WBS/Control functionality rely on the range of classes that software developers are prepared to add. The addition of planning and control functionality, through WBS/Control, has itself demonstrated the extensibility of the AFS architecture. The role of the developer is, therefore, crucial if the functionality is to be extended. Essentially, a developer is someone with programming skills and some knowledge of the AFS architecture. Furthermore, the application of object-oriented principles supports the developer in creating and integrating new functionality.

11.3.2 Adding new functionality

The addition of new classes in WBS/Control will require the creation of the class itself, its class manager and, if necessary, the creation of an associated dialog. This discussion concentrates on the creation of the class and class manager.

The class manager is registered with the simulator through the *Managers* class. This 'registration' means that the architecture will recognise the class and allow the class manager to create objects of the class. The application of object-oriented principles will mean that the data and behaviour associated with the class will be encapsulated within it. Invariably, however, new classes will inherit the properties of an existing class. The object-oriented property of 'reuse' means that the new class will only need to support the differences between it and existing classes of a similar type. For example, a new machine class will need to provide the code which differentiates it from the existing machine classes: the common code will be reused through inheritance. Depending on the implementation, objects of the new class can use and be used by other classes without modification.

A new class will need to include appropriate send and receive messages if objects of the class are to be able to communicate with other existing objects. The message passing capability means that classes do not require any knowledge of other classes except the structure of the messages they support. The decoupling of classes in this way supports the addition of new classes. This is because the classes (and objects) are configured, via their messaging capabilities, at simulation 'run' time.

The functionality of WBS/Control has been extended through the introduction of new low level classes: *GenDemand* and *Std_Dmrp* are examples. These particular classes have been introduced to extend the modelling capability to include the Distributed Manufacturing Resources Planning (DMRP) control system (Love, 1992; Love & Barekat, 1988). DMRP is a development of the MRP system which relies on the special circumstances which arise in cellular manufacturing systems.

When considering the addition of new classes in order to extend the modelling capability of WBS/Control, the relationship of DMRP to MRP has provided an appropriate test problem. The macro approach to class definition (discussed in chapter 9) would address this problem by adding a single new, and comprehensive, DMRP class. The development of appropriate low level classes provides a more efficient mechanism. It is only necessary to determine the additional functionality which would combine with the existing low level classes providing a MRP-based approach to extend to a DMRP approach. The required functionality will then define the additional low level class(es) which need to be developed. The class *GenDemand* has since been developed for this purpose. Figure 11.13 illustrates two pc-routings: one modelling a MRP system and the second a DMRP system.

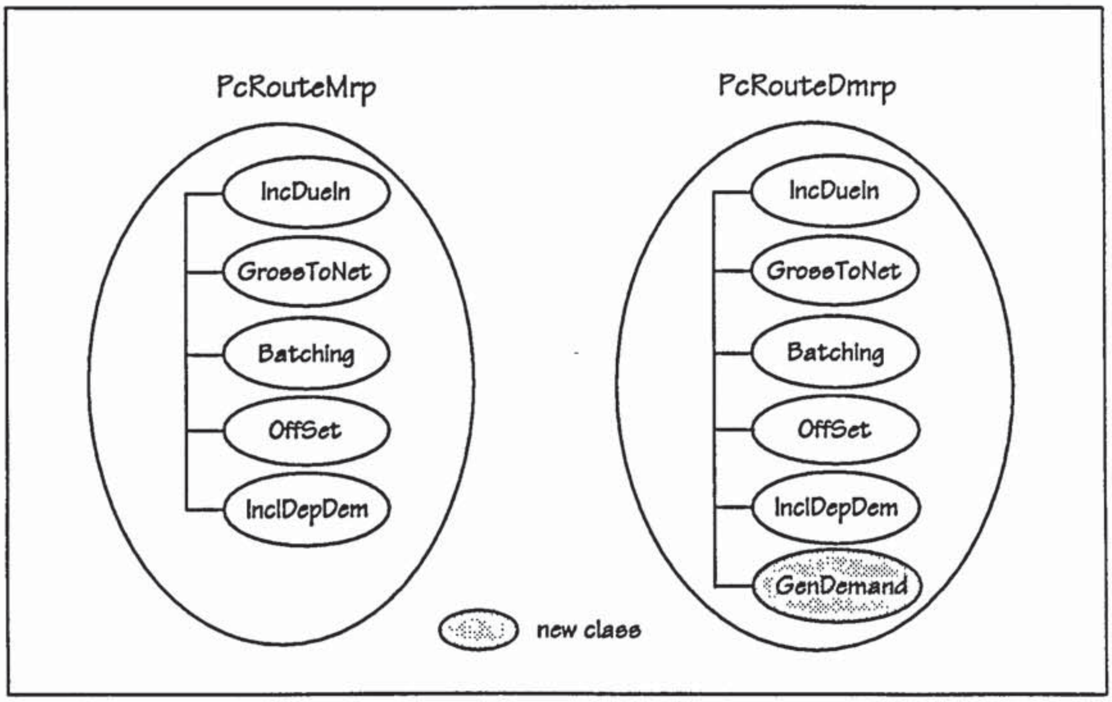


Figure 11.13: Extending WBS/Control through the introduction of a new class.

New functionality has been satisfactorily added to WBS/Control. The above discussion has been restricted to the addition of the class *GenDemand*. However, the library has been continually developed during this research project: as the new classes have been developed they have gradually extended the modelling functionality. It is intended to continue this process and thereby extend further the modelling capabilities of the class library. It is considered that WBS/Control is an extensible class library.

11.4 Summary

This chapter has discussed the application of WBS/Control. It is possible to construct simulation models of planning and control systems from the functionality offered by this class library; the developed models properly integrate with models of the shop floor facility. The user is able to build models without recourse to programming; the manufacturing simulator, AFS combined with WBS/Control, is completely data driven.

The functionality of WBS/Control can be extended through the addition of new or modified classes; there are potentially no restrictions to the functionality which can be added. It is concluded that WBS/Control is an extensible planning and control class library.

12. Conclusions and further work

This chapter will summarise the conclusions which have been reached from this research project. In the light of these conclusions, several opportunities for further work will be outlined.

12.1 WBS/Control - a planning and control class library

12.1.1 The problem summary

One of the core themes of this thesis has been the importance of the fit between a manufacturing planning and control system and the host organisation: in particular the relationship between the control system and the manufacturing system. It is apparent that design approaches do not properly evaluate, or assess, the impact of a design on the manufacturing facility. In other words, an understanding of how the new (or modified) planning and control system will perform in the context of the complete manufacturing system is unlikely to be gained until after the system has been implemented and is running. It is appropriate to reiterate the observation made by Rolstadås (1998a):

> *"Even if the* [production management] *system is fully defined through the user and system specifications, it is difficult for end-users to fully understand the properties of the system until after they have tried it in real-life. Experience shows that after implementation many user viewpoints are created, calling for a changed design."*

Rolstadås (1988a) indicates the role simulation can take in support of the design process. However, it is here that one of the problems can be traced: there is no simulation-based modelling tool which can properly support the process of designing planning and control systems. This thesis has reviewed the different applications of simulation to modelling manufacturing planning and control systems. Many of the simulation applications have addressed the relative performance of operational issues, such as lot sizing policies and scheduling rules, for specific problem situations. The more fundamental application in the design of planning and control systems is rarely considered.

It has been suggested that one of the reasons for the lack of attention to design issues may be the lack of an appropriate simulation-based modelling tool. The requirements for a simulation-based modelling tool capable of addressing both design and operational issues of manufacturing planning and control systems were determined. A summary of these requirements is as follows:

- to adopt a user-oriented perspective, or 'world view';
- to configure models without recourse to programming;
- to properly integrate with shop floor operations;
- to be able to model different planning and control systems;
- to be able to model hybrid control systems;
- to include elements of the planning and control infrastructure;
- to be able to extend the current functionality.

The following section will review how the current functionality offered by WBS/Control meets these extensive requirements.

12.1.2 Meeting the requirements

WBS/Control adopts a user-oriented perspective. The functionality currently offered by the range of classes which populate the class library represents that which would be found in the manufacturing planning and control domain. It is considered that the terminology which has been used and the implementation of the classes is consistent with a user-oriented perspective. A potential user will be able to map the 'real world' to that supported by the classes in the library. Wherever possible, consistency has been maintained between the real and modelled worlds.

To further support this approach, the user is able to configure models of different planning and control systems without any recourse to programming. This facility sets WBS/Control apart from other (simulator-based) approaches. Although the construction of the shop floor models may be possible without the user having to do any programming, the same can not be said of the inclusion of models of the control system.

Different planning and control systems can be modelled using WBS/Control. The development of low level classes represents a novel application of object-oriented principles to the problem of modelling planning and control systems. These low level classes form the 'building blocks' from which different control systems can be configured. Furthermore, in conjunction with the concept of a pc-routing, the low level classes provide the necessary flexibility to support the application of WBS/Control to design problems. As the number of low level classes increases then so does the range of different planning and control systems which can be modelled. The capability is such that standard systems, hybrid systems and 'new' designs can be represented using the low level classes and pc-routings.

The integration of WBS/Control into the architecture of the Advanced Factory Simulator (AFS) means that it is possible to build more comprehensive and realistic simulation models. The models of the planning and control system properly integrate with the shop floor operations. This is not the same as the interfacing of a shop floor model and a control system model. It is suggested that it is only through the proper integration of such models that the emergent properties of these systems can begin to be investigated.

The inclusion of elements of the planning and control infrastructure also supports integration of the type described above. The elements which are currently present, although not sophisticated, demonstrate that it is possible to include aspects of the 'policies, procedures and practices' which support the more computer-based elements of an organisation's planning and control system. The functionality of the class library supports the modelling of systems rather than their algorithms. Moreover, it will be possible to model dynamically the performance of policy review decisions.

WBS/Control is not restricted to the modelling of a single level of the planning and control hierarchy: it is possible to include the development of a master production schedule through to shop floor control within the same model. This means that it is possible to investigate the interfaces between the different levels of the planning hierarchy.

The current functionality offered by WBS/Control is not static: it is possible to add new or modified classes to the library. This means that it is possible to model a host of different planning and control systems, the only necessity being that the appropriate classes are provided by a software developer. This aspect of WBS/Control does, therefore, require someone with programming skills. The adoption of object-oriented methods means that it will be possible to add new classes with the minimum of disruption to existing classes in the library. It is concluded that WBS/Control is an extensible class library.

In summary, it is considered that the current development of WBS/Control makes a significant contribution to the problem of modelling manufacturing planning and control systems. Furthermore, the structure of the library supports further development.

12.2 The application of object-oriented methods

The development of WBS/Control has followed object-oriented principles: based on this experience, it is considered appropriate to include some comment on object-orientation. It is interesting to note that the experiences discussed here confirm those of other authors. This discussion will be relevant to developers and not users or modellers: these problems are, in effect, hidden from the end user.

The learning curve associated with any new technique should not be underestimated: object-orientation is no exception. Although viewing the problem domain in terms of 'objects' may be more 'natural', the development of object-oriented software requires both development and programming discipline: simply adopting object-oriented principles does not guarantee that the associated benefits will be achieved.

The use of class hierarchies and polymorphism can have drawbacks. From a point of view of debugging software, for example, it can prove difficult to trace an error when there may be several layers of inheritance.

The development of classes is not as straightforward as one might imagine. It is inevitable, therefore, to find that many classes moved through several iterations as the project developed: some classes became redundant as new classes were identified. The definition of classes has direct implication on the class hierarchies. Again, a number of iterations have taken place. The experience gained from the development has suggested that as more classes are developed, the class hierarchies themselves develop and tend to 'settle down'. The development of object libraries, as with most forms of software development, is a considerable task. Furthermore, it is not until a significant number of classes populate the library that the real benefits start to be gained.

The experiences described should not detract from the benefits which can be gained through the adoption of object-oriented methods which were discussed in chapter 8. From the point of view of WBS/control it is concluded that it would not have been possible to efficiently provide the functionality, based on the above requirements, using alternative techniques. It is important, however, for developers to be aware of some of the potential problems: object-orientation is not infallible.

12.3 Whole business simulation

The concept of whole business simulation (WBS) (Love et al., 1992) was introduced in chapter 1. WBS would be used to simulate the operations of the 'whole business' through the combination of real software systems and specialist simulation elements. A model of this scope, it is argued, will support the evaluation of different decisions in terms of the performance of the business. An overview of the scope of WBS is shown in Figure 12.1; the different core elements of the model are identified.

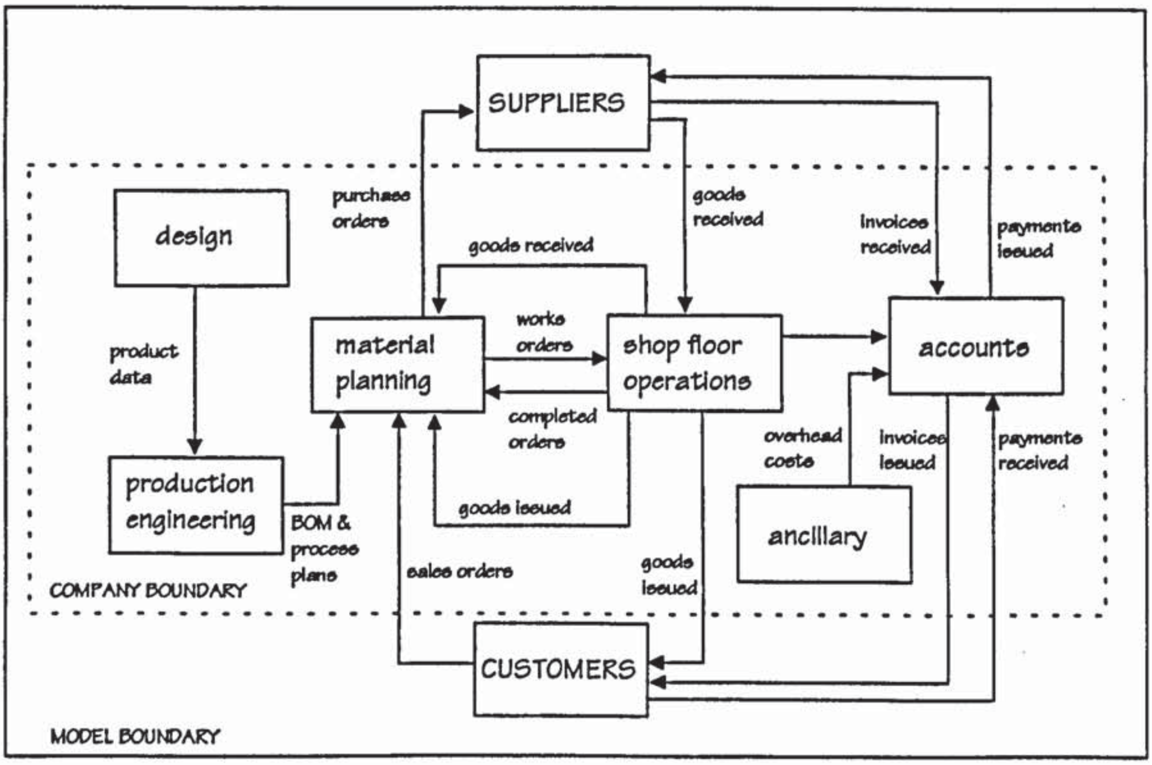


Figure 12.1: The core elements and key transactions of WBS (after Love et al., 1992).

It was previously mentioned that there are two ways in which the elements of a WBS model can be provided - either through the inclusion of real systems or through the provision of specialist simulation elements. For example, the 'accounts' element of Figure 12.1 is implemented in the 'WBS development system' (see Love & Barton, 1993) using dBFLEX: a fully functional accounting system. Similarly, the 'material planning' element is provided by UNIPLAN: a commercial MRP system. It is intended that different element representations should be interchangeable. For example, the 'material planning' element, represented by UNIPLAN could be replaced by a different representation: either another real system or a model of a material planning system.

One of the contributions of this work to the wider WBS research programme is the development of specialist simulation elements (i.e. classes) which will support the modelling of different planning and control systems. The flexibility of these elements extends the modelling capability of WBS: standard systems, hybrid systems and new designs can be modelled.

12.4 Further work

12.4.1 Additional functionality

The most immediate area for further work is to extend the modelling capability of WBS/Control through the addition of new classes.

The current functionality is not claimed to be representative of the planning and control domain. The initial intention was to demonstrate that it would be possible to derive classes which would model planning and control systems. The low level classes and the pc-routing concept provides sufficient flexibility such that different systems can be modelled: from the more 'standard' to potentially new designs. With this initial intention achieved, a greater coverage of planning and control systems is now required. This will mean the development of a range of classes.

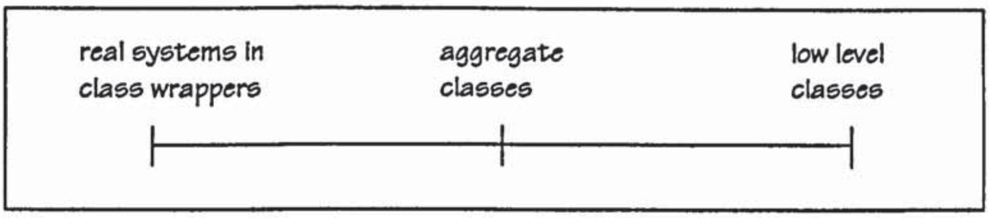


Figure 12.2: Different types of planning and control classes.

The range of different types of classes illustrated in Figure 12.2 was first introduced in chapter 9. This range will need to be supported by the architecture of WBS/Control. The role of class wrappers and aggregate classes require further discussion; the low level classes have been discussed in detail already.

Real systems (in class wrappers) have been used within the WBS demonstration system (see Love & Barton, 1993); the 'wrappers' provide the mechanism through which real (i.e. commercial systems) can be included in a simulation model. Essentially, 'wrappers' represent the interface between the real system and other elements in the model. The communication between systems and elements is supported in this way. A simple example is the interface to a real MRP system receiving the message to "do a MRP run": this would be interpreted by the wrapper and the appropriate action(s) taken.

If more real systems are to be included then appropriate wrappers will need to be developed: although wrappers are system specific there may be an opportunity to develop a 'wrapper' class hierarchy.

Aggregate classes will provide a less detailed representation of the respective planning and control functions. These have been described as high level classes in Boughton & Love (1994), in contrast to the detailed approach offered by low level classes. Aggregate classes would be useful in situations where a detailed model may not be required, alternatively they may replace detailed aspects of a simulation model in order to reduce simulation 'run time'.

12.4.2 Industrial applications

One interesting application of WBS/Control (and thereby AFS) would be in a real industrial context, i.e. to support an organisation in the design of a new or modified planning and control system. This, after all, is one of the envisaged applications of WBS/Control. An industrial application would provide a considerable test: it has been suggested that a tool of this kind would support the evaluation of a control system design and be able to assess its impact on the performance of the manufacturing facility. Performing such analysis may uncover potential problems within the design which can then be addressed before, rather than during or following, the system implementation. Furthermore, the flexibility of WBS/Control should support the organisation through the transition from the 'as-is' to the 'to-be' situations.

It is through such an application that the potential of WBS/Control can be thoroughly evaluated. In addition, an application of this kind would undoubtedly identify aspects of the class library which would need addressing: for example the addition of new (or modified) functionality or the modification of the user-interface.

12.4.3 The development of a design methodology

The application of WBS/Control to industrial problems itself raises the question of 'how' the modelling tool should be included in the planning and control system design process. The different approaches to the design of manufacturing control systems were discussed in chapter 4: it was concluded that a modelling tool, such as WBS/Control, was not used to assess the impact of a design on the manufacturing system or, for that matter, the business as a whole.

It would appear appropriate to investigate the development of a design methodology which would include some form of design evaluation of the scope advocated in this thesis. It is suggested that the development of a methodology along the lines of the Lucas manufacturing system design methodology (Lucas Engineering & Systems, 1991) would be an appropriate area of further work. This particular methodology develops a design from concept through to detailed design, including assessment of performance at the 'steady state' and 'dynamic design' stages.

A similar approach could be adopted to develop a planning and control system design: the design could be tested (in the context of the manufacturing system) at the various stages of development. The inclusion of WBS/Control as the means of modelling the designs would require a range of both aggregate and low level classes. The implicit flexibility is an essential attribute for use in such design situations.

12.4.4 The relationship between WBS/Control and real systems

One interesting corollary to the work described in this thesis is the relationship between the planning and control system model, as developed through WBS/Control, and the final production version implemented in the organisation. In essence, the planning and control system model would be a prototype of the intended system: the elements of the model would map to the required elements of the real system.

The link between the prototype and the real system warrants further investigation: it may be possible to support the development of the production system through the elements defined in the (prototype) model. The use of the classes in this way bears some similarity with an approach suggested by Rolstadås (1988b) and the software templates used by Muhlemann et al. (1995, 1990).

12.5 Closing remarks

This thesis has reported on the development of a class library called WBS/Control, which is based on object-oriented principles and discrete-event simulation. The functionality, both current and future, offered by WBS/Control means that different planning and control systems can be modelled: not only the more standard implementations of such systems but also hybrid systems. When investigating manufacturing planning and control systems, WBS/Control can be applied to both design and operational issues. WBS/Control integrates with an existing manufacturing simulator to provide a more complete modelling environment.

References

Adiga, S., 1993, "Object-oriented software systems: concepts", in *Object-oriented software for manufacturing systems*, editor: Adiga, S., Chapman & Hall, London., pp. 9-24.

Adiga, S. & Glassey, C.R., 1991, "Object-oriented simulation to support research in manufacturing systems", *International Journal of Production Research*, **29** (12), pp. 2529-2542.

Alasuvanto, J., Eloranta, E., Fuyuki, M., Kida, T. & Inoue, I., 1988, "Object oriented programming in production management - two pilot systems", *International Journal of Production Research*, **26** (5), pp. 765-776.

Askin, R.G. & Standridge, C.R., 1993, *Modeling and analysis of manufacturing systems*, John Wiley & Sons, New York.

Ball, P.D., 1994, *Design of an expandable manufacturing simulator through the application of object-oriented principles*, PhD Thesis, The University of Aston in Birmingham, UK.

Ball, P.D. & Love, D.M., 1994, "Expanding the capabilities of manufacturing simulators through application of object-oriented principles", *Journal of Manufacturing Systems*, **13** (6), pp. 412-423.

Ball, P.D., Boughton, N.J. & Love, D.M., 1994, "Extending the boundaries of manufacturing simulation", *International Journal of Manufacturing System Design*, **1** (2), pp. 99-109.

Banerjee, S.K., Kourouklis, A.P. & Penman, J., 1993, "Manufacturing planning and control decision model - a design methodology", *Proceedings of the International Conference on Managing Integrated Manufacturing: Organization, Strategy and Technology*, editors: Hassard, J.S., Forrester, P.L., Hawksley, C. & Tang, N.K.H., 22-24 September, Keele University, UK, pp. 659-669.

Banks, J., 1994, "Software for simulation", *Proceedings of the 1994 Winter Simulation Conference*, editors: Tew, J.D., Manivannan, S., Sadowski, D.A. & Seila, A.F., pp. 26-33.

Banks, J., Aviles, E., McLaughlin, J.R. & Yuan, R.C., 1991, "The simulator: new member of the simulation family", *Interfaces*, **21** (2), pp. 76-86.

Barber, K.D. & Hollier, R.H., 1986a, "The use of numerical analysis to classify companies according to production control complexity", *International Journal of Production Research*, **24** (1), pp. 203-222.

Barber, K.D. & Hollier, R.H., 1986b, "The effects of computer-aided production control systems on defined company types", *International Journal of Production Research*, **24** (2), pp. 311-327.

Basnet, C.B., Karacal, S.C. & Beaumariage, T.G., 1990, "Experiences in developing an object-oriented modeling environment for manufacturing systems", *Proceedings of the 1990 Winter Simulation Conference*, editors: Balci, O., Sadowski, R.P. & Nance, R.E., pp. 477-481.

Benchmark Research, 1994a, "Computers in manufacturing survey", in *Manufacturing Computer Solutions*, November, pp. 25-37.

Benchmark Research, 1994b, *The industrial IT report*, Benchmark Research Ltd., Swanley, Kent, UK.

Bennett, D., Lewis, C. & Oakley, M., 1988, *Operations management*, Philip Allan.

Bennett, D.J. & Forrester, P.L., 1993, *Market-focused production systems: design and implementation*, Prentice Hall, UK.

Berry, W.L. & Hill, T., 1992, "Linking systems to strategy", *International Journal of Operations and Production Management*, **12** (10), pp. 3-15.

Bertrand, J.W.M. & Wijngaard, J., 1986, "The structuring of production control systems", *International Journal of Operations and Production Management*, **6** (2), pp. 5-20.

Bertrand, J.W.M., Wortmann, J.C. & Wijngaard, J., 1990, *Production control: a structural and design oriented approach*, Manufacturing Research and Technology, **11**, Elsvier, Amsterdam, The Netherlands.

Bhuskute, H.C., Duse, M.N., Gharpure, J.T., Pratt, D.B., Kamath, M. & Mize, J.H., 1992, "Design and implementation of a highly reusable modeling and simulation framework for discrete part manufacturing systems", *Proceedings of the 1992 Winter Simulation Conference*, editors: Swain, J.J., Goldsman, D., Crain, R.C. & Wilson, J.R., pp. 680-688.

Bikos, A., Williams, G.B. & O'Connor, R.F., 1994, "Assembly systems sensitivity to component delivery: a logistics study using simulation", *Proceedings of the 4th International Conference on Factory 2000 - Advanced Factory Automation*, 3-5 October 1994, University of York, UK, IEE publication no. 398, pp. 638-644.

Bilberg, A. & Alting, L., 1991, "When simulation takes control", *Journal of Manufacturing Systems*, **10** (3), pp. 179-193.

Bischak, D.P. & Roberts, S.D., 1991, "Object-oriented simulation", *Proceedings of the 1991 Winter Simulation Conference*, editors: Nelson, B.L., Kelton, W.D. & Clark, G.M., pp. 194-203.

Booch, G., 1994, *Object-oriented analysis and design with applications* (2nd edition), The Benjamin Cummings Publishing Company, California, US.

Boughton, N.J. & Love, D.M., 1994, "An object-oriented approach to modelling manufacturing planning and control systems", *Proceedings of the 10th National Conference on Manufacturing Research*, editors: Case, K. & Newman, S.T., 13-15 September, Loughborough University of Technology, UK, pp. 426-430.

Boyle, A. & Kaldos, A., 1994, "Performance evaluation of flexible manufacturing systems", *Proceedings of the 4th International Conference on Factory 2000 - Advanced Factory Automation*, 3-5 October 1994, University of York, UK, IEE publication no. 398, pp. 434-441.

Bravoco, R.R. & Yadav, S.B., 1985, "A methodology to model the dynamic structure of an organisation", *Information Systems*, **10** (3), pp. 299-317.

Brennan, L. & Gupta, S.M., 1993, "A structured analysis of material requirements planning systems under combined demand and supply uncertainty", *International Journal of Production Research*, **31** (7), pp. 1689-1707.

Bridge, K., 1990, *The application of computerised modelling techniques in manufacturing system design*, PhD Thesis, The University of Aston in Birmingham, UK.

Brooks, F.P., 1979, *The mythical man-month: essays on software engineering*, Addison-Wesley Publishing Company, Reading, Massachusetts, US.

Browne, J., 1988, "Simulation and simulation models", in *Computer-Aided Production Management*, editor: Rolstadås, A., Springer-Verlag, pp. 124-134.

Browne, J. & Davies, B.J., 1984, "The phased release of batches into a job shop", *International Journal of Operations and Production Management*, **4** (4), pp. 16-27.

Browne, J., Haren, J. & Shivnan, J., 1988, *Production management systems: a CIM perspective*, Addison-Wesley Publishing Company, Wokingham, England.

Burbidge, J.L., 1971, *The principles of production control* (3rd edition), MacDonald & Evans, London.

Burbidge, J.L., 1975, *The introduction of group technology*, Heinemann, London.

Burbidge, J.L., 1985, "Production planning and control - a personal philosophy", *Computers in Industry*, **6** (1985), pp. 477-487.

Burbidge, J.L., 1988, "Period Batch Control", in *Computer-Aided Production Management*, editor: Rolstadås, A., Springer-Verlag, pp. 71-75.

Burbidge, J.L., 1992, "Change to group technology - process organisation is obsolete", *International Journal of Production Research*, **30** (5), pp. 1202-1219.

Busby, J.S. & Williams, G.M., 1993, "The value and limitations of using process models to describe the manufacturing organisation", *International Journal of Production Research*, **31** (9), pp. 2179-2194.

CACI, 1993, *Reference manual for SIMFACTORY II.5 and SIMPROCESS* (release 6.0), CACI Products Company, Surrey, UK.

Carrie, A., 1988, *Simulation of manufacturing systems*, John Wiley & Sons, UK.

Chaharbaghi, K., 1990, "Using simulation to solve design and operational problems", *International Journal of Operations and Production Management*, **10** (9), pp. 89-105.

Chang, A-M., Kannan, P.K. & Wong, B.O., 1991, "Design of an object-oriented system for manufacturing planning and control", *Proceedings of the 2nd Rensselaers International Conference on Computer Integrated Manufacture*, 21-23 May, pp. 2-8.

Christy, D.P. & Watson, H.J., 1983, "The application of simulation: a survey of industrial practice", *Interfaces*, **13** (5), pp. 47-52.

Chu, C-H. & Shih, W-L., 1992, "Simulation studies in JIT production", *International Journal of Production Research*, **30** (11), pp. 2573-2586.

Clark, P., Newell, S., Burcher, P., Bennett, D., Swan, J. & Sharifi, S., 1991, "Computer aided production management: an example of organisation-and-technology systems", *Proceedings of the OMA-UK 6th International Conference: Achieving Competitive Edge*, editors: Bennett, D. & Lewis, C., 25-26 June, Aston University, pp. 103-108.

Clarke, S.R., 1988, *Aspects of the design and control of manufacturing systems subject to demand uncertainty*, PhD Thesis, The University of Aston in Birmingham, UK.

Collins, N. & Watson, C.M., 1993, "Introduction to ARENA", *Proceedings of the 1993 Winter Simulation Conference*, editors: Evans, G.W., Mollaghasemi, M., Russell, E.C. & Biles, W.E., pp. 205-212.

Colquhoun, G.J. & Baines, R.W., 1991, "A generic IDEF0 model of process planning", *International Journal of Production Research*, **29** (11), pp. 2239-2257.

Colquhoun, G.J., Baines, R.W. & Crossley, R., 1993, "A state of the art review of IDEF0", *International Journal of Computer Integrated Manufacturing*, **6** (4), pp. 252-264.

Consultants' Conspectus, 1995, *Object technology*, March, Prime Marketing Publications Ltd., Herts., UK.

Conway, R. & Maxwell, W.L., 1986, "XCELL: a cellular, graphical factory modelling system", *Proceedings of the 1986 Winter Simulation Conference*, editors: Wilson, J., Henriksen, J. & Roberts, S., pp. 160-163.

Cook, S. & Daniels, J., 1993, "Essential techniques for object-oriented design", *Proceedings of the Seminar Series on new Directions in Software Development: Object-Oriented Techniques*, editor: Jackson, M., University of Wolverhampton, UK, pp. 20-36.

Cook, D.P., 1994, "A simulation comparison of traditional, JIT, and TOC manufacturing systems in a flow shop with bottlenecks", *Production and Inventory Management Journal*, 1st quarter, **35** (1), pp. 73-78.

Corke, D.K., 1977, *Production control in engineering*, Edward Arnold (Publishers) Ltd., London.

Cox, B.J. & Novobilski, A.J., 1991, *Object-oriented programming: an evolutionary approach*, Addison-Wesley Publishing Company, Reading, Massachusetts, US.

Crookall, J.R., 1985, "Simulation, emulation and computer integrated manufacture", *Proceedings of the 1st International Conference on Simulation in Manufacturing*, 5-7 March, Stratford-upon-Avon, UK, pp. 339-347.

Dahl, O.J. & Nygaard, K., 1966, "SIMULA - an ALGOL-based simulation language", *Communications of the ACM*, **9** (9), pp. 671-678.

Dale, B.G. & Russell, D., 1983, "Production control systems for small group production", *OMEGA, The International Journal of Management Science*, **11** (2), pp. 175-185.

Devereux, S., Smith, P. & Wood, D., 1994, "A survey of the use of design methodologies for implementing change in manufacturing companies in the United Kingdom", *International Journal of Manufacturing System Design*, **1** (1), pp. 51-58.

Doumeingts, G., 1985, "How to decentralize decisions through the GRAI model in production management", *Computers in Industry*, **6** (1985), pp. 501-514.

Doumeingts, G., 1988, "Systems analysis techniques", in *Computer-Aided Production Management*, editor: Rolstadås, A., Springer-Verlag, pp. 201-224.

Duchessi, P., Schaninger, C.M. & Hobbs, D.R., 1989, "Implementing a manufacturing planning and control information system", *California Management Review*, Spring, **31** (3), pp. 75-90.

Flapper, S.D.P., Miltenburg, G.J. & Wijngaard, J., 1991, "Embedding JIT into MRP", *International Journal of Production Research*, **29** (2), pp. 329-341.

Fox, R.E., 1982, "OPT: an answer for America, part 2", *Inventories and Production Magazine*, **2** (6).

Fritz, S., Schmid, F. & Wu, B., 1993, "A survey of the current practice and development in computer aided manufacturing systems design", *Proceedings of the International Conference on Managing Integrated Manufacturing: Organization, Strategy and Technology*, editors: Hassard, J.S., Forrester, P.L., Hawksley, C. & Tang, N.K.H., 22-24 September, Keele University, UK, pp. 673-689.

Gardiner, S.C. & Blackstone, J.H., 1993, "Impact of lot-sizing on the financial performance of dispatching techniques in an MRP-planned fabrication and assembly environment", *International Journal of Production Research*, **31** (7), pp. 1595-1610.

Giauque, W.C. & Sawaya, W.J., 1992, "Strategies for production control", *Production and Inventory Management Journal*, 3rd quarter, **33** (3), pp. 36-39.

Ghobadian, A., 1992, "MRPII - the key to successful implementation", *Proceedings of the 8th International Conference on Computer-Aided Production Engineering*, editor: McGeough, J.A., Edinburgh, UK, pp. 183-190.

Glassey, C.R. & Adiga, S., 1989, "Conceptual design of a software object library for simulation of semiconductor manufacturing systems", *Journal of Object-Oriented Programming*, Nov/Dec, **2** (4), pp. 39-43.

Gooden, D.I., 1988, *The development of a total system simulation modelling approach for management decision support*, MPhil Thesis, The University of Aston in Birmingham, UK.

Graham, I., 1991, *Object oriented methods*, Addison-Wesley Publishing Company, Wokingham, England.

Grünwald, H., Striekwold, P.E.T. & Weeda, P.J., 1989, "A framework for quantitative comparison of production control concepts", *International Journal of Production Research*, **27** (2), pp. 281-292.

Harrell, C.R. & Leavy, J.J., 1993, "ProModel Tutorial", *Proceedings of the 1993 Winter Simulation Conference*, editors: Evans, G.W., Mollaghasemi, M., Russell, E.C. & Biles, W.E., pp. 184-189.

Hayward, C., 1992, "Keeping software under control", *Engineering*, September, **232** (8), pp. 40.

Hendry, L.C. & Kingsman, B.G., 1989, "Production planning systems and their applicability to make-to-order companies", *European Journal of Operational Research*, **40** (1989), pp. 1-15.

Hendry, L.C., Wilson, E., Kingsman, B.G., Worden, L. & Mercer, A., 1993, "Integrating production planning in make-to-order companies", *Proceedings of the International Conference on Managing Integrated Manufacturing: Organization, Strategy and Technology*, editors: Hassard, J.S., Forrester, P.L., Hawksley, C. & Tang, N.K.H., 22-24 September, Keele University, UK, pp. 427-435.

Herring, C., 1990, "ModSim: a new object-oriented simulation language", *Proceedings of the SCS Multi-Conference on Object-Oriented Simulation*, editor: Guasch, A., 17-19 January, pp. 55-60.

Hicks, C. & Braiden, P.M., 1994, "Competitive issues in make to order manufacturing", *INRIA - European workshop on integrated manufacturing systems engineering*, 12-14 December, Grenoble, France, pp. 395-407.

Hicks, C. & Braiden, C.M., 1995, "Company classification and manufacturing control and resource issues in make to order manufacturing", *Proceedings of the European Conference on Management of Technology: Technological Innovation and Global Challenges*, editors: Bennett, D & Steward, F., 5-7 July, Aston University, UK, pp. 485-492.

Hill, J.F. & Brook, I.C., 1994, "The selection and design of applicable PPC systems for batch manufacturing environments", *Proceedings of the 11th Conference of the Irish Manufacturing Committee (IMC-11): Lean Production - From Concept to Product*, editor: Storrar, A.M., 31 Aug - 02 Sept, The Queens University of Belfast, UK, pp. 937-946.

Hill, T., 1991, *Production / operations management: text and cases* (2nd edition), Prentice Hall, London.

Hill, T., 1993, *Manufacturing strategy* (2nd edition), Macmillan, London.

Hlupic, V. & Paul, R.J., 1993, "Simulation software in manufacturing environments: a users' survey", *Journal of Computing and Information Technology*, **1** (3), pp. 205-212.

Ho, K-F. & Ridgway, K., 1994, "Design of cellular manufacturing systems using GRAI methodology", *Proceedings of the 10th National Conference on Manufacturing Research*, editors: Case, K. & Newman, S.T., 13-15 September, Loughborough University of Technology, UK, pp. 169-173.

Huettner, C.M. & Steudel, H.J., 1992, "Analysis of a manufacturing system via spreadsheet analysis, rapid modelling, and manufacturing simulation", *International Journal of Production Research*, **30** (7), pp. 1699-1714.

Ince, D., 1993, "An introduction to object-oriented technology", *Proceedings of the Seminar Series on new Directions in Software Development: Object-Oriented Techniques*, editor: Jackson, M., University of Wolverhampton, UK, pp. 1-19.

Ingersoll Engineers, 1985, *Integrated manufacture*, IFS (Publications) Ltd., UK.

Ingersoll Engineers, 1990, *Competitive manufacturing: the quiet revolution*, December, Ingersoll Engineers Ltd., Rugby, UK.

Jackman, J. & Johnson, E., 1993, "The role of queueing network models in performance evaluation of manufacturing systems", *Journal of the Operational Research Society*, **44** (8), pp. 797-807.

Jackson, A.T. & Love, D.M., 1994, "Evaluating the impact of engineering decisions on the support departments of a manufacturing business", *Proceedings of the 11th Conference of the Irish Manufacturing Committee (IMC-11): Lean Production - From Concept to Product*, editor: Storrar, A.M., 31 Aug - 02 Sept, The Queens University of Belfast, UK, pp. 729-736.

Joannis, R. & Krieger, M., 1992, "Object-oriented approach to the specification of manufacturing systems", *Computer-Integrated Manufacturing Systems*, **5** (2), pp. 133-145.

Kasales, C.J. & Sturrock, D.T., 1991, "Introduction to SIMAN IV", *Proceedings of the 1991 Winter Simulation Conference*, editors: Nelson, B.L., Kelton, W.D. & Clark, G.M., pp. 106-111.

Kellock, B., 1992, "Unlocking the rewards of cellular manufacture", *Machinery and Production Engineering*, 21 February, **150** (3821), pp. 32-40.

King, C.U. & Fisher, E.L., 1986, "Object-oriented shop-floor design, simulation, and evaluation", *Proceedings of the 1986 Fall Industrial Engineering Conference*, Institute of Industrial Engineers, December, Boston Spa, Massachusetts, US, pp. 131-137.

King, J.R., 1972, "Production planning and control (PPC) by computer - a survey", *The Production Engineer*, October, pp. 333-336.

Kirkwood, R, Smith, S. & Tranfield, D., 1989, "The implementation cube for advanced manufacturing systems", *International Journal of Operations and Production Management*, **9** (8), pp. 5-19.

Kochhar, A.K., 1984, "Implementation of computer aided production control systems - the gap between success and failure", *BPICS Control*, **10** (6), pp. 15-19.

Kochhar, A.K., 1987, "A company snapshot", Appendix B in "A study of computer aided production management in UK batch manufacturing", Monniot, J.P., Rhodes, D.J., Towill, D.R. & Waterlow, J.G., *International Journal of Operations and Production Management*, **7** (), pp. 41-57.

Kochhar, A.K. & McGarrie, B., 1991, "Selection and effective implementation of manufacturing control systems - the need for a structured approach", *Proceedings of the 26th Annual European Conference of BPICS*, pp. 241-256.

Kochhar, A.K., Suri, A.K., Oldham, K., Thacker, S.M. & McGarrie, B., 1992, "A structured methodology for the effective implementation of manufacturing control systems", *Proceedings of the ACME Conference*, pp. 109-114.

Kochhar, A.K., Kennerley, M. & Davies, A., 1994, "A framework for performance indicators in manufacturing control systems", *Proceedings of the 1st International Conference of the European Operations Management Association: Operations Strategy and Performance*, editors: Platts, K.W., Gregory, M.J. & Neely, A.D., 27-29 June, University of Cambridge, UK, pp. 85-92.

KPMG Management Consultants, 1994, *Object technology survey results*, KPMG Management Consulting, London.

Krajewski, L., King, B.E., Rizman, L.P. & Wong, D.S., 1987, "Kanban, MRP and shaping the manufacturing environment", *Management Science*, **33** (1), pp. 39-57.

Larsen, N.E. & Alting, L., 1993, "Criteria for selecting a production control philosophy", *Production Planning and Control*, **4** (1), pp. 54-68.

Law, A.M. & Kelton, W.D., 1991, *Simulation modeling and analysis*, McGraw Hill, New York.

Lenz, J.E., 1985, "MAST: a simulation as advanced as the FMS it studies", *Proceedings of the 1st International Conference on Simulation in Manufacturing*, 5-7 March, Stratford-upon-Avon, UK, pp. 313-324.

Lewis, C.D., 1981, *Scientific inventory control* (2nd edition), Butterworths, Kent, UK.

Lewis, P.A. & Love, D.M., 1993, "The use of generic elements for the design of kanban systems", *Proceedings of the 9th National Conference on Manufacturing Research*, editors: Bramley, A. & Mileham, T., 7-9 September, University of Bath, UK, pp. 67-71.

Lewis, P.A., 1994, *A systemic approach to the design of cellular manufacturing systems*, PhD Thesis, The University of Aston in Birmingham, UK.

Lilegdon, W.R., 1992, "Manufacturing decision making with FACTOR", *Proceedings of the 1992 Winter Simulation Conference*, editors: Swain, J.J., Goldsman, D., Crain, R.C. & Wilson, J.R., pp. 361-365.

Little, D. & Johnson, S.B., 1990, "Survey of UK manufacturing control practice - a brief report", *BPICS Control*, **16** (3), pp. 31-33.

Lockett, A.G., Barrar, P.R.N. & Polding, M.E., 1991, "MRPII systems: success factors in the process of implementation", in *Production Research: Approaching the 21st Century*, editors: Pridham, M, & O'Brien, C., Taylor & Francis, London, pp. 263-269.

Lopez, D.A. & Haughton, D.A., 1993, "World-class manufacturing in Northwest electronics companies: distressing results for technology and mpc applications", *Production and Inventory Management Journal*, 4th quarter, **34** (4), pp. 56-60.

Love, D.M., 1992, "Concepts, implementation and potential of DMRP", *IEE Seminar on "DMRP and Cellular Manufacture"*, 4 June, Savoy Place, London, UK.

Love, D.M. & Barekat, M., 1988, "An introduction to the decentralised, distributive MRP philosophy: a solution to control problems in cellular manufacturing systems", *Proceedings of the 3rd International Conference on Computer-Aided Production Engineering*, 01-03 June, University of Michigan, US, pp. 317-323.

Love, D.M., Barton, J.A. & Cope, N., 1992, "Whole business simulation and engineering applications", *Proceedings of the 8th International Conference on Computer-Aided Production Engineering*, editor: McGeough, J.A., Edinburgh, UK, pp. 166-173.

Love, D.M. & Barton, J.A., 1993, "Using whole business simulation to set operational policies in an integrated manufacturing system", *Proceedings of the International Conference on Managing Integrated Manufacturing: Organization, Strategy and Technology*, editors: Hassard, J.S., Forrester, P.L., Hawksley, C. & Tang, N.K.H., 22-24 September, Keele University, UK, pp. 145-157.

Lucas Engineering & Systems, 1991, *Mini guides: the Lucas manufacturing systems engineering handbook*, Lucas Engineering & Systems Ltd., UK.

Luscombe, M., 1993, *MRPII: integrating the business*, Butterworth Heinmann, Oxford.

MacCarthy, B.L. & Liu, J., 1993, "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling", *International Journal of Production Research*, **31** (1), pp. 59-79.

MacIntosh, R., 1992, "Modelling techniques for manufacturing systems integration", *Proceedings of the 8th National Conference on Manufacturing Research*, 8-10 September, University of Central England in Birmingham UK, pp. 229-233.

Maji, R.K., 1988, "Tools for development of information systems in CIM", *Advanced Manufacturing Engineering*, **1** (October), pp. 26-34.

Maull, R., Hughes, D., Childe, S., Weston, N., Tranfield, D & Smith, S., 1990, "A methodology for the design and implementation of resilient CAPM systems", *International Journal of Operations and Production Management*, **10** (9), pp. 27-36.

Maull, R.S., Childe, S.J. & Hughes, D.R., 1991, "A user led methodology for the design and implementation of integrated production management systems", in *Computer Applications in Production and Engineering: Integration Aspects*, editors: Doumeingts, G., Browne, J. & Tomljanovich, M., Elsvier Science Publishers B.V. (North Holland).

Maull, R.S. & Childe, S.J., 1993, "A step-by-step guide to the identification of an appropriate computer-aided production management system", *Production Planning and Control*, **4** (1), pp. 69-76.

Maull, R., Childe, S., Weaver, A., Bennett, J., Brown, P., O'Brien, J. & High, S., 1994, "The role of IDEF0 in process re-engineering", *Proceedings of the 10th National Conference on Manufacturing Research*, editors: Case, K. & Newman, S.T., 13-15 September, Loughborough University of Technology, UK, pp. 90-94.

McCarthy, I., Chodari, A. & Ridgway, K., 1994, "A model of the production management systems in a make to stock company", *Proceedings of the 10th National Conference on Manufacturing Research*, editors: Case, K. & Newman, S.T., 13-15 September, Loughborough University of Technology, UK, pp. 208-212.

McFadden, F.R., 1989, "Object-oriented techniques in computer integrated manufacturing", *Proceedings of the 22nd Annual HICSS (1989 IEEE)*, **3**, pp. 64-69.

McKay, K.N., Safayeni, F.R. & Buzacott, J.A., 1988, "Job-shop scheduling theory: what is relevant?", *Interfaces*, **18** (4), pp. 84-90.

Miller, J.G. & Vollmann, T.E., 1985, "The hidden factory", *Harvard Business Review*, **63** (5), pp. 142-150.

Minifie, J.R. & Heard, E., 1985, "On the generalizability of MRP simulation results", *Engineering Costs and Production Economics*, **9** (1985), pp. 211-217.

Mittal, S. & Wang, H.P., 1992, "Simulation of JIT production to determine number of kanbans", *International Journal of Advanced Manufacturing Technology*, **7** (5), pp. 292-305.

Mize, J.H., Bhuskute, H.C., Pratt, D.B. & Kamath, M., 1992, "Modelling of integrated manufacturing systems using an object-oriented approach", *IIE Transactions*, July, **24** (3), pp. 14-26.

Monden, Y., 1994, *Toyota production system: an integrated approach to just-in-time* (2nd edition), Chapman & Hall, London.

Muhlemann, A.P., Price, D.H.R., Sharp, J.A., Afferson, M. & Andrews, J.K., 1990, "Information systems for use by production management in smaller manufacturing enterprises", *Proceedings of the IMechE, part B: Journal of Engineering Manufacture*, **204**, pp. 191-196.

Muhlemann, A.P., Price, D.H.R., Afferson, M. & Sharp, J.A., 1995, "A computer based approach for enhancing manufacturing decision making in smaller manufacturing enterprises: a longitudinal case study", *OMEGA, The International Journal of Management Science*, **23** (1), pp. 97-107.

Murphy, J.C. & Braund, S.L., 1990, "Management accounting and new manufacturing technology", *Management Accounting*, **68** (2), pp. 38-39.

Najmi, A. & Stein, S.J., 1989, "Comparison of conventional and object-oriented approaches for simulation of manufacturing systems", *Proceedings of the 1989 IIE Integrated Systems Conference*, Atlanta, US, pp. 471-476.

Newman, W.R. & Sridharan, V., 1995, "Linking manufacturing planning and control to the manufacturing environment", *Integrated Manufacturing Systems*, **6** (4), pp. 36-42.

Nof, S.Y., 1994, "Critiquing the potential of object orientation in manufacturing", *International Journal of Computer Integrated Manufacturing*, **7** (1), pp. 3-16.

Norrie, D.H. (editor), 1992, *Proceedings of the International Conference on Object-Oriented Manufacturing Systems*, 4-6 May, The University of Calgary, Alberta, Canada.

O'Keefe, R.M. & Haddock, J., 1991, "Data-driven generic simulators for flexible manufacturing systems", *International Journal of Production Research*, **29** (9), pp. 1795-1810.

O'Reilly, J.J., 1993, "Introduction to SLAM II and SLAMSYSTEM", *Proceedings of the 1993 Winter Simulation Conference*, editors: Evans, G.W., Mollaghasemi, M., Russell, E.C. & Biles, W.E., pp. 179-183.

Orlicky, J., 1975, *Material requirements planning*, McGraw Hill Book Company, New York, US.

Ovrin, P., 1990, "The performance of manufacturing planning and control methods in different production facilities: a literature survey", paper submitted to *The 6th International Working Seminar on Production Economics*, 19-23 February, Igls, Innsbruck.

Papadopoulos, H.T., Heavey, C. & Browne, J., 1993, *Queueing theory in manufacturing systems analysis and design*, Chapman & Hall, London.

Parnaby, J., 1986, "The design of competitive manufacturing systems", *International Journal of Technology Management*, **1** (3,4), pp. 385-396.

Parnaby, J., Bhattacharyya, S.K., Burns, N.D., Hessey, M.F. & Larner, A., 1986, *Minicomputers and microcomputers in engineering and manufacture*, Collins, London.

Parnaby, J., Johnson, P. & Herbison, B, 1987, "Development of the JIT-MRP factory control system", *Proceedings of the 2nd International Conference on Computer-Aided Production Engineering*, editor: McGeough, J.A., Edinburgh, UK, pp. 17-22.

Passler, E., Hutchinson, G.K., Rudolph, K. & Stanek, W., 1983, "Production system design: a directed graph approach", *International Journal of Manufacturing Systems*, **2** (2), pp. 107-116.

Paul, R.J., 1991, "Recent developments in simulation modelling", *Journal of the Operational Research Society*, **42** (3), pp. 217-226.

Pels, H.J. & Wortmann, J.C., 1990, "Modular design of integrated databases in production management systems", *Production Planning and Control*, **1** (3), pp. 132-146.

Persentili, E. & Alptekin, S., 1993, "Integration of simulation modeling and computer aided production management in computer integrated enterprise", *Computers and Industrial Engineering*, **25** (1-4), pp. 191-194.

Pidd, M., 1992a, *Computer simulation in management science* (3rd edition), John Wiley & Sons, UK.

Pidd, M., 1992b, "Guidelines for the design of data driven generic simulators for specific domains", *SIMULATION*, **59** (4), pp. 237-243.

Pidd, M., 1992c, "SKIM: a simulation sketchpad for plant design", *Journal of the Operational Research Society*, **43** (12), pp. 1121-1133.

Pidd, M., 1995, "Object-orientation, discrete simulation and the three-phase approach", *Journal of the Operational Research Society*, **46** (3), pp. 362-374.

Plaia, A. & Carrie, A., 1995, "Application and assessment of IDEF3 - process flow description capture method", *International Journal of Operations and Production Management*, **15** (1), pp. 63-73.

Prabhu, P., Sharit, J. & Drury, C., 1992, "Classification of temporal errors in CIM systems: development of a framework for deriving human-centred information requirements", *International Journal of Computer Integrated Manufacturing*, **5** (2), pp. 68-80.

Ptak, C.A., 1991, "MRP, MRPII, OPT, JIT and CIM - succession, evolution or necessary combination", *Production and Inventory Management Journal*, 2nd quarter, **32** (2), pp. 7-11.

Rees, L.P., Huang, P.Y., Taylor, B.W., 1989, "A comparative analysis of an MRP lot-for-lot system and a kanban system for a multistage production operation", *International Journal of Production Research*, **27** (8), pp. 1427-1443.

Rembold, B. & Tanchoco, J.M.A., 1994, "A modular framework for the design of material flow systems", *International Journal of Production Research*, **32** (1), pp. 1-21.

Riis, J.O., "Situational production management: a practical theory for the development and application of production management", *Production Planning and Control*, **5** (3), pp. 240-252.

Rogers, P. (editor), 1994, "Special issue: object-oriented manufacturing systems", *International Journal of Computer Integrated Manufacturing*, **7** (1).

Rolstadås, A., 1988a, "Production management systems", in *Computer-Aided Production Management*, editor: Rolstadås, A., Springer-Verlag, pp. 3-19.

Rolstadås, A., 1988b, "Flexible design of production planning systems", *International Journal of Production Research*, **26** (3), pp. 507-520.

Rose, M., 1991, "Production planning at Texas Instruments improves service and reduces costs", *Industrial Engineering (IE)*, January, **23** (1), pp. 33-36.

Roth, A.V. & Miller, J.G., 1992, "Success factors in manufacturing", *Business Horizons*, Jul-Aug, **35** (4), pp. 73-81.

Rothenberg, J., 1986, "Object-oriented simulation: where do we go from here?", *Proceedings of the 1986 Winter Simulation Conference*, editors: Wilson, J., Henriksen, J. & Roberts, S., pp. 302-311.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. & Lorensen, W., 1991, *Object-oriented modelling and design*, Prentice Hall, London, UK.

Sarker, B.R. & Fitzsimmons, J.A., 1989, "The performance of push and pull systems: a simulation and comparative study", *International Journal of Production Research*, **27** (10), pp. 1715-1731.

Schlussel, B.M., 1990, "Twelve reasons MRPII can fail your company and what you can do about it", *Proceedings of the 33rd International Conference of the American Production and Inventory Control Society*, 8-12 October, New Orleans, US, pp. 436-437.

Schofield, N.A. & Bonney, M.C., 1981, "The use of manufacturing profiles to predict the applicability of computer systems", *International Journal of Operations and Production Management*, **1** (2), pp. 108-128.

Schonberger, R.J., 1982, *Japanese manufacturing techniques: nine hidden lessons in simplicity*, The Free Press, New York, US.

Schonberger, R.J., 1986, *World class manufacturing*, The Free Press, New York, US.

Scott, B., 1994, *Manufacturing planning systems*, McGraw Hill, London.

Shannon, R.E., 1992, "Introduction to simulation", *Proceedings of the 1992 Winter Simulation Conference*, editors: Swain, J.J., Goldsman, D., Crain, R.C. & Wilson, J.R., pp. 65-73.

Shewchuk, J.P. & Chang, T-C., 1991, "An approach to object-oriented discrete-event simulation of manufacturing systems", *Proceedings of the 1991 Winter Simulation Conference*, editors: Nelson, B.L., Kelton, W.D. & Clark, G.M., pp. 302-311.

Shimizu, M., 1991, "Application of an integrated modelling approach to the design and analysis of manufacturing systems", *Advanced Manufacturing Engineering*, **3** (January), pp. 3-17.

Sillince, J.A.A. & Sykes, G.M.H., 1993, "Integrating MRPII and JIT: a management rather than technical challenge", *International Journal of Operations and Production Management*, **13** (4), pp. 18-31.

Simulation Study Group, 1991, *Simulation in UK manufacturing industry*, The Management Consulting Group, University of Warwick Science Park, Coventry, UK.

Sinha, R.K. & Hollier, R.H., 1984, "A review of production control problems in cellular manufacture", *International Journal of Production Research*, **22** (5), pp. 773-789.

Slack, N., Chambers, S., Harland, C., Harrison, A. & Johnson, R., 1995, *Operations Management*, Pitman Publishing, London.

Sridharan, V., Newman, W.R., Chapman, S.M. & Jacobs, R.A., 1989, "Evaluating the interaction between the manufacturing environment and planning and control systems", *Journal of Manufacturing Systems*, 8 (4), pp. 329-332.

Suri, R. & Diehl, G.W., 1985, "MANUPLAN: a precursor to simulation for complex manufacturing systems", *Proceedings of the 1985 Winter Simulation Conference*, editors: Gantz, D., Blais, G, & Solomon, S., pp. 411-420.

Taylor, D.A., 1990, *Object-oriented technology: a managers guide*, Addison-Wesley Publishing Company, Reading, Massachusetts, US.

Thompson, W.B., 1994, "A tutorial for modeling with the WITNESS visual interactive simulator", *Proceedings of the 1994 Winter Simulation Conference*, editors: Tew, J.D., Manivannan, S., Sadowski, D.A. & Seila, A.F., pp. 517-521.

Thomson, V. & Graefe, U., 1989, "CIM - a manufacturing paradigm", *International Journal of Computer Integrated Manufacturing*, **2** (5), pp. 290-297.

Ting, J-J., 1991, *An object-oriented, distributed approach to the development of integrated manufacturing systems*, PhD Thesis, University of Michigan, US.

Tsubone, H., Matsuura, H. & Tsutu, T., 1991, "Hierarchical production planning system for a two stage process", *International Journal of Production Research*, **29** (4), pp. 769-785.

Turner, E. & Saunders, D., 1994, "Surveying the manufacturing planning and control environment", *Proceedings of the 4th International Conference on Factory 2000 - Advanced Factory Automation*, 3-5 October 1994, University of York, UK, IEE publication no. 398, pp. 613-619.

Ülgen, O.M. & Thomasma, T., 1990, "SmartSim: an object-oriented simulation program generator for manufacturing systems", *International Journal of Production Research*, **28** (9), pp. 1713-1730.

Ulrich, H. & Dürig, W., 1992, "Object-oriented realisation of planning tools for production planning and control", *Proceedings of the International Conference on Object-Oriented Manufacturing Systems*, editor: Norrie, D.H., 4-6 May, The University of Calgary, Alberta, Canada, pp. 158-167.

Umeda, S., 1992, "A manufacturing-oriented simulation package to support systems planning and its operation", *Proceedings of the 1992 Winter Simulation Conference*, editors: Swain, J.J., Goldsman, D., Crain, R.C. & Wilson, J.R., pp. 890-898.

Van der Linden, P.M.J. & Grünwald, H., 1980, "On the choice of a production control system", *International Journal of Production Research*, **18** (2), pp. 273-279.

Vollmann, T.E., Berry, W.L. & Whybark, D.C., 1992, *Manufacturing Planning and Control Systems* (3rd edition), Irwin.

Wainwright, C., Tucker, D. & Leonard, R., 1994, "Integrated manufacture for SME's", *Proceedings of the 10th National Conference on Manufacturing Research*, editors: Case, K. & Newman, S.T., 13-15 September, Loughborough University of Technology, UK, pp. 330-334.

Waldren, D., 1992, "What follows MRPII? Enterprise resource planning", *Professional Engineering*, May, **5** (5), pp. 22-23.

Wang, C-Y. & Fulton, R.E., 1994, "Information system design for optical fibre manufacturing using an object-oriented approach", *International Journal of Computer Integrated Manufacturing*, **7** (1), pp. 61-73.

Waterlow, J.G. & Monniot, J.P., 1986, *A study of the state of the art in computer-aided production management*, ACME Directorate, SERC.

Webster, J. & Williams, R., 1991, "Images and realities in the computerisation of managerial work: case studies of computer-aided production management", *Proceedings of the OMA-UK 6th International Conference: Achieving Competitive Edge*, editors: Bennett, D. & Lewis, C., 25-26 June, Aston University, pp. 115-120.

Wemmerlöv, U. & Hyer, N.L., 1989, "Cellular manufacturing in the U.S. industry: a survey of users", *International Journal of Production Research*, **27** (9), pp. 1511-1530.

Wharton, T.J. & Reid, R.D., 1990, "Manufacturing planning and control: how well are we doing?", *Production and Inventory Management Journal*, 3rd quarter, **31** (3), pp. 51-55.

Wight, O.W., 1981, *MPRII: unlocking America's productivity potential*, Oliver Wight Limited Publications, Boston, US.

Wild, R., 1989, *Production and operations management* (4th edition), Cassell Educational Ltd., UK.

Wisner, J.D., 1995, "A review of the order release policy research", *International Journal of Operations and Production Management*, **15** (6), pp. 25-40

Wu, B., 1994, *Manufacturing systems design and analysis: context and techniques* (2nd edition), Chapman & Hall, London.

Yourdon, E., 1990, "Auld lang syne", *BYTE*, **15** (10), pp. 257-263.

Yut, G.A. & Chang, T.C., 1994, "A five step object-oriented architecture for process planning", *International Journal of Production Research*, **32** (3), pp. 545-562.

Zäpfel, G. & Missbauer, H., 1993, "New concepts for production planning and control", *European Journal of Operations Research*, **67** (1993), pp. 297-320.

Appendix 1: Summary source code documentation

This appendix lists summary documentation for the planning and control class library, WBS/Control, at 31st August 1994. The documentation summarises that found in the Borland Pascal with Objects (v7.0) source code and which can be found in the appropriate *.PAS file. The format of the source code documentation follows that found with the files specific to the Advanced Factory Simulator (AFS); this provides both clarity and continuity between AFS and developed class libraries. The format used for documenting the software is as follows:

```
UNIT ExampleUnit
{ ------------------------------------------------------------

    TYPE:
      CLASS: tExampleClass
        or
      LIBRARY: tExampleLibrary
        or
      PROGRAM: tExampleProgram

    DATES:
      CREATED: njb : 01/01/94
      UPDATED: njb : 16/11/94

    SUMMARY:
        This is a summary of the different classes held within
      the UNIT and provides a brief description of the
      functionality offered by each class.  Important points
      to notice are included here.

        The body of the source code will be sufficiently
      documented (i.e. commented); a potential developer
      should be able to follow both the structure and logic of
      the code.

        Because the source code is likely to be modified, all
      (major) modifications are listed in the following
      section.  Comments are added appropriately to the body
      of the source code.

    DETAIL:
      05/01/94 : the following modification has been made to
                 tExampleClass ....

    ------------------------------------------------------------
}
   The source code follows ...
```

(note: a complete discussion of the classes found in the AFS architecture is included in Ball (1994).)

WBS/Control classes

The summary of each class of WBS/Control includes a list of the class attributes as well as its relationships to other classes: either through inheritance or the Borland Pascal USES clause. The class structures, hierarchies and relationships are further illustrated in Appendix 2. The methods of each class have not been included.

tPartSet

UNIT name : Part_Obj
parent class : Activity.tActivity
attributes
  cOwnClass
  cObjectName
  pPartList
uses
  tPartList
  tPartData
  tPCRoute
  tLotSize
class manager
  UNIT name : Part_Mgr
  class name : tPartObjMngr
  parent class : Actv_Mgr.tActivityMngr

This class combines with tPartList and tPartData to form a parts database; all the 'information' about individual parts which are included in a simulation model are entered, accessed and/or modified via this class. The information is used by other classes within the various material planning and control activities. This class is currently called tPart_Obj.

The class tPartObjMngr is the class manager to tPart_Obj: it manages the functions associated will all objects of the class tPart_Obj.

Note: the role of all class managers is to manage the functions associated with their respective class. These functions relate to load and save, text file upload as well as enabling communication to take place between the user (via the dialog in the user-interface) and the objects of the class.

tPartList

UNIT name : PartList
uses
  tPartData

This class holds a TSortedCollection of pointers to objects of type tPartData; the pointers are sorted by cPartName. An object of this class forms the link between an object of type tPartSet and the individual part 'records' represented by objects of type tPartData.

tPartData

UNIT name : PartData
attributes
  cPartName
  cPartOrigin
  nLowLevelCode
  nLeadTime
  nPlanningHorizon

    nSafetyStock
    nReOrderPoint
    pLotSizingPolicy
    nMinOrderQty
    pMtlPlanRoute
  uses
    tPCRoute
    tLotSize

The class tPartData records all the data attributes specific to individual parts included in a simulation model and organised within an object of class tPartSet.

tBom

UNIT name : Bom_Obj
parent class : Activity.tActivity
attributes
    cOwnClass
    cObjName
    pParentList
uses
    tParentList
    tParent
class manager
    UNIT name : Bom_Mgr
    class name : tBomObjMngr
    parent class : Actv_Mgr.tActivityMngr

This class represents the interface to a manufacturing bill of materials; all the parent child relationships are added and/or modified via this class. This class is currently called tBom_Obj.

tParentList

UNIT name : ParentList
uses
    tParent

This class holds a TSortedCollection of pointers to objects of class tParent; the list is sorted by cParentName. This list holds provides access to all of the part objects which are manufactured from components or raw materials.

tParent

UNIT name : Parent
attributes
    cParentName
uses
    tChildList
    tChild

The class tParent is responsible for all its child parts, i.e. components or raw materials.

tChildList

UNIT name : ChildList
uses
  tChild

This class holds TSortedCollection of pointers to objects of class tChild; the list is sorted according to cChildName.

tChild

UNIT name : Child
attributes
  cChildName
  nNumberOf

This class records the relationship between the parent part and the child, namely the number of child parts required to manufacture the parent part.

tStockData

UNIT name : Stock_Ob
parent class : tActivity
attributes
  cOwnClass
  cObjName
  bRecordTransactions
  pStkList
  pTransList
uses
  StkList
  StkItem
  TransList
  TransData
class manager
  UNIT name : Stock_Mgr
  tStockObjMngr
  parent class : Actv_Mgr.tActivityMngr

This class combines with the classes tStkList and tStkItem to form an inventory system; for all of the modelled parts, the stock data are entered and modified via an object of this class. Access to 'inventory' records can be made by the modeller via the dialog; simulation entities access (i.e. query and modify) the information via messages. This class is currently called tStock_Obj.

There is also a stock audit trail facility: recording all transactions for all of the parts held on the inventory system. This facility can be 'toggled' on or off; the default in 'on'. Access to the audit trail is via tStockData.

tStkList

UNIT name : StkList
uses
tStkItem

This class holds TSortedCollection of pointers to tStkItem; the pointers are sorted according to cPartNumber.

tStkItem

UNIT name : StkItem
attributes
cPartNumber
nOnHand
nDueIn
nAlloc
nProjectedOnHand

This class is essentially a stock record for cPartNumber: recording the standard inventory values. The methods to update any of these values are included in this class, however access to any inventory record must be made via an object of class tStockData.

tTransList

UNIT name : TransList
uses
tTransData

This class hold TSortedCollection of pointers to objects of type tTransData; the pointers are sorted by cTransactionDate.

tTransData

UNIT name : TransData
attributes
cTransactionDate
cTransactionTime
cPartNumber
cTransactionData
cTransactionType
cOnHandBefore
cTransactionQty
cOnHandBalance

Essentially this class represents a stock audit trail and records all transactions, i.e. stock movements, for all the parts held within the inventory system. This facility can be 'toggled' on or off; the default in 'on'. Access to the audit trail is possible only via tStockData.

tAbsOrder

UNIT name : AbsOrder
parent class : tActivity
attributes
  cOwnClass
  cObjName
  pOrderList
uses
  tOrderList
  tOrderData
class manager
  UNIT name : AbOrdMgr
  tAbsOrderMgr
  parent class : Actv_Mgr.tActivityMngr

This class is an abstract class and represents the common attributes and methods identified with the different 'order' types available in WBS/Control. Because it is an abstract class, no objects will ever exist since there are no real world equivalents.

tOrderList

UNIT name : OrderLst
uses
  tOrderData
  tMpsOrder
  tPlannedOrder

This class holds a TSortedCollection of pointers to objects of type tOrderData; the pointers are sorted by nDueDate and cPartNumber.

tOrderData

UNIT name : OrderData
attributes
  cOrderType
  cOrderNumber
  cPartNumber
  nDueDate
  nLaunchDate
  nOrderQty
  cOrderStatus

This class represents the basic order type and is the parent class to the more specific orders such as tMpsOrder.

tMps

UNIT name : Mps
parent class : tAbsOrder
attributes
  cOwnClass
  cObjName
  nNextMpsEntry
uses
  tAbsOrder
  tOrderList
  tMpsOrder
class manager
  UNIT name : Mps_Mgr
  tMpsMgr
  parent class : AbOrdMgr.tAbsOrderMgr

This class represents the set of master production schedule (MPS) orders. This class is visible to other objects in a simulation model and can send and receive messages. Access to individual MPS orders, in order to modify them, is made via an object of this class.

tMpsOrder

UNIT name : MpsOrder
parent class : OrderData.tOrderData
attributes
  cDemandType
  cDemandOrigin
uses
  tOrderData

This class represents individual MPS orders.

tSuggestedOrders

UNIT name : POrds_Obj
parent class : tAbsOrder
attributes
  bTrimPlannedOrders
uses
  tAbsOrder
  tOrderLst
  tPlannedOrder
class manager
  UNIT name : POrds_Mgr
  tPOrdsObjMngr
  parent class : AbOrdMgr.tAbsOrderMgr

This class represents the set of orders generated by the material planning activities (based on the configured pc-routing(s)). They are 'suggested' orders and will remain so until confirmed, either by an entity in the simulation model (e.g. a material planner) or by the modeller via the user-interface. Those suggested which are confirmed become works (or purchase) orders and are released to the shop floor (or on to the purchasing department).

Before the material planning processes take place, the previously suggested orders are deleted - a new set of planned (or suggested) orders are generated by the configured pc-routing(s). This class is currently called tPOrds_Obj.

tPlannedOrder

UNIT name : PlannedOrder
parent class : OrderData.tOrderData
attributes
    nDueInQty
    nLowLevelCode
    nMtlPlanningPasses
uses
    tOrderData

This class represents the individual suggested orders generated by the material planning activities; objects of this class inherit the data and behaviour of tOrderData.

tWPOrders

UNIT name : WPOrders
parent class : AbsOrder.tAbsOrder
attributes
    nNextOrderNumber
uses
    tAbsOrder
    tOrderLst
    tOrderData
class manager
    UNIT name : WPOrdsMg
    tWPOrdsMgr
    parent class : AbOrdMgr.tAbsOrderMgr

This class represents the set of works and purchase orders. The orders held within classes of this type will either be 'confirmed' suggested orders or orders loaded directly from a text file. The order quantities register with the inventory system in terms of 'due in' and 'allocated' for the appropriate parts and their components respectively.

The individual works (or purchase) orders are instances of the class tOrderData.

tDemand

UNIT name : Demand
parent class : Activity.tActivity
attributes
    cOwnClass
    cObjName

pDemandList
uses
tDemandList
tDemandData
class manager
UNIT name : DemandMgr
tDemandMgr
parent class : Actv_Mgr.tActivityMngr

This class combines with tDemandList and tDemandData to form the product demand spanning some or all of the simulation experiment. How well the modelled production unit (or factory) meets the demand placed upon it is recorded accordingly: this provides a record of delivery performance in terms of customer service.

tDemandList

UNIT name : DemandList
uses
tDemandData

This class holds a TSortedCollection of pointers to objects of class tDemandData; the pointers are sorted according cDueDate and cPartNumber.

tDemandData

UNIT name : DemandData
attributes
cPartNumber
nDueDate
nSaleDate
nOrderQty
cDemandStatus
nDaysLate

Objects of this class represent the individual elements of the demand pattern used in the simulation model.

tCalendar

UNIT name : Calendar
parent class : Activity.tActivity
attributes
cOwnClass
cObjName
ConfigStatus
cYearStartDate
nDaysPerPeriod
nHoursPerDay
pPeriodList
uses
tPeriodList

tPeriodData
class manager
UNIT name : CalMgr
tCalendarMgr
parent class : Actv_Mgr.tActivityMngr

This class combines with tPeriodList and tPeriodData to form a shop calendar. All the information about each working week or period is held within this class set. The shop calendar is configured via the dialog of this class. Access to all period information (c.f. tPeriodData class) is made via an object of this class.

tPeriodList

UNIT name : PeriodList
uses
tPeriodData

This class holds TSortedCollection of pointers to objects of class tPeriodData; the pointers are sorted according to cPeriodNumber.

tPeriodData

UNIT name : PeriodData
attributes
cPeriodNumber
cWeekCommencing
cEndOfWeek
nDaysAvailable
nDailyOverTime

This class defines the data and methods associated with the individual periods of a shop calendar, namely the available capacity in terms of hours. The period represents the week: Monday to Sunday during which manufacture can be planned. Each period is assigned a unique number as well as being defined by its dates. The production time available (default = 5 days) can be reduced for shut-downs and holidays or extended via the sanctioning of overtime.

tWcData

UNIT name : WcData
parent class : Activity.tActivity
attributes
cOwnClass
cObjName
pWcList
uses
tWkCentre
tWcList
class manager
UNIT name : WcDatMgr
tWcDataMgr

parent class : Actv_Mgr.tActivityMngr

This class combines with tWcList, tWkCentre, tWcLoading, tLoadinData, tMcList and tMcData to form a 'work centre information system'. It is within objects of these classes that the associated work centre information is located and where the results of any capacity analysis are held: for each work centre, for each period of the shop calendar.

Initially the data is derived from the appropriate shop floor objects, however it is intended to be able to introduce errors and discrepancies between the physical system and the information system. This will support the analysis of the effects of such errors.

tWcList

UNIT name : WcList
uses
  tWkCentre

This class holds a TSortedCollection of pointers to objects of class tWkCentre; the pointers are sorted according to cWcName.

tWkCentre

UNIT name : WkCentre
attributes
  WcWarmUpTm
  WcLoadTm
  WcUnLoadTm
  WcEfficiency
  cWcName
  pMachineList
  pWcLoading
uses
  tMach_Abs (AFS abstract machine class)
  tWorkCntr (AFS work centre class)
  tWorkStat (AFS work station class)
  tWcLoading
  tLoadingData
  tMcList
  tMcData

This class represents the data (and methods) associated with shop floor work centres. This information summarises all of the data of individual tMcData objects and is utilised by capacity planning objects.

tWcLoading

UNIT name : WcLoading
uses
  tLoadingData

This class holds a TSortedCollection of pointers to objects of class tLoadingData; the pointers are sorted according to cPeriodNumber.

tLoadingData

UNIT name : LoadingData
attributes
    cPeriodNumber
    nCurrentCapacityLoading
    nPlannedCapacityLoading
    nAvailableCapacity

This class represents the calculated loading for a work centre; the loading is determined by some form of configured capacity analysis and is based on the current suggested orders and works orders. (This would not necessarily apply to any rough-cut capacity planning.) The loading is determined for each period of the shop calendar.

tMcList

UNIT name : McList
uses
    tMcData

This class holds TCollection of pointers to objects of class tMcData.

tMcData

UNIT name : McData
attributes
    cMcNumber
    tmWarmUp
    tmLoad
    tmUnLoad
    nEfficiency

This class encapsulates the properties of machine (i.e. work station) data. This data is used by capacity planning objects when determining work centre loading.

Initially this data is derived from the shop floor objects themselves. However, it is possible to introduce errors into the information-based system and thereby be able to evaluate the effects of such errors: differences between the physical system and the information system are not uncommon in manufacturing.

tRouteData

UNIT name : RoutData
parent class : Activity.tActivity
attributes
    cOwnClass
    cObjName

pParts
uses
tPartNumb (AFS part number class)
tRouting (AFS routing class)
tParts
tPartRoute
class manager
UNIT name : RtDatMgr
tRouteDataMgr
parent class : Actv_Mgr.tActivityMngr

This class combines with tParts, tPartRoute, tOpList and tOpData to provide a 'routing information database'. This database is utilised by the material planning process (if configured to do so) when determining the capacity requirements for a particular set of suggested and works orders. Access to the 'routing' information is possible only via tRouteData.

At the start of the simulation run, the routing system is derived from the 'shop floor' routings. The provision of a routing database separate from the shop floor routings means that it will be possible to introduce 'errors and discrepancies' between the physical system and the information system: it will be possible to model the effects of such errors, which are common in manufacturing.

tParts

UNIT name : Parts
uses
tPartRoute

This class holds a TCollection of pointers to objects of tPartRoute

tPartRoute

UNIT name : PartRoute
attributes
cPartNumber
pOpList
uses
tRouting (AFS routing class)
tOpList
tOpData

This class encapsulates the part routing information for all of the parts modelled.

tOpList

UNIT name : OpList
uses
tOpData

This class holds a TSortedCollection of pointers to objects of class tOpData; the pointers are sorted by nOpNumber.

tOpData

UNIT name : OpData
attributes
  nOpNumber
  cOpType
  cWcName
  SetUpTm
  CycleTm

This class encapsulates the data and methods associated with individual operations on a part routing.

Initially this data is derived from the shop floor routing for the part, however it is the intention to be able to introduce errors and so be able to model the effects of differences between the physical system and the information system. This concept can be extended to many aspects of WBS/Control classes: for example, the BOM and the inventory system.

tAbstractMtlPlanning

parent class : Activity.tActivity
attributes
  cOwnClass
  nCurrentLowLevelCode
  bSetCurrentLowLevelCode
class manager
  UNIT name : AbMtlMgr
  tAbMtlMgr
  parent class : Actv_Mgr.tActivityMngr

This class represent the common data and behaviour of the low level classes which themselves represent elements of different planning and control systems. Objects of the low level classes are in essence 'actors' performing their respective operations on other classes. It is through such low level classes that the flexibility necessary to be able to model different planning and control systems is provided. Furthermore the application of WBS/Control to design issues is possible through the implicit flexibility of these classes.

This abstract class is the parent to all of the current low level classes and is currently called tAbsMtlPlanning.

tBatching

UNIT name : Batching
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : BatchMgr

tBatchingMgr
parent class : AbMtlMgr.tAbMtlMgr

This class 'batches' a suggested order quantity according to the lot sizing policy of the part number.

tCapReqPlan

UNIT name : Crp
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : CrpMgr
  tCapacityMgr
  parent class : AbMtlMgr.tAbMtlMgr

This class is an 'aggregate' (or high level) class which provides a less detailed representation of the capacity requirements planning (CRP) process. This is not a low level class.

Objects of class tCapReqPlan determine the (machine) capacity requirements for the derived set of suggested (works) orders as well as including the resource requirements for the current works orders: the result being the identification of the overall work centre loading. (Operators, or material planners, with the appropriate skills will be able make decisions about the suggested works orders based on an assessment of the work centre loading.)

tDemToMps

UNIT name : DemToMps
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : DemToMgr
  tDemToMgr
  parent class : AbMtlMgr.tAbMtlMgr

This class is a simple class which translates the current product demand (as defined by objects held within tDemand) into a future production programme, defined by the master production schedule (MPS).

tGenROLevls

UNIT name : GenROLevels
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : GenROLMgr
  tGenROLMgr
  parent class : AbMtlMgr.tAbMtlMgr

This class compares the current 'on hand' stock value for a part number with its reorder level and generates a suggested order if the 'on hand' value is below that of the reorder level.

tGenDemand

UNIT name : GenDemand
parent class : IncDepDem.tIncDepDem
class manager
    UNIT name : GenDemMgr
    tGenDemandMgr
    parent class : IncDep_M.tIncDepDemandMgr

This class is used within a DMRP control system configuration. DMRP is a control system based on MRPII which exploits the special circumstances of cellular manufacture.

It generates a demand, in terms of a MPS entry, for those components which are required for manufacture at one cell but which are themselves manufactured at a different cell.

tGrossToNet

UNIT name : GrossToNet
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
    UNIT name : GToN_Mgr
    tGrossToNetMgr
    parent class : AbMtlMgr.tAbMtlMgr

This class 'nets' the suggested order quantity against the current 'on hand' stock. This class is typically found in MRP-based models.

tIncDepDem

UNIT name : IncDepDem
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
    UNIT name : IncDep_Mgr
    tIncDepDemandMgr
    parent class : AbMtlMgr.tAbMtlMgr

This class ('include dependent demand') explodes the demand for a parent item into appropriate suggested orders for its component parts.

tInclDueIn

UNIT name : InclDueIn
parent class : AbsMtlPlanning.tAbsMtlPlanning
attributes
    bOrderComplete
    nEarliestDate
    nCurrentDate
    cCurrentPartNumber
class manager

UNIT name : InclDu_M
tInclDueInMgr
parent class : AbMtlMgr.tAbMtlMgr

This class adds any 'due in' orders (i.e. confirmed works or purchase orders) to the current 'on hand' stock value on the date the order is expected.

tOffSet

UNIT name : OffSet
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : OffSet_M
  tOffSetMgr
  parent class : AbMtlMgr.tAbMtlMgr

This class 'offsets' the launch date of a suggested order by the defined lead-time of the part.

tOnset

UNIT name : OnSet
parent class : AbsMtlPlanning.tAbsMtlPlanning
class manager
  UNIT name : OnSetMgr
  tOnSetMgr
  parent class : AbMtlMgr.tAbMtlMgr

This class 'onsets' the due date of a suggested order by the defined lead-time of the part.

tStdDmrp

UNIT name : StdDmrp
parent class : StdMrp.tStdMrp
attributes
  pStdDmrpConfig
uses
  tStdMrp
  tGenDemand
class manager
  UNIT name : StdDmrpMgr
  tStdDmrpMgr
  parent class : StdMrp_M.StdMrpMgr

This is a pre-configured set which represents a basic DMRP planning and control system. An object of this class would relieve the modeller of the necessity to configure this particular control system.

This class is a child of the class tStdMrp.

tStdMrp

UNIT name : StdMrp
parent class : AbsMtlPlanning.tAbsMtlPlanning
attributes
    pStdMrpConfig
uses
    tInclDuIn
    tGrossToNet
    tBatching
    tOffSet
    tIncDepDem
class manager
    UNIT name : StdMrp_M
    tStdMrpMgr
    parent class : AbMtlMgr.tAbMtlMgr

This class is a pre-configured set which represents a basic MRP model. A pc-routing containing an object of this class is equivalent to one which contains all of the required low level objects, the only difference being this class automatically configures the required low level objects.

This class is a parent of tStdDmrp.

tStdRop

UNIT name : StdRop
parent class : AbsMtlPlanning.tAbsMtlPlanning
attributes
    pStdRopConfig
uses
    tGenROLevels
    tBatching
    tOnSet
class manager
    UNIT name : StdRop_M
    tStdRopMgr
    parent class : AbMtlMgr.tAbMtlMgr

This class supports a standard ROP system configuration: an object of this class automatically configures the required low level objects to represent a system of this kind.

tLotSize

UNIT name : LotSize
parent class : Activity.tActivity
attributes
    cOwnClass
class manager
    UNIT name : LotSzMgr

tLotSzMgr
parent class : Actv_Mgr.tActivityMngr

This class is an abstract lot size (batching rule) class and encapsulates the common data and methods of the batching rules currently available in WBS/Control. This class is a parent class and will not exist as an object.

tLotForLot

UNIT name : Lot4Lot
currently called tLot4Lot
parent class : LotSize.tLotSize
class manager
  UNIT name : LForLMgr
  tLotForLotMgr
  parent class : LotSzMgr.tLotSzMgr

This class represents the lot-for-lot batching policy.

tEOQ

UNIT name : Eoq
parent class : LotSize.tLotSize
attributes
  nEOQ
  nAvgDemand
  nUnitCost
  nCarryCharge
  nOrderCost
  nHoldingCost
class manager
  UNIT name : EoqMgr
  tEoqMgr
  parent class : LotSzMgr.tLotSzMgr

This class determines a batch quantity based on the standard economic order quantity calculation.

tFixedQty

UNIT name : FixedQty
parent class : LotSize.tLotSize
class manager
  UNIT name : FOrdMgr
  tFixedOrderQtyMgr
  parent class : LotSzMgr.tLotSzMgr

This class determines a lot size based on a 'fixed' order quantity which set by the modeller: the determined lot size is the greatest of either the initial order quantity or the fixed order quantity.

tMultiOrder

UNIT name : MultiOrd
parent class : LotSize.tLotSize
class manager
  UNIT name : MultiMgr
  tMultiOrdMgr
  parent class : LotSzMgr.tLotSzMgr

This class determines the lot size based on multiples of a set order quantity.

tPcRoute

UNIT name : PCRoute
parent class : Activity.tAcivity
attributes
  cOwnClass
  cObjName
  bConfigured
  nPlanningDuration
  pActiveList
  pCapList
uses
  tAbsMtlPlanning
  tActiveList
  tCapList
class manager
  UNIT name : PCRout_M
  tPCRouteMgr
  parent class : Actv_Mgr.tActivityMngr

This class defines the planning and control routing, i.e. the pc-routing, to which the low level objects are configured. The duration of the material planning activity is set with this class. It is the pc-routing concept and subsequent implementation which supports the necessary flexibility of WBS/Control. The material and capacity planning activities are performed via the pc-routing.

tActiveList

UNIT name : ActiveList
attributes
  nTotalMtlPlanningObjects
uses
  tAbsMtlPlanning

This class holds a TCollection of pointers to objects of type tAbsMtlPlanning, i.e. the low level objects representing the material planning activities are held on this list.

tCapList

UNIT name : CapList
uses
  tAbsMtlPlanning

This class holds a TCollection of pointers to objects of class tAbsMtlPlanning; these objects are specifically capacity related (as opposed to material planning, these are organised on tActiveList). This class is under development.

tAbstractPlanningSkill

UNIT name : AbPlanSk
currently called tAbPlanSk
parent class : SkillAbs.tAbsSkill
attributes
  pTaskList
  pOwner
  pPlanningRoutingList
  bConfiguredToAllRoutings : BOOLEAN;
  bSetTimeOfPlanning : BOOLEAN;
  bPerformedPlanning : BOOLEAN;
  tmPerformPlanning : OpDate.tTime;
  dtPerformPlanning : OpDate.tDate;
  nPlanningFrequency : INTEGER;
  cFrequencyType
uses
  tSkillAbs (AFS abstract skill)
  tPCRoute
class manager
  UNIT name : AbPlnSkMgr
  tAbPlnSkMgr
  parent class : SkAb_mgr.tAbsSkillMngr

This abstract class encapsulates the common methods and behaviour associated with the two current planning-type skills: tPlanning and tMasterSchedulingSkill. This class inherits the more general 'skill' properties from the AFS class tAbsSkill.

In essence, the skills developed for WBS/Control define the 'softer side' of the respective material planning activities. Skill are assigned to operators and thereby provide the operator with the appropriate 'know-how' to perform certain tasks. The planning related skills define when the planning activity (defined by the assigned pc-routing(s)) will take place and with what frequency: for example, an operator could be assigned to a MRP routing and set to action it every 7 days at 09.00 hrs.

tPlanningSkill

UNIT name : Planning
parent class : AbPlanSk.tAbPlanSk
attributes
  cOwnClass
uses
  tPCRoute

- class manager
    - UNIT name : PlanMgr
    - tPlanningMgr
    - parent class : AbPlnSkMgr.tAbPlnSkMgr

tMasterSchedulingSkill

- UNIT name : MastSched
- currently called tMastSched
- parent class : AbPlanSk.tAbPlanSk
- attributes
    - cOwnClass
- uses
    - tPCRoute
- class manager
    - UNIT name : SchedMgr
    - tSchedulingMgr
    - parent class : AbPlnSkMgr.tAbPlnSkMgr

tStoresMgtSkill

- UNIT name : StoresMgt
- parent class : SkillAbs.tAbsSkill
- attributes
    - cOwnClass
    - pTaskList
    - pOwner
    - bSetInitialTime
    - tmCheckinTray
    - dtCheckInTray
    - nNumberOrdersToBookInNext
    - tmToProcessGRN
    - nBatchQty
    - nCheckFrequency
    - cFrequencyType
- uses
    - tSkillAbs (AFS abstract skill class)
- class manager
    - UNIT name : Smgt_mgr
    - tStoresMgtMgr
    - parent class : SkAb_mgr.tAbsSkillMngr

This skill provides the capability to undertake basic stores management activities: an operator with this skill is able to book parts into and out of stock based on completed works orders. The 'policy side' to this activity is included: the frequency of when a stores person checks their 'goods in' tray is defined via the dialog of this skill as well as the number of orders 'booked in' at one time. The duration of the activity is determined by the number of orders to book in and the time per order: this data is also set via the dialog.

tSalesSkill

- UNIT name : Sales
- parent class : SkillAbs.tAbsSkill
- attributes
  - cOwnClass
  - pOwner
  - bSetTimeOfSales
  - tmPerformSales
  - dtPerformSales
- uses
  - tSkillAbs, (AFS abstract skill class)
  - tPartNumb (AFS part number class)
  - tStores (AFS physical stores i.e. CentralStores)
  - tStock_Ob
- class manager
  - UNIT name : SalesMgr
  - tSalesMgr
  - parent class : SkAb_mgr.tAbsSkillMngr

This class is currently called tSales and is under development. In essence, this skill makes a sale based on the product demand defined in the tDemand object of the model. This is achieved by the sales person (i.e. an operator assigned with 'sales' skills), at the appropriate demand due dates, comparing the inventory status for the parts and providing sufficient stock are shown to be available, the demand is met - the stocks (both physical and inventory records) are adjusted accordingly.

tMngrCtl

This UNIT declares all of the WBS/Control class managers within the wider AFS architecture. There exists a similar UNIT which declares all of the class managers associated directly with the shop floor functionality provided by AFS.

Appendix 2: WBS/Control class structures, hierarchies and relationships

The following figures represent the class structures, hierarchies and relationships of the different classes currently found in WBS/Control.

Figure A2.1 illustrates the key which applies to all of the subsequent figures.

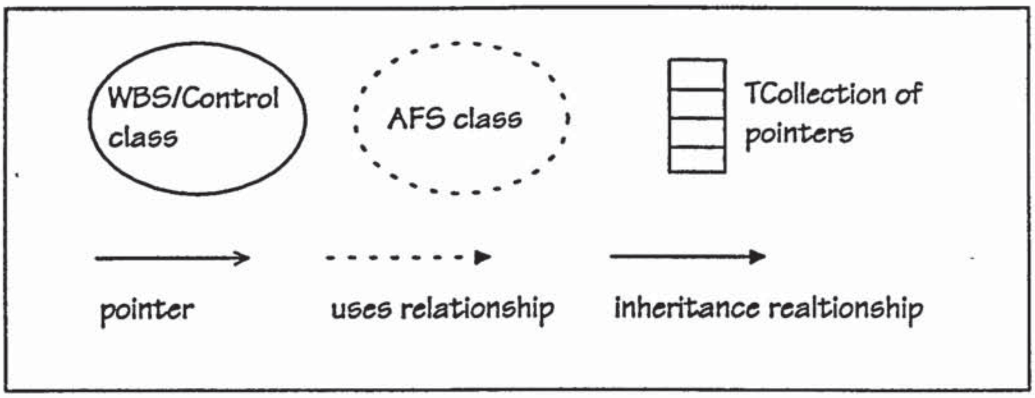


Figure A2.1: Key to figures.

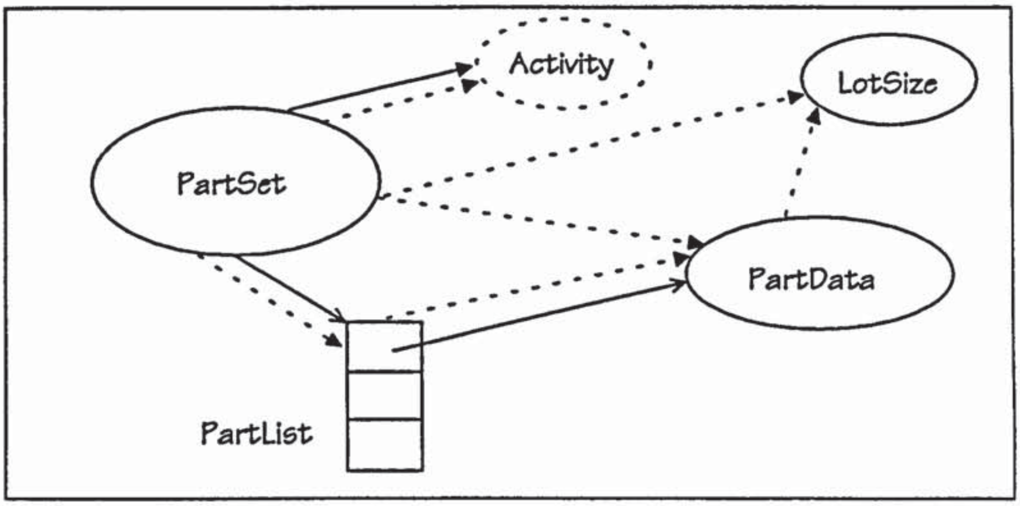


Figure A2.2: *PartSet*, *PartList* and *PartData*.

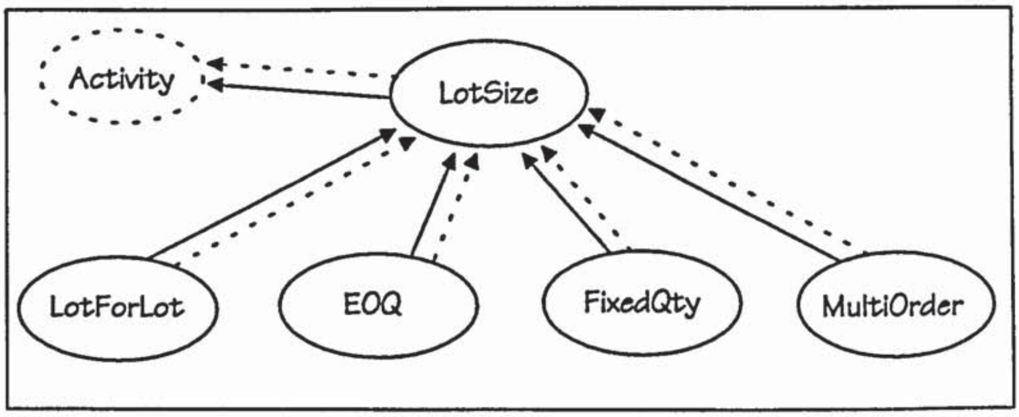


Figure A2.3: *LotSize*.

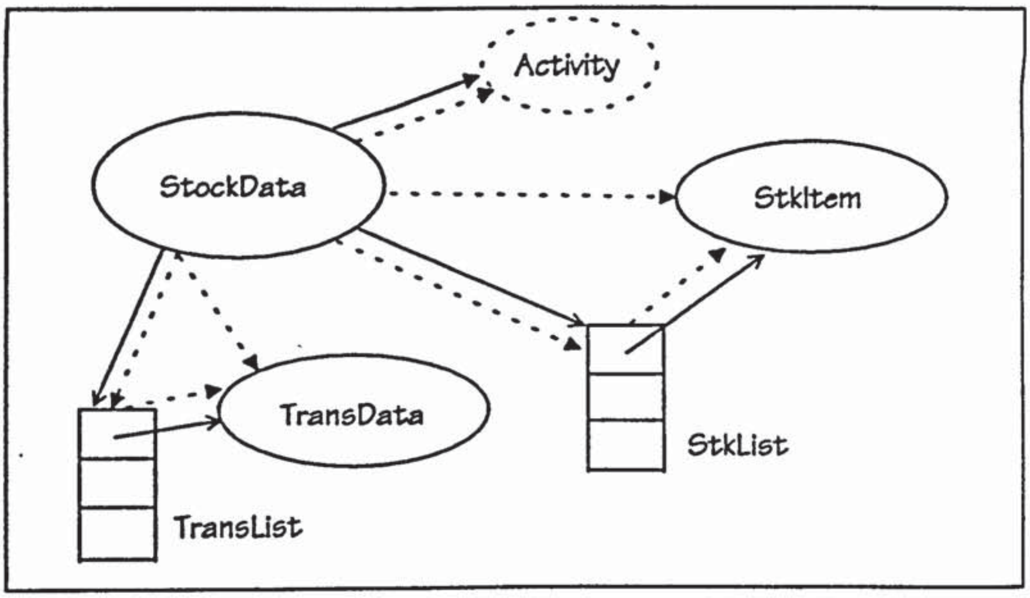


Figure A2.4: *StockData, StkList, StkItem, TransList* and *TransData.*

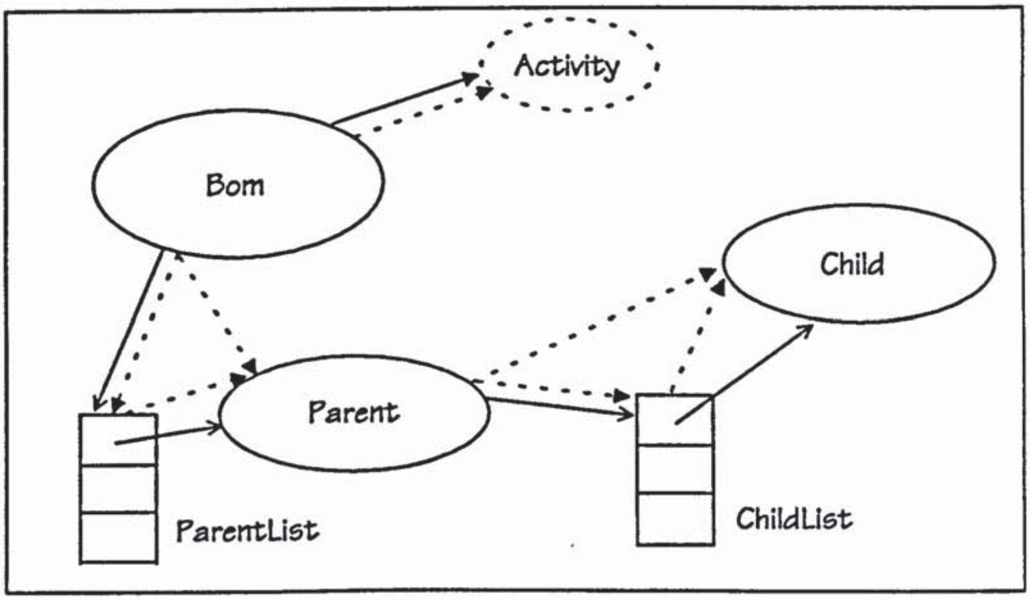


Figure A2.5: *Bom, ParentList, Parent, ChildList* and *Child.*

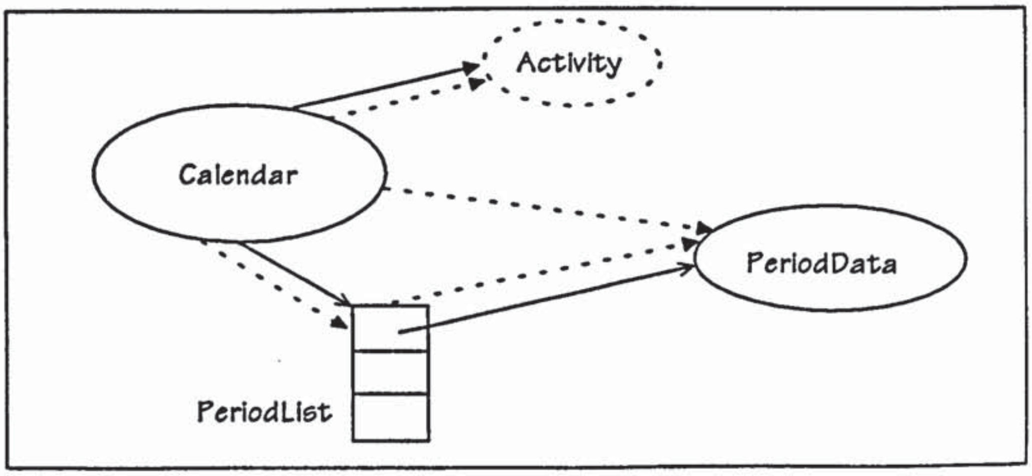


Figure A2.6: *Calendar, PeriodList* and *PeriodData.*

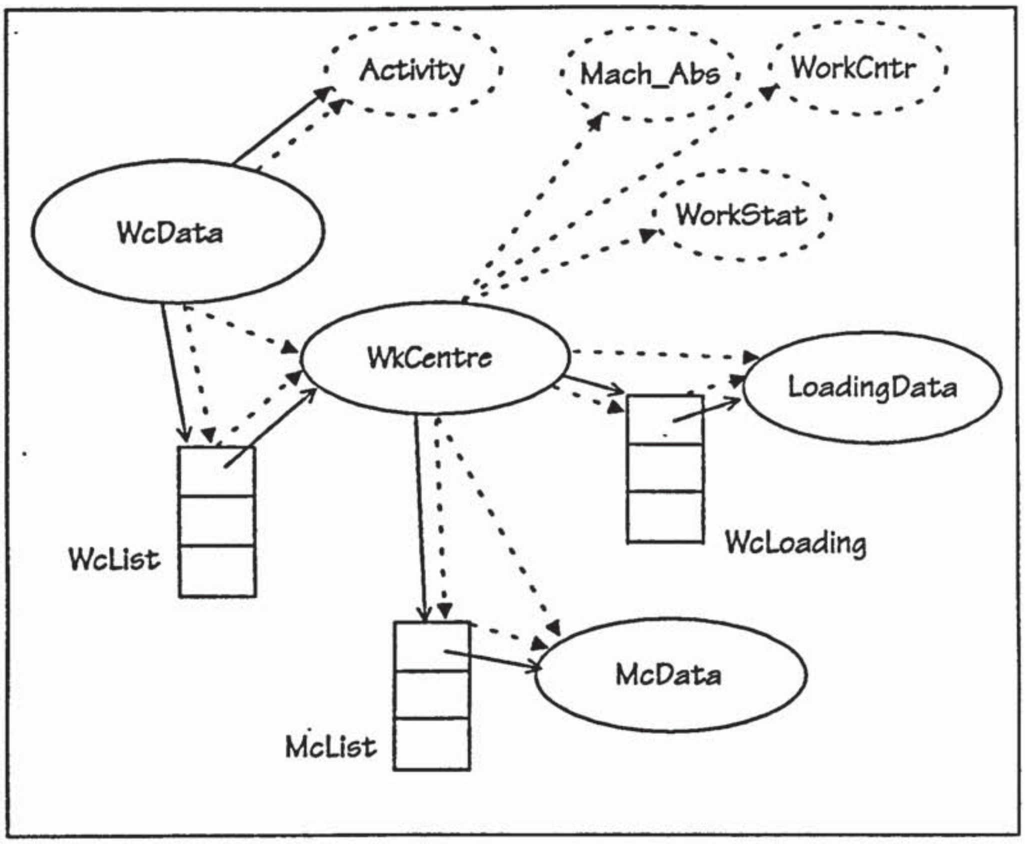


Figure A2.7: *WcData, WcList, WkCentre, WcLoading, LoadingData, McList* and *McData.*

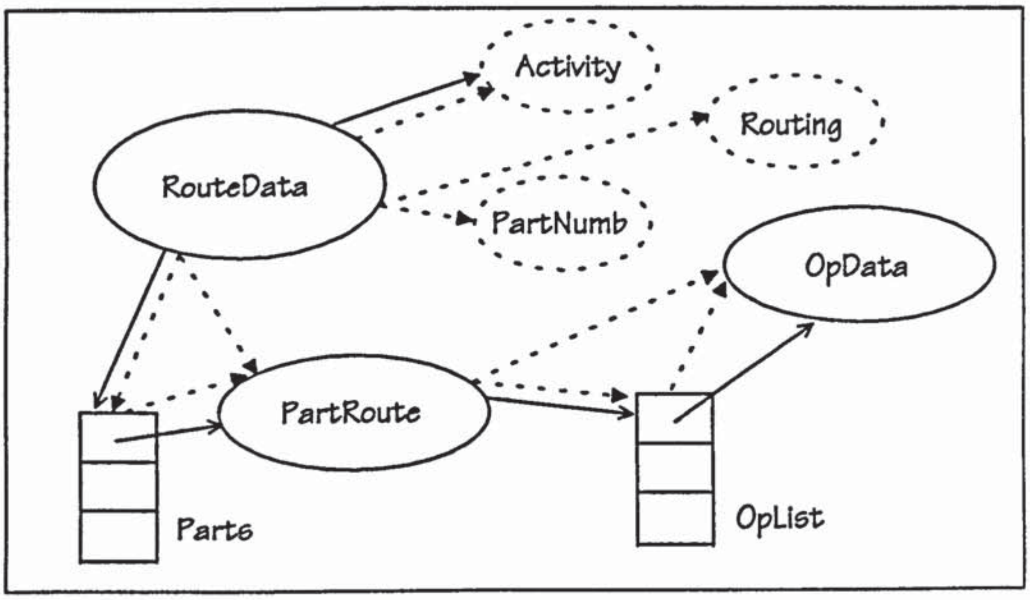


Figure A2.8: *RouteData, Parts, PartRoute, OpList* and *OpData.*

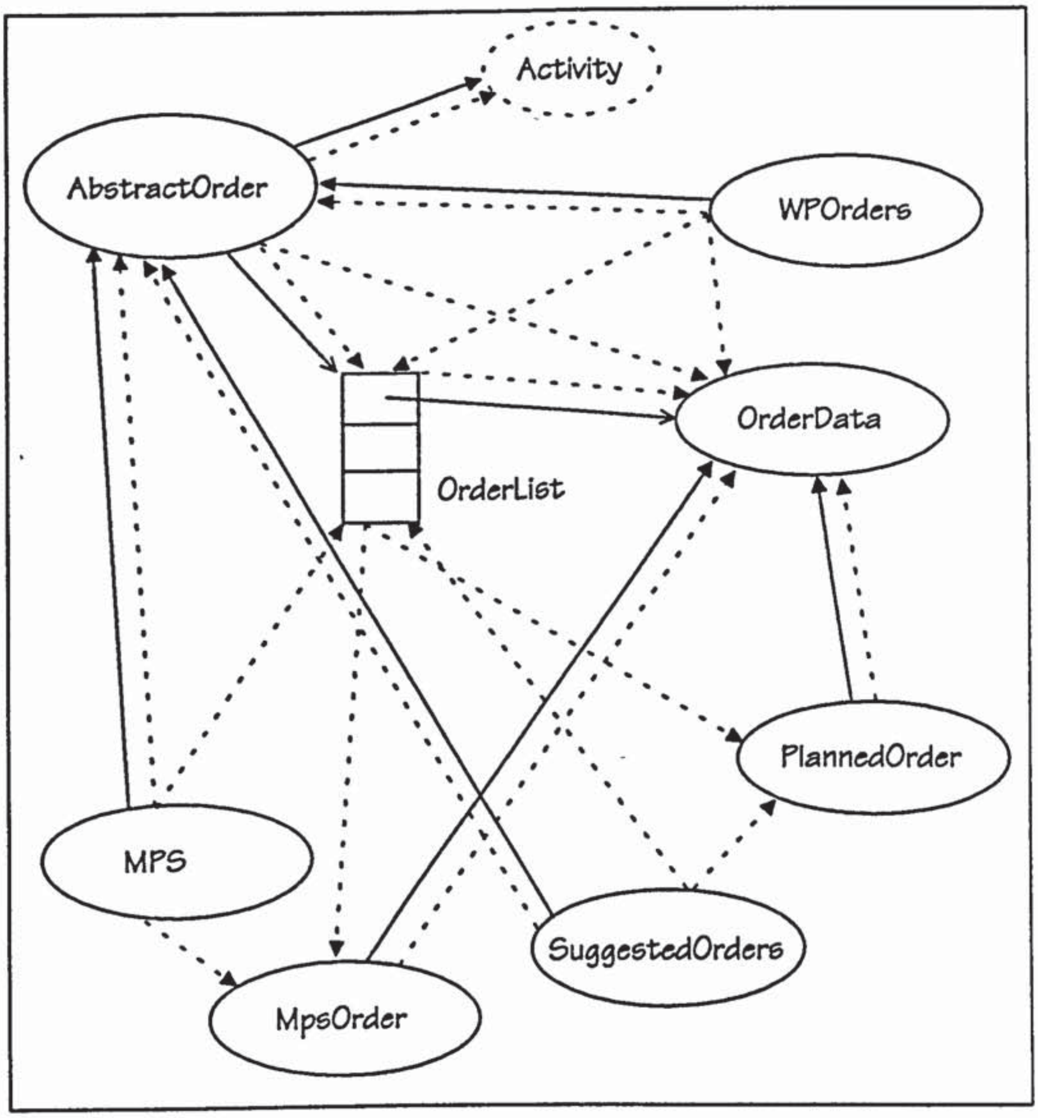


Figure A2.9: *Mps*, *SuggestedOrders* and *WorksAndPurchaseOrders*.

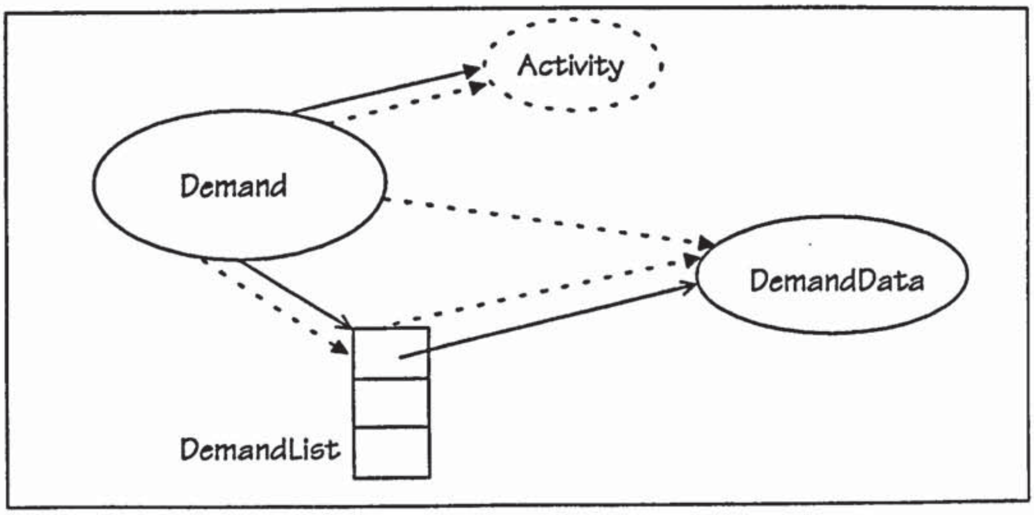


Figure A2.10: *Demand*, *DemandList* and *DemandData*.

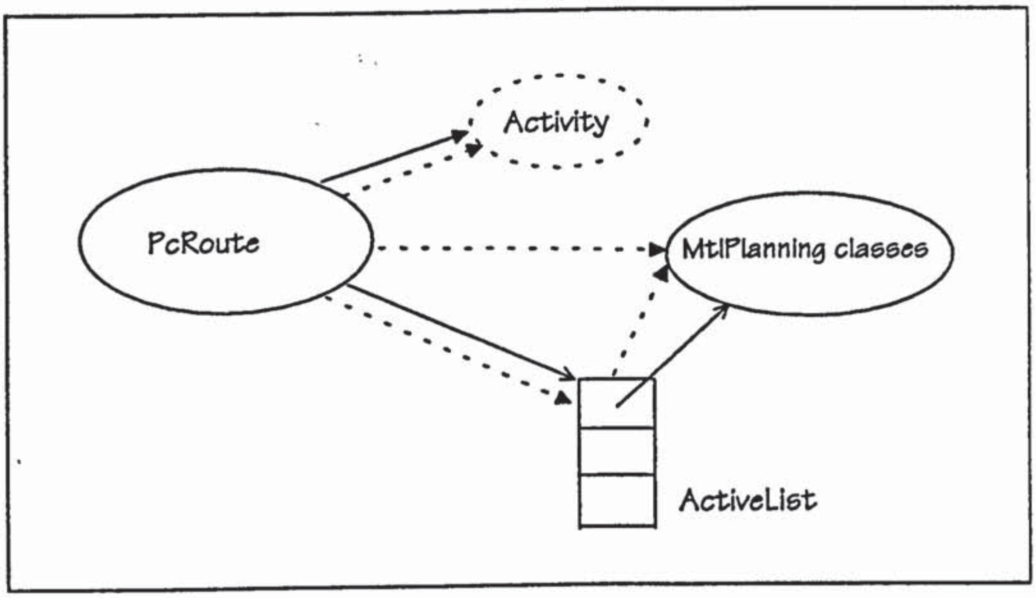


Figure A2.11: *PcRoute* and *ActiveList*.

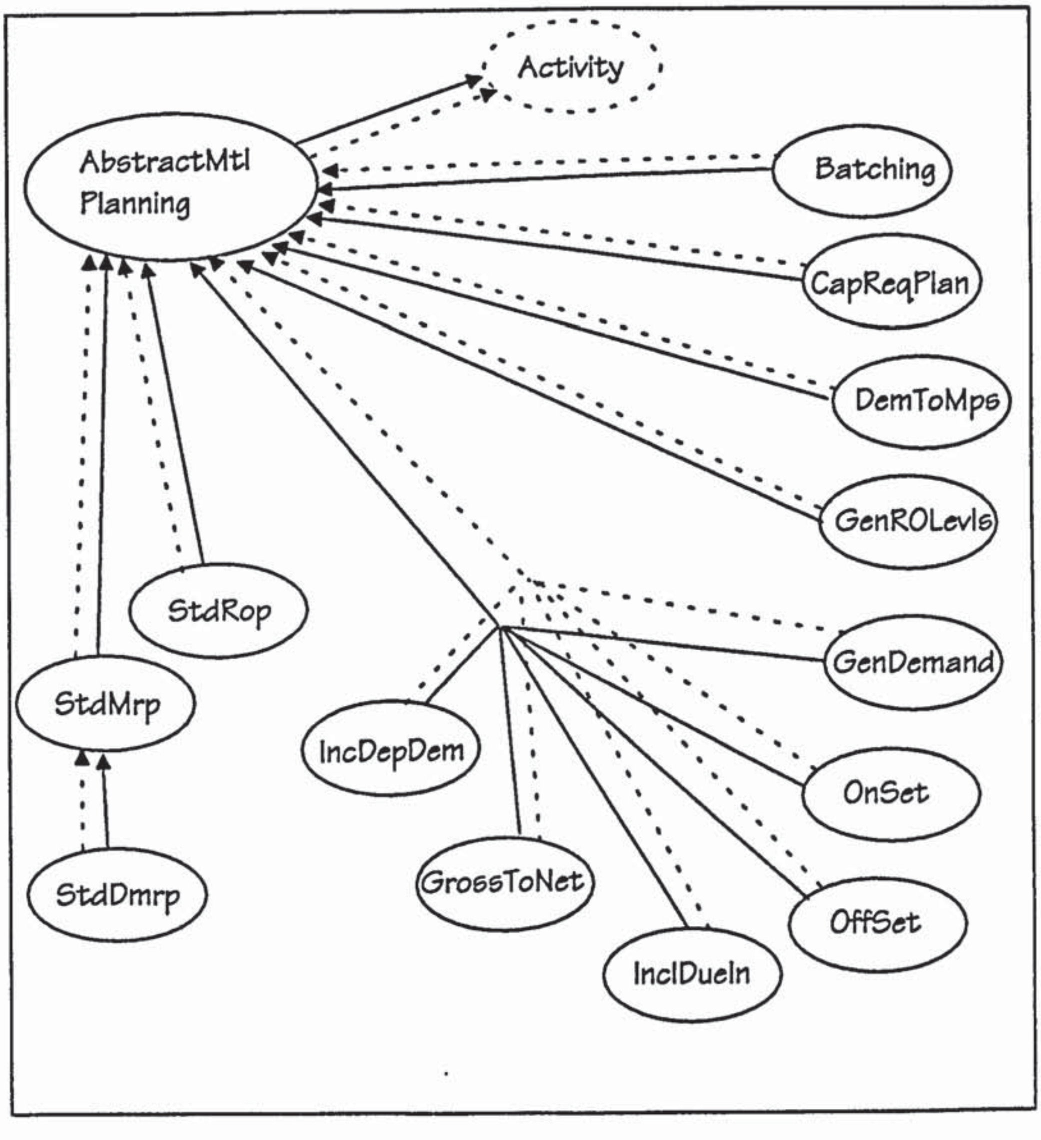


Figure A2.12: Material planning classes.

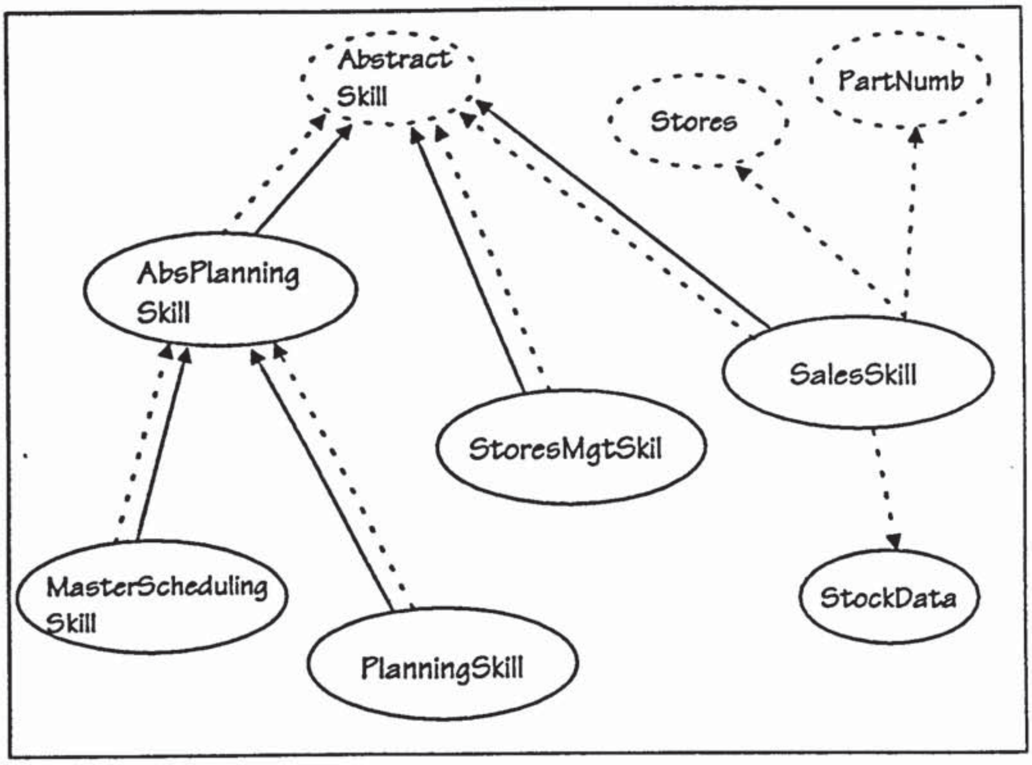


Figure A2.13: WBS/Control skills.

Appendix 3: Commercial software

The following table details the commercial software which has been utilised for the development of WBS/Control and the Advanced Factory Simulator (AFS).

Application	**Commercial Software**	**Version**	**Supplier**
operating environment	Windows	3.1	Microsoft Corporation. UK Office: Winash, Wokingham, Berks, RG11 1TP.
WBS/Control, AFS and graphics	Borland Pascal with Objects	7.0	Borland International Inc. 1800 Green Hills Road, PO Box 66001, Scotts Valley, CA, USA.
user-interface	Visual Basic	3.0	Microsoft Corporation. UK Office: Winash, Wokingham, Berks, RG11 1TP.
results presentation	Graphics Server	2.2	Bits Per Second Ltd. Regent Hill, Brighton, BN1 3ED, UK.
date and time utilities	Object Professional	1.12	Turbo Power Software. PO Box 66747, Scotts Valley, CA, USA.

(**note:** with the exception of the addition of WBS/Control, the above requirements are equivalent to those documented in Ball (1994).)

Appendix 4: Hardware and software requirements

The following details the hardware and software requirements for developing and using the planning and control class library WBS/Control along with the Advanced Factory Simulator (AFS). The hardware requirements represent the recommended minimum.

Hardware

Hardware type	IBM compatible pc
processor specification	486, 25 MHz
video	Super VGA
memory (RAM)	4.0Mb
hard disk space - development	12.5Mb
hard disk space - operation	2.5Mb

The hard disk memory requirements specified above is AFS, WBS/Control related: it does not include the installation and use of the development environments, i.e. Borland Pascal with Objects (v7.0) and Microsoft's Visual Basic (v3.0).

Software - Development

The following commercial software is required for the development of AFS and WBS/Control:

Application	Software
user-interface	Visual Basic (v3.0); Graphics server (v2.2)
WBS/control, AFS	Borland Pascal with Objects (v7.0)
graphics	Borland Pascal with Objects (v7.0)
help system	Microsoft Word; Microsoft help compiler

The required AFS and WBS/Control source files will also need to be installed. The files required will depend upon the aspect of the AFS system and/or WBS/Control which is being modified; the files will include both Borland Pascal and Visual Basic development files.

Software - Operation

The following commercial software is required for the operation of AFS and WBS/Control:

Software	Required Files
Graphics Server (v2.2)	gsw.exe, gswdll.dll, graph.vbx
Visual Basic (v3.0)	vbrun300.dll, cmdialog.vbx, msmasked.vbx, spin.vbx, grid.vbx, gauge.vbx
Borland Pascal with Objects (v7.0)	ddeml.dll

The files listed above will need to be placed in the 'c:\windows\system' directory; these files are not necessary, however, if the development system has already been installed. (note: drive C is assumed to be the local hard drive.)

The following AFS and WBS/Control files are required for operation:

File Name	Description	Directory
afscore.exe	simulator application	c:\afs\system
afsintf.exe	user-interface application	c:\afs\system
afsgraph.exe	graphical display application	c:\afs\system
icondll.dll	icon DLL for graphics	c:\afs\system
config.txt	configuration file to link classes	c:\afs\system
afshelp.hlp	AFS help file, used by Windows help	c:\afs\help
res*.bmp	user-interface bitmaps	c:\afs\bitmaps

(note: with the exception of the addition of WBS/Control, the above requirements are equivalent to those documented in Ball (1994).)

Appendix 5: Modelling using WBS/Control

This section will describe three planning and control system models which have been constructed from the current functionality offered by WBS/Control; these models are wholly integrated with a shop floor model configured from the functionality supported by the Advanced Factory Simulator (AFS). In all cases it is the planning and control system model which drives production.

The purpose of these models is to demonstrate that it is possible to build planning and control system models in the way described in this thesis. Furthermore, it is intended to illustrate the salient features of the class library, namely its inherent flexibility, its ability to model across the planning and control hierarchy and its ability to model more than the control algorithms.

The model

The same shop floor model has been used with each of the three planning and control system models. This provides an opportunity to model three different situations where the only essential difference is the type of planning and control system employed.

Figure A5.1 illustrates the model. The department (Dept01) contains four work centres, *wc01*, *wc02*, *wc03* and *wc04*, each containing a single work station: *wc01*, *wc02* and *wc03* containing manual machines and *wc04* containing a manual assembly machine. Operators have been assigned to each work station; each operator is assigned the appropriate skill: *Op01*, *Op02* and *Op03* can operate the respective manual machines and are assigned a 'single machine' skill, whereas *Op04* has been assigned the 'assembly' skill in order to operate the *AssyManual01* machine. Also included in the model are a central stores, a cell leader, a material handler and a hand truck.

The model includes planning and control related personnel, namely a material planner, a master scheduler, a stores person and a sales person.

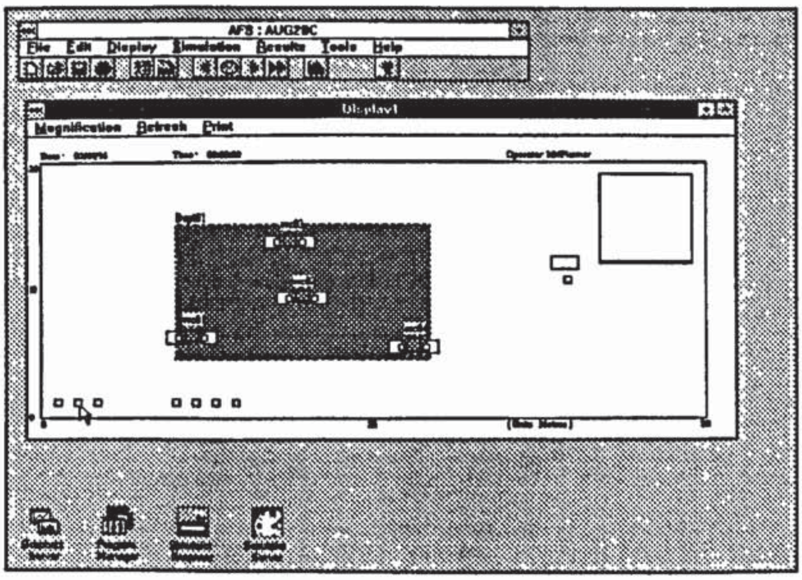


Figure A5.1: The core model.

The model represents a production unit which manufactures four parts: one end item and three components. The bill of materials is illustrated in Figure A5.2.

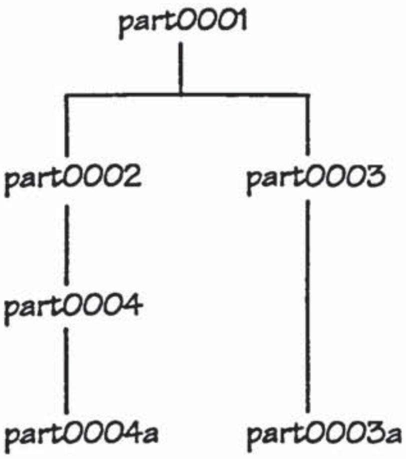


Figure A5.2: The product structure.

The raw materials 'part0003a' and 'part0004a' are used to manufacture parts 'part0003' and 'part0004' respectively. The parent-child relationships are all one-to-one except for 'part0001' and 'part0002': two of 'part0002' are required for every one 'part0001'.

The same initial conditions are used for each of the models and are listed below:

	DEMAND	
part number	due date	order qty
part0001	07/01/94	10
part0001	14/01/94	25

	PART DETAILS		
part number	lead time	order policy	reorder point
part0001	1 day	lot-for-lot	25
part0002	1 day	lot-for-lot	25
part0003	3 days	lot-for-lot	25
part0004	1 day	lot-for-lot	25

	STOCK STATUS		
part number	on hand	due in	allocated
part0001	0	0	0
part0002	0	0	0
part0003	0	0	0
part0004	0	0	0
part0003a	1000	0	0
part0004a	1000	0	0

As the above initial conditions will indicate, the production unit does not have any outstanding orders (i.e. work-in-progress) and capacity is available.

Model A

The planning and control system employed in this model is a reorder point (ROP) control system. The pc-routing representing this system was configured from the following low level classes: *IncDueIn*, *GenROLevls*, *Batching* and *OnSet*.

There are two sub-models included here. The first performs the material planning, based on the ROP model, every working day at 08.15 hrs; the second utilises the same ROP pc-routing but the material planning activities are performed at 08.15 and 13.15 hrs (daily). In both cases the demand pattern is interpreted as 'sales from stock' which is made at 15.00 hrs of the due date.

The resulting production activity is summarised below:

Model A (i): ROP, perform material planning x1 per day:-

DATE	ORDERS LAUNCHED (part number, quantity)	ORDERS COMPLETE (part number, quantity)	COMMENT
03/01/94	part0001, 25		
	part0002, 25	part0002, 25	
	part0003, 25	part0003, 25	
	part0004, 25	part0004, 25	
04/01/94	part0002, 50		
	part0003, 25	part0003, 25	
	part0004, 25	part0004, 25	
05/01/94	part0004, 50	part0001, 25	
		part0002, 50	
		part0004, 50	
06/01/94			no activity: no orders generated
07/01/94			sale made: part0001, 10
8-9/01/94			no activity: weekend
10/01/94	part0001, 10	part0001, 10	
11/01/94	part0002, 20	part0002, 20	
	part0003, 10	part0003, 10	
12/01/94	part0004, 20	part0004, 20	
13/01/94			no activity: no orders generated
14/01/94			sale made: part0001, 25
15-16/01/94			no activity: weekend
17/01/94	part0001, 25		
18/01/94	part0002, 50		

	part0003, 25	part0003, 25	
19/01/94	part0004, 50	part0001, 25	
		part0002, 50	
		part0004, 50	

Model A(ii): ROP, perform material planning x2 per day:-

DATE	ORDERS LAUNCHED (part number, quantity)	ORDERS COMPLETE (part number, quantity)	COMMENT
03/01/94	part0001, 25		
	part0002, 25	part0002, 25	
	part0003, 25	part0003, 25	
	part0004, 25	part0004, 25	
	part0002, 50		
	part0003, 25	part0003, 25	
	part0004, 25	part0004, 25	
04/01/94	part0004, 50	part0001, 25	
		part0002, 50	
		part0004, 50	
5-6/01/94			no activity: no orders generated
07/01/94			sale made: part0001, 10
8-9/01/94			no activity: weekend
10/01/94	part0001, 10	part0001, 10	
	part0002, 20	part0002, 20	
	part0003, 10	part0003, 10	
11/01/94	part0004, 20	part0004, 20	
12-13/01/94			no activity: no orders generated
14/01/94			sale made: part0001, 25
15-16/01/94			no activity: weekend
17/01/94	part0001, 25		
	part0002, 50		
	part0003, 25	part0003, 25	
18/01/94	part0004, 50	part0001, 25	
		part0002, 50	
		part0004, 50	

The results from these two runs are as would be expected for the type of models described: model A (ii) achieves the same end result in terms of on hand stocks but does so in less time and results in more days with 'no activity'. The final inventory position being 'on hand' quantities of 25 for all part numbers. Essentially, the two models illustrate a change of planning policy.

Model B

This model employs a MRP-based planning and control system model. The pc-routing is configured using the following low level classes: *IncDueIn*, *GrossToNet*, *Batching*, *OffSet* and *InclDepDem*. The initial demand is translated into a MPS by the master scheduler. The frequency of the scheduling and planning activities is daily at 08.15 hrs and 08.30 hrs respectively.

The following summarises the production activities:

DATE	ORDERS LAUNCHED (part number, quantity)	ORDERS COMPLETE (part number, quantity)	COMMENT
03/01/94	part0003, 10	part0003, 10	
04/01/94	part0004, 20	part0004, 20	
05/01/94	part0002, 20	part0002, 20	
06/01/94	part0001, 10	part0001, 10	
07/01/94			demand met: part0001, 10
8-9/01/94			no activity: weekend
10/01/94	part0003, 25	part0003, 25	
11/01/94	part0004, 50	part0004, 50	
12/01/94	part0002, 50	part0002, 50	
13/01/94	part0001, 25	part0001, 25	
14/01/94			demand met: part0001, 25
15-16/01/94			no activity: weekend
17/01/94			no activity: no orders suggested by MRP system model

The results are again what one would expect based on a model of this kind. Based on a lot-for-lot order policy and with the material planning being performed daily, the orders were released daily for the quantity required. The final inventory position being zero 'on hand' for all part numbers.

Model C

This model represents a simple hybrid planning and control system: a combination of MRP and ROP. Within the model, two planning routings (pc-routing) are included: one is configured to the ROP system described in model A and the second is configured to the MRP model described in model B. A third pc-routing exists which translates the demand into the MPS; this pc-routing is utilised by the master scheduler.

The material requirements for parts 'part0003' and 'part0004' are determined by the ROP system whereas the requirements for parts 'part0001' and 'part0002' are determined by the MRP model. The planning frequency for both approaches is daily at 08.30 hrs (MRP) and 08.35 hrs (ROP).

The following summarises the production activity generated by this combination of material planning systems:

DATE	ORDERS LAUNCHED (part number, quantity)	ORDERS COMPLETE (part number, quantity)	COMMENT
03/01/94	part0003, 25	part0003, 25	ROP generated orders
	part0004, 25	part0004, 25	
04/01/94			no activity: no orders suggested
05/01/94	part0002, 20	part0002, 20	MRP generated order
	part0004, 20	part0004, 20	ROP generated order
06/01/94	part0001, 10	part0001, 10	MRP generated order
	part0003, 10	part0003, 10	ROP generated order
07/01/94			demand met: part0001, 10
8-9/01/94			no activity: weekend
10/01/94			no activity: no orders suggested
11/01/94			no activity: no orders suggested
12/01/94	part0002, 50	part0002, 50	MRP generated order
	part0004, 50	part0004, 50	ROP generated order
13/01/94	part0001, 25	part0001, 25	MRP generated order
	part0003, 25	part0003, 25	ROP generated order
14/01/94			demand met: part0001, 25
15-16/01/94			no activity: weekend
17/01/94			no activity: no orders suggested by MRP or ROP system models

The sequence of the planning activities meant that following the generation of an order by the MRP model and based on the MPS, a subsequent order was generated by the ROP model to replenish the stocks allocated to the MRP-driven order. For example, on the 12/01/94 an order for 50 'part0002' was released in advance of the subsequent production of 'part0001'. The ROP planning which followed generated an order for the same quantity in order to replace the amount of 'part0004' used to manufacture 'part0002' and to maintain its reorder level. The final stock status being: 'part0001' (0), 'part0002' (0), 'part0003' (25) and 'part0004' (25).