AN APPROACH TO THE DEVELOPMENT OF EXPERT SYSTEMS WITHIN PRODUCTION PLANNING AND CONTROL

MARTIN ALISTAIR RODGER
Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM
September 1990

THE UNIVERSITY OF ASTON IN BIRMINGHAM

AN APPROACH TO THE DEVELOPMENT OF EXPERT SYSTEMS WITHIN PRODUCTION PLANNING AND CONTROL

MARTIN ALISTAIR RODGER
Doctor of Philosophy
1990

The thesis presents an account of an attempt to utilize expert systems within the domain of production planning and control. The use of expert systems was proposed due to the problematical nature of a particular function within British Steel Strip Products' Operations Department; the function of Order Allocation, allocating customer orders to a production week and site.

Approaches to tackling problems within production planning and control are reviewed, as are the general capabilities of expert systems. The conclusions drawn are that the domain of production planning and control contains both `soft' and `hard' problems, and that while expert systems appear to be a useful technology for this domain, this usefulness has by no means yet been demonstrated. Also, it is argued that the main stream methodology for developing expert systems is unsuited for the domain.

A problem-driven approach is developed and used to tackle the Order Allocation function. The resulting system, UAAMS, contained two expert components. One of these, the scheduling procedure was not fully implemented due to inadequate software.

The second expert component, the product routing procedure, was untroubled by such difficulties, though it was unusable on its own; thus a second system was developed. This system, MICRO-X10, duplicated the function of X10, a complex database query routine used daily by Order Allocation. A prototype version of MICRO-X10 proved too slow to be useful but allowed implementation and maintenance issues to be analysed.

In conclusion, the usefulness of the problem-driven approach to expert system development within production planning and control is demonstrated but restrictions imposed by current expert system software are highlighted in that the abilities of such software to cope with `hard' scheduling constructs and also the slow processing speeds of such software can restrict the current usefulness of expert systems within production planning and control

Expert System, Soft System Methodology, Production Planning and Control.

ACKNOWLEDGEMENTS

I would like to thank the personnel of British Steel Strip Products for their assistance during the project, in particular those who participated in the supervision of the project; John Roberts, Tony Edwards, Mike Stone, Ron Mercer, Mike Davis.

I would also like to thank the students and staff of the I.H.D. scheme, particularly Hardial Saggoo, Clive Wright, Mike Wood and Alistair Cochran, for their advice and guidance. Also thanks to my long suffering tutor, John Edwards, for keeping the project on an even keel.

Finally, a special thanks to Austin Bird, Dave Chard, John Fletcher, Glyn Hardwick, Peter Lee, John Peterson, Janet Flyn, Ken Jenkins, Chris Bennett, Tony Evans, Mark Davis and Brian Ralphs, the Allocators and Planners of South Block who put up with me for four long years. Without their helpfulness and patience the project would not have been possible. Indeed, it was their project.

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter briefly describes the background to the project and presents a rough chronology of the project work.

1.1 BACKGROUND TO THE PROJECT

This section describes the background to the project in terms of the organizations and people involved in the project and the reasons for their involvement. The problem situation then encountered during the project is described in chapter 2.

1.1.1 INTERDISCIPLINARY HIGHER DEGREES SCHEME

This thesis describes a project carried out within British Steel Strip Products Group through the Interdisciplinary Higher Degree (IHD) scheme at Aston University. Being an IHD project, it has certain features which separate it from normal postgraduate research studies.

The IHD scheme was founded in 1968 to accommodate closer co-operation between the university and industry. IHD projects may be based in any organization. However, two criteria must be satisfied. Firstly an IHD project must address a problem of genuine practical concern to the co-operating organization. Secondly it must provide sufficient scope and originality to form the basis for research.

The practice within the IHD scheme was for management of the project to include a supervisory team which would meet regularly to review the progress of the project. The team would be drawn from both the co-operating organization and the university.

1.1.2 THE BASIS FOR THE PROJECT

The co-operating organization in this project was the Strip Products Group of the British Steel Corporation. (By the completion of the project it had been renamed British Steel Strip Products as the British Steel Corporation had by then been privatized.)

The origins of the project lay in links between the Management Services department of the Strip Products Group and Aston University's IHD scheme. These links led the Management Services department to consider potential applications of expert systems within Strip Products Group with the possibility of this leading to an IHD project.

The Management Services department considered the function of Production Planning and Control (PP&C) as a possible focus for such a project. In particular, the Order Allocation department based at the Group head offices appeared to provide sufficient scope for an IHD project. Basically, the Order Allocation department loads orders onto the production facilities of the Strip Product Group. The Order Allocation task was considered to be problematical and a likely application area for expert systems. The eventual project description was as follows:-

> 1. The Strip Products Group owns a complex set of facilities including Blast Furnaces, BOS Converters, Concast machines, Hot Strip Mills, Pickle Lines, Cold Reduction Mills, Coating Lines and a number of Finishing Lines.
> 2. Every order taken from a customer, for the purpose of this project, causes use of a Hot Strip Mill. Thereafter it causes the use of a combination of one or more of the `down stream' facilities. It also causes use of `up stream' facilities but these will be outside the scope of this project. The combination of facilities appropriate to an order is referred to as the routing of that order.
> 3. At any given time because of the orders we have already taken each of these facilities has a load earmarked over a period of days/weeks.
> 4. The problem is to determine the routes appropriate to the next order coming forward for consideration, how soon the order might complete any one of the routes, and based on that how soon we could expect to deliver the required tonnages to the customer.
> 5. Much, but not all, of the data describing the load earmarked for our facilities resides in our computer files as does the data describing the orders on hand. It is envisaged that a computer system assisted solution may be feasible. Incorporating present expertise in an "expert

system" is also a possibility.
6. One aspect of the project will be to decide on the criteria to assess the success of the "solution".

Two points should be stressed with regard to this description of the project task. Firstly it was required that the problem should be framed or defined in terms of "criteria of success". Thus the project task could not be considered simply as the development of an expert system to computerize some of the "present expertise" within the Order Allocation department. Secondly, the use of expert systems was not obligatory. Thus tackling the allocation `problem' was the first priority. Nonetheless, expert systems were not to be dismissed out-of-hand. Indeed expert systems were seen as the significant aspect of the project in terms of the research.

1.1.3 THE SUPERVISORY TEAM

Members of the supervisory team from within British Steel consisted of the Group Management Services Manager, the Group Operations Manager and the Central PP&C Manager. The Central PP&C Manager was directly responsible for the work of the Order Allocation department and thus took the role of "client".

During the course of the project there was much change occurring within British Steel. Included in this were changes in the personnel and the organization of the central Operations Department. By the end of the project, the position of Group Operations Manager had disappeared. This led to the Operations Director becoming part of the supervisory team. Personnel changes affected both the other positions involved in the supervisory team during the course of the project. However previous members of the supervisory team remained in contact with the project work to a greater or lesser extent. These changes in personnel did not affect the outcome of the project.

The IHD scheme and the Operations and Information Management division of the Aston Business School provided members of the supervisory team from Aston University.

1.1.4 THE AUTHOR'S BACKGROUND

The project work was carried out by the author. Given the comparatively open nature of the project task, the background of the author may be seen as playing an important part in the framing of the project task.

The author's original discipline was Mechanical Engineering. This work gradually involved more and more computer programming eventually becoming exclusively so and extending beyond Mechanical Engineering applications. The author had also completed a course of study encompassing Checkland's Soft System Methodology (1) and had been using Soft Systems Methodology within his work for some twelve months prior to the start of this project.

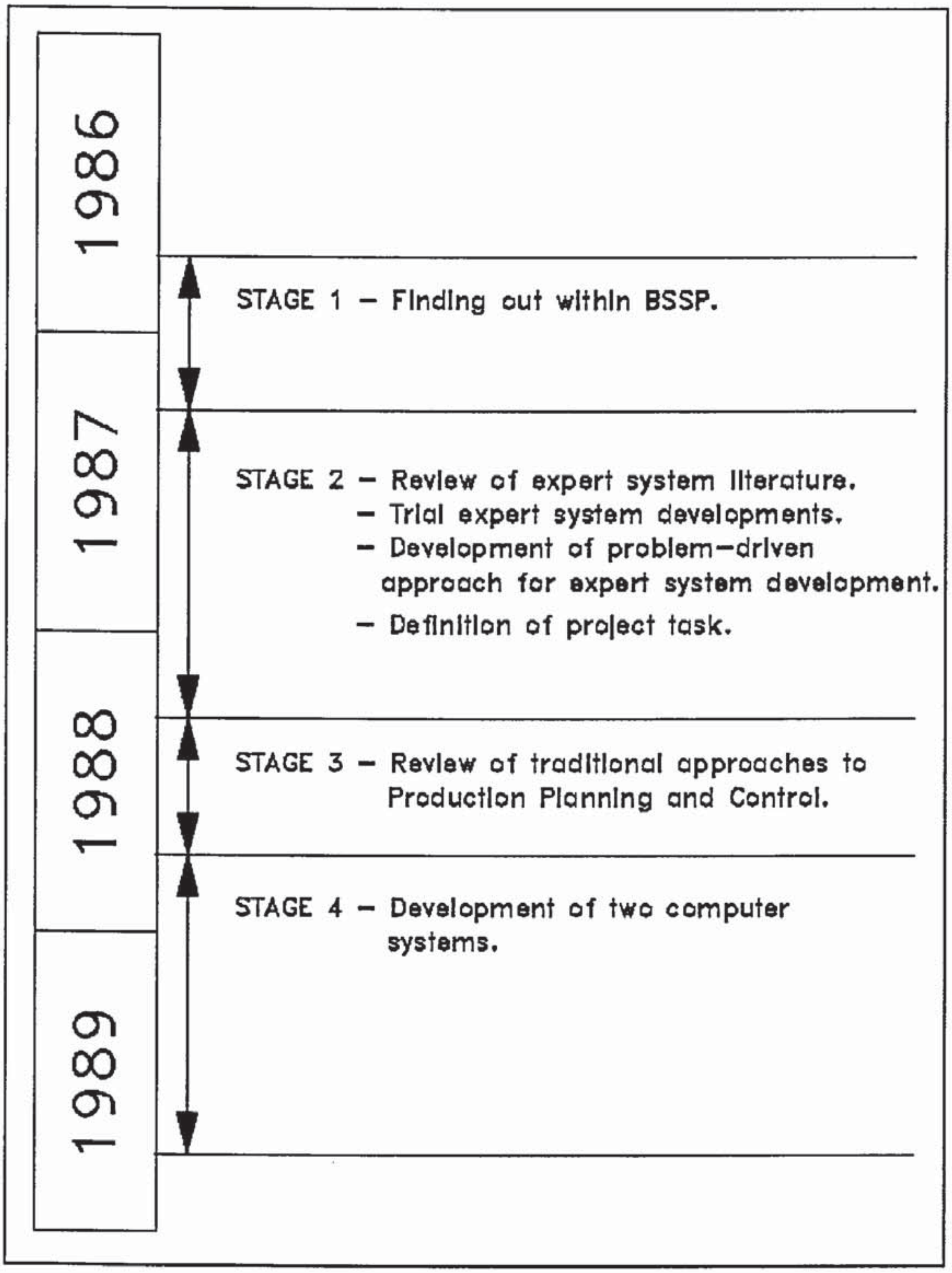
1.2 CHRONOLOGY OF THE PROJECT

The project was carried out between October 1986 and September 1989.

The work carried out during the thirty-six month duration of the project can be roughly divided into four stages as illustrated in figure 1.1. The first of these stages, covering some six months, was mainly taken up with an extensive finding-out exercise covering most aspects of the operations and sales functions of British Steel Strip Products.

The second stage covered some twelve months. Most of this period was involved with identifying the abilities of expert system technology. The main outcome from this work was a problem-driven

FIGURE 1.1 A ROUGH CHRONOLOGY OF THE PROJECT WORK.



approach developed specifically for the development of expert systems with the domain of PP&C. Also during this period the project task was defined.

The third stage covered some six months. This period was mainly taken up with work reviewing conventional approaches to PP&C.

The final stage covered the last twelve months of the project. During this time work was carried out on two computer systems. The two developments achieved varying levels of success, all contributing to the conclusions reached through the project work. Since the end of the project, limited amounts of work has been carried out with regard to one of these systems.

CHAPTER 2
THE PROBLEM SITUATION

The initial section of this chapter attempts to provide a structured account of the problem situation. In this, some detail has been left out for the sake of clarity. This is followed by a description of the work carried out to structure this `problem situation'. The final section discusses alternative approaches which were considered at this stage in the project as well as briefly discussing the scope of literature relevant to the project task.

2.1 THE PROBLEM SITUATION

This section presents an account of the problem situation. For the sake of clarity, it is not an exhaustive account. It attempts only to provide an appreciation of the situation coupled with enough detail where required to allow understanding of the work carried out during the project.

2.1.1 BRITISH STEEL AND BRITISH STEEL STRIP PRODUCTS

This section briefly describes the history and internal organization of British Steel (BS).

Developments within British Steel

British Steel plc is the fourth largest non-communist steelmaker in the world (2) and arguably the second largest after Nippon Steel as the other contenders are conglomerates of separate steel firms (3). It was formed in 1967 by the amalgamation of the largest steel companies within the UK to form the British Steel Corporation (BSC).

In 1980, BSC employed 125,000 people and was suffering from overmanning, overcapacity and a poor economic performance. In response, BSC undertook a severe programme of demanning and the closing of

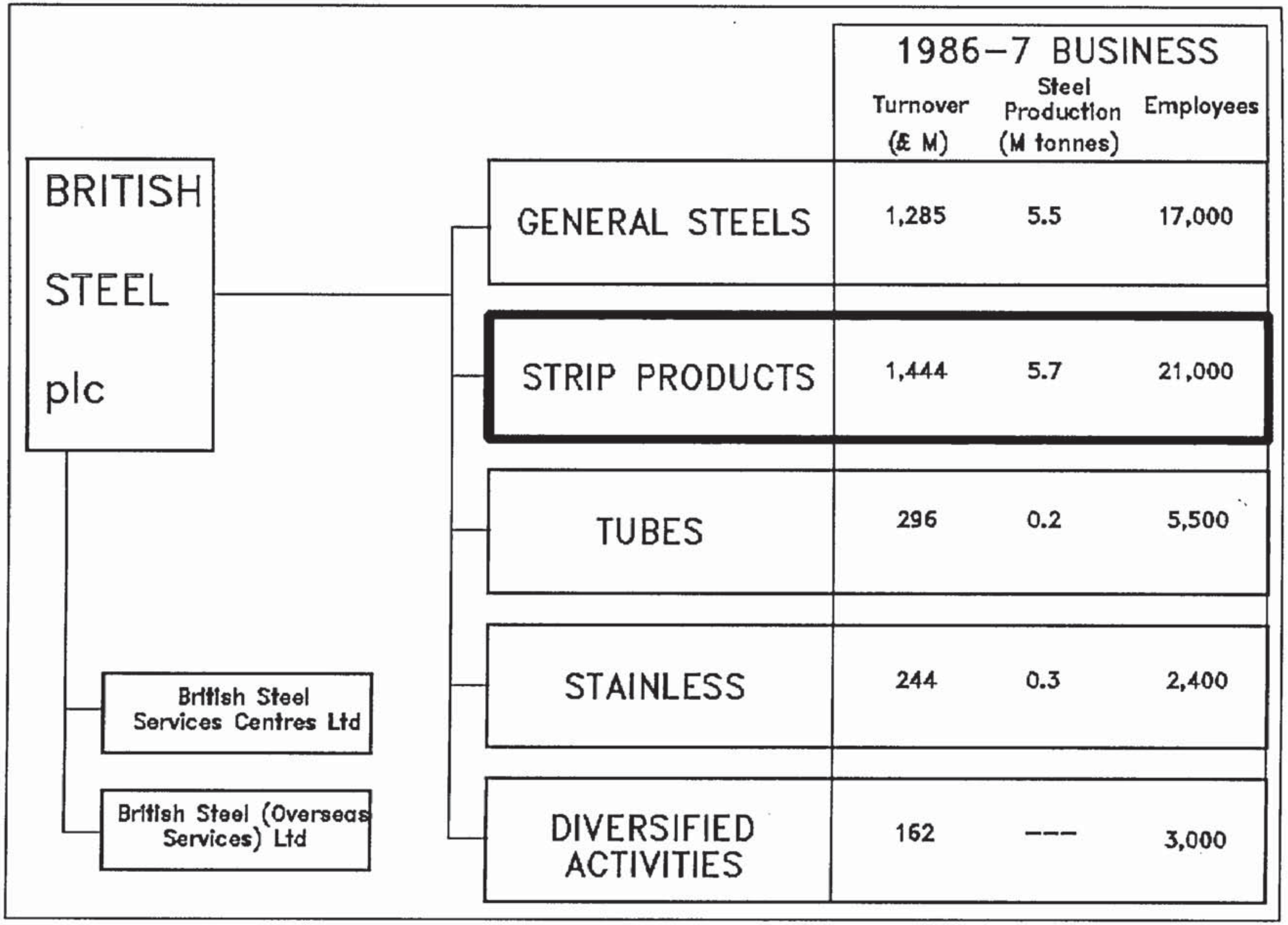


FIGURE 2.1 THE ORGANIZATION OF BRITISH STEEL'S BUSINESSES

uneconomic plant while also investing in new plant. As a result of this, by 1986, the work force had been reduced to 50,000 and BSC was making a profit. BSC's performance continued to improve allowing the corporation to be privatized in 1988 becoming British Steel plc (BS). The act of privatization made little difference to the internal operation of BS. Rather it marked only a milestone in the changes which had been sweeping the company since the severe reorganization of the early eighties.

The Organization of British Steel

BS is organized into five major product groups; General Steels, Strip Products, Tubes, Stainless and Diversified Activities. (Diversified Activities mainly manufacture railway products.) As illustrated in figure 2.1, the first two of these are by far and away the largest groups. These two groups between them operate all of the five integrated steel works within BS. Thus it is that British Steel Strip Products (BSSP), called the Strip Products Group prior to privatization, accounts for almost half of BS activity. It was with BSSP that this project was concerned.

In 1986 BSSP employed 21,000 people, produced 5.7 million tonnes of steel and had a turnover of almost one and a half billion pounds. In that year, BSSP's share of the steel strip market was approximately 60% within the UK and some 1.5% world wide.

2.1.2 AN OVERVIEW OF BSSP'S PRODUCTION AND THE ALLOCATION TASK

The production process of steel strip is first described followed by an account of developments within British Steel Strip Products (BSSP). Finally an overview of the allocation task is given. This section thus provides background information to allow the functioning of BSSP to be described in more detail in the following sections.

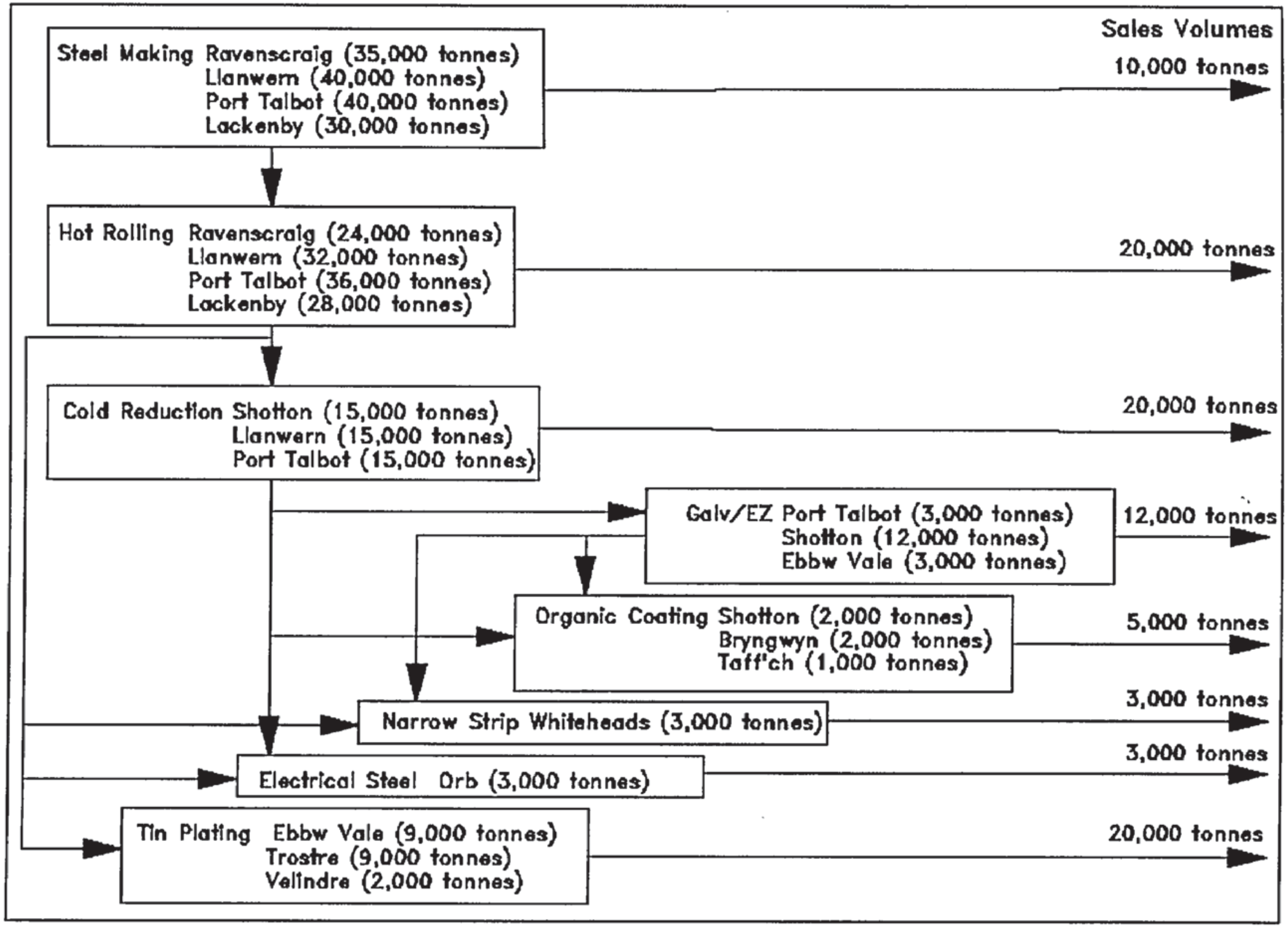


FIGURE 2.2 THE PRODUCTION FACILITIES USED BY BSSP IN 1986.

The Basic Production Processes of BSSP

BSSP naturally forms a single entity within British Steel. The production of the various types of steel strip comprise one very large interactive process, as illustrated in figure 2.2.

The first step in producing steel strip is the production of slabs weighing in the order of twenty tonnes each. Liquid steel is converted into slabs either directly by the continuous casting process or by first moulding billets from which slabs are rolled.

From slabs, hot rolled coil is produced in a hot mill. Hot rolled strip varies in thickness from 1.5mm to 12mm. After hot rolling, strip products can be sold in coil form or in cut lengths. Hot rolled strip is also sold having first been pickled.

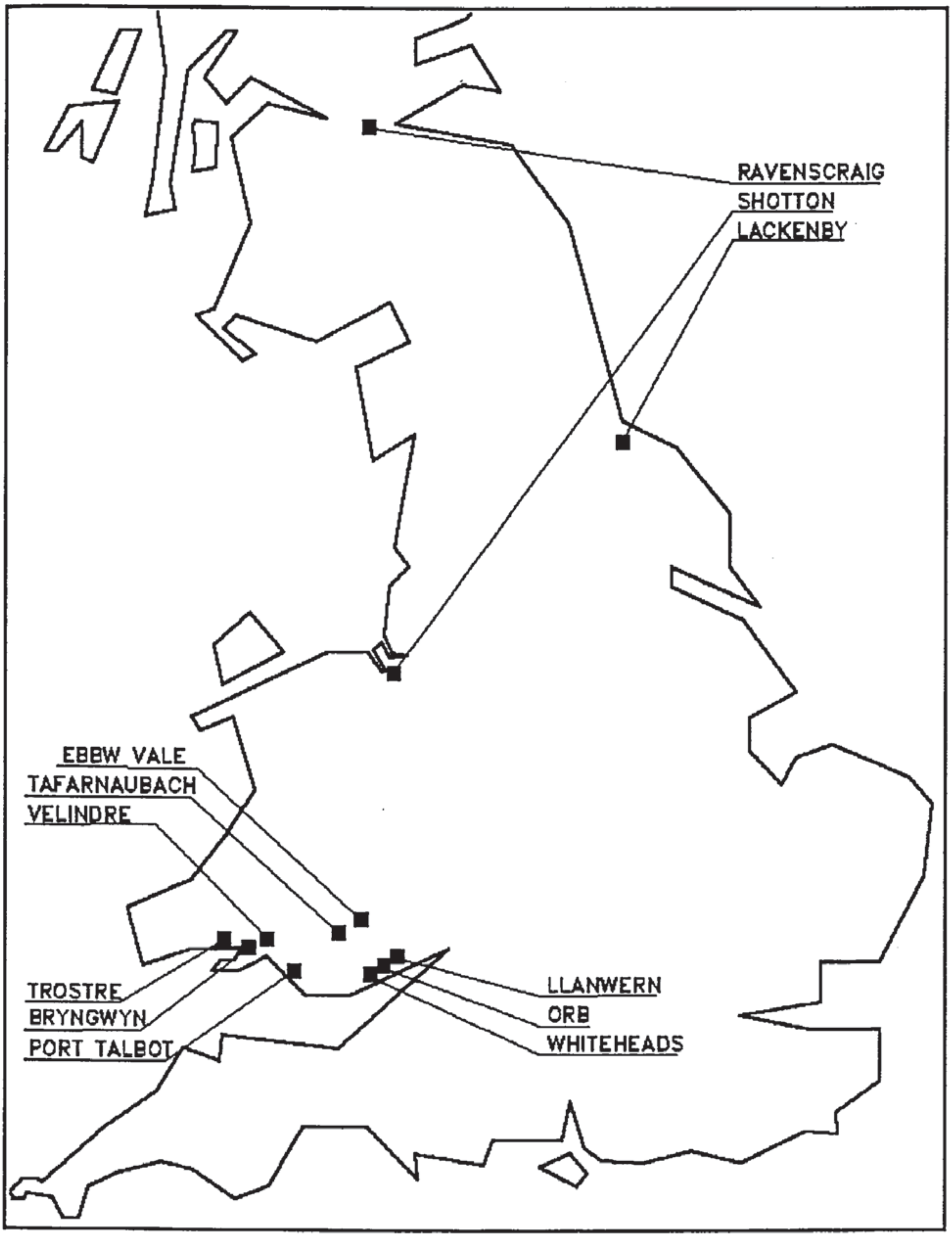
Pickling is normally the first process carried out in a cold mill where the coil is further reduced in thickness. Cold reduction reduces the strip to a thickness varying from 0.3mm to 3mm. Cold reduced steel can be sold in this form as "hard iron". WHen galvanizing steel "hard iron" is used. Usually though it is annealed to relieve work hardening caused by cold reduction, and then tempered to impart some hardness back into it.

Further processing in the main consists of coating the coil with some form of protective coating. One of two processes can be used to coat strip with Zinc, hot-dipped galvanizing or electro-plating. Certain alloys of zinc are also applied using these processes.

A number of other products are termed Organically Coated Steel (OCS). Here a plastic coating is painted or bonded onto the steel substrate which may have already been Zinc-coated.

Tin plating represents another form of coating. Two other facilities within BSSP carry out what could be called finishing

FIGURE 2.3 LOCATION OF PRODUCTION FACILITIES USED BY BRITISH STEEL STRIP PRODUCTS



operations. These are the production of narrow strip products and the production of electrical steel components.

This then describes the basic production processes of BSSP. As tinplate, narrow strip and electrical steel orders are not allocated centrally within BSSP, they are not of direct concern to the project.

The Production Facilities Used by BSSP

As well as showing the processes involved in steel strip production, figure 2.2 shows the rough capacities of each process and the twelve sites on which the capacity is situated.

The locations of the production facilities used by BSSP are shown in figure 2.3. Of the works run by BSSP, all but two of these are situated in South Wales, Ravenscraig being situated in Scotland and Shotton being located in North Wales. The twelfth steelworks mentioned, Lackenby, is run by BS General Steels and provided hot rolled products for the market for BSSP. It is situated on Teeside.

Developments within BSSP

The demanning within BSSP was severe. The workforce at Llanwern steel works for instance fell from 12,000 to 4,000. Such demanning was just as severe as that at Shotton and Ebbw Vale where steelmaking plant was closed down leaving only finishing and coating facilities. Sites had been closed completely; Gartcosh in 1986 and Velindre in 1989. In both these cases the decision was precipitated by changes in the market. Velindre for instance was closed because two major tinplate customers had been taken over by companies who themselves produced tinplate.

Major investment in new and improved plant during the eighties included new and modernized coating lines at Shotton, continuous annealing at Trostre, continuous casting, vacuum degassing and galvanizing lines at Llanwern.

In the main BSSP does not rely on external processing of steel. One exception is the use of Lackenby, a steel works run by BS General Steels, to provide Hot Rolled products for the market. Another exception is the subcontracting out of Galvanizing work to foreign steel companies. This `Off-shore Galve' is a temporary arrangement prior to the LLanwern Galvanizing line becoming operational.

Closures and investments were more than a modernization and rationalization of BSSP. They were also more than a reaction to the market share available to BSSP. They indicated trends in the strip products market as a whole for different products and quality of product (4). It should be stated that the facilities inherited by BSSP should not be seen as an imperfect combination of capacity and sites. The positioning of Ebbw Vale at the head of a Welsh valley may currently appear ridiculous. But as Warren indicated, in 1936 Ebbw Vale was seen as a "theoretically optimum location". Strip steel processing in Britain had "never been free from the trammels of the past" (5). There is no reason to suggest that this situation will change. Thus such incidents of closure as well as investment can be expected to recur.

<u>An Overview of the Allocation Task</u>

The function of concern within BSSP is that of Order Allocation. Within BSSP, most orders are made to order rather than supplied from stock. This is normal within strip steel production (6). The orders are received centrally and it is Order Allocation which controls their release to the works. That is Order Allocation determine which works will fulfil the order and when it will be delivered.

Orders released to a works often generates `interworks' orders.

For instance, Shotton works depends on the supply of hot rolled coil from other works. Thus an order for cold reduced coil released to Shotton will result in Shotton placing an interworks order for hot rolled coil on, say, Ravenscraig. Zinc-coated and OCS orders may involve a number of interworks orders.

The basis for all operations within BSSP is laid down in the Annual Operation Plan (AOP). The AOP is developed from sales forecasts and the production plans of individual sites. It determines quarterly levels of interworks orders and sales outputs. The completed AOP is costed and approved by the BSSP board. Allocators are not directly involved in the production of the AOP.

Allocators are however involved in the production of weekly plans. These are based on the AOP as well as the actual order load, the backlog of orders at the works, weekly production plans provided by the works and commercial priorities. It is on the basis of these weekly plans that the allocators released orders to the works.

As well as controlling the flow of orders out to the works, Order Allocation provides control over the flow of orders entering the orderbook. Sales personnel accept orders from customers, promising a delivery week. Allowable delivery weeks are provided by Order Allocation based on the queue of orders awaiting production.

2.1.3 COMPLICATIONS WITHIN THE BSSP PRODUCTION PROCESS

The description of strip steel production and the task of allocation given in the section above is greatly complicated by the realities of strip steel production. An account of some of the complicating factors is now given. The effect of these factors on the allocation task is illustrated anecdotally.

Sequencing Constraints

The rolling of steel strip requires the sequencing of orders through the mills. In hot and cold mills, such a sequence is known as a `round'. A round represents the production which can be carried out using a set of rolls. After this, production has to stop while the rolls are replaced. The sequence is determined by width and surface finish. A round is coffin-shaped; starting at a medium width to heat up the mill, the width can then be increased but then as the edges of the strip start to mark the rolls, the width is gradually reduced. Also as the round proceeds the surface finish of the strip produced deteriorates.

The sequencing of rounds is not a concern of the allocation function. This is done at the mill. However, Order Allocation has to allow rounds to be sequenced. Generally then allocators will be conscious of the mix of widths and finish qualities being loaded onto a works.

Sequencing constraints cause some types of orders to be grouped and processed only occasionally. An instance of this is what are termed "wide weeks". All orders for wide strip, roughly strip wider than 1200mm, are only processed during wide weeks which usually occur alternately at Port Talbot and Llanwern. It is the task of allocators to ensure that wide orders are not launched onto the mills during the wrong weeks.

Coating lines run continuously. Sequencing is affected by the type of coating, the weight of coating (or colour in the case of OCS), the surface finish, and the width and gauge of the substrate. The complexity of the process results in `campaigns' for certain types of orders in a manner similar to wide orders in the hot and cold mills but with the production runs less frequent. The low frequency and the larger number of upstream processes often means that Order Allocation progress the `campaign' through the upstream mills to ensure the substrate arrives on time.

Sequencing also affects steel making. The type of steel and also the dimensions of slabs affect the sequence of production. A steel grade is usually specified in terms of mechanical properties, usually tensile strength or ductile properties specified by some standard, for instance British Standard BS1449. However making a particular chemistry of steel will often involve the minimum production of hundreds of tonnes of steel.

An example of the complications caused by this were the guidelines imposed by Ravenscraig with regard to steel-make quantities. On one occasion an order for four tonnes of rather special steel was placed on Ravenscraig. The minimum steel make to cover this order was one hundred tonnes. Order Allocation was asked to justify the order. This justification required an input from the Commercial Department to decide whether the importance of the order deserved such treatment; it was a trial order for a Japanese car manufacturer then setting up production in the UK.

The reorganization within BSSP had gradually caused individual steel works to be more cost conscious. A year after the trial four tonne order, Ravenscraig were asking for a minimum three hundred tonne steel-make requirement. Thus Order Allocation was required to identify special steel orders and group them into batches. Later still, such orders were being accepted individually due to the slackening of the order book. Order Allocation however still checked with Ravenscraig unless the specified minimum steel make had been achieved.

So although not generally involved in the sequencing problems present at mill level, Order Allocation often monitors order loads to ensure the sequencing is not too restricting and sometimes become involved in more direct control of the flow of production through the launching of campaigns and the progressing of such campaigns through the various stages of production.

Quality Control and Capacity Measurement

The unit of steel-making is a "melt". The unit of strip steel production is a coil. A customer will order a quantity in tonnes and this will result in the steel being made and coils rolled. But one tonne of steel does not result in one tonne of hot rolled coil being produced. Figure 2.2 shows this: the capacities do not add up. This complication is coped with in the concept of "yield". One aspect of yield is due to the realities of the process; the topping and tailing of a coil after processing, the extra weight of coating, etc,. However, yield also accounts for losses due to quality problems.

When producing high quality steel, the quantity of steel made for the order may exceed the order requirement, thus allowing for the probability that some of the steel will fail quality control tests and not be usable for that order. The result is often that the customer is not provided with the quantity he originally ordered. He may have ordered fifty tonnes but receive on that order forty or sixty tonnes. Sometimes an order will not be satisfied with the quantity produced and part of the order will be remade from scratch.

Material failing quality control is not wasted. The mill will look for incoming orders for which the coil is suitable. If not, Order Allocation will look for orders yet to be released to the same end. Failing this, the coil may be offered to a customer at a reduced price.

This added complexity within the BSSP operation has a number of implications. Firstly, the capacity taken by allocated orders cannot be equated easily to the quantity of strip specified in the orders. Indeed, a significant percentage of orders is fulfilled from material 'downgraded' during quality control.

Secondly, the quantities delivered to customers can exceed the quantities required by those customers as specified in the orders. Such over-production has resulted in mills being overloaded with orders by Order Allocation. The onus is on Order Allocation to identify why a mill is overloaded. They after all are loading the orders onto the mills.

A third implication is the general acceptance of the presence of outstanding orders called "arrears". Arrears do have the benefit of easing the problems associated with sequencing a mill. Indeed sequencing constraints can result in some orders being processed early at the expense of causing arrears. Order Allocation take account of the levels of arrears when loading the mills.

A final consequence of yield is the existence of stocking arrangements for important customers to ensure that production difficulties do not affect such customers. Finished stock is held for some customers. Others have negotiated with individual works to maintain levels of 'in-process stock'; that is the quantity of work-in-progress as well as finished stock. The creation of stock orders of this sort is a Commercial task. Being normally high quality products, the level of such orders can be of great significance to Order Allocation.

Beyond the concept of yield is another complication. The unit of capacity normally used within BSSP is tonnes. In hot and cold mills, a `round' is scheduled not in tonnes but in strip length. In coating mills the area of strip is a better measurement of capacity. Thus an order book with a high proportion of heavy gauge orders may underload a mill while a high proportion of light gauge orders may overload a mill.

A further capacity unit is `packing units'. Allocators are often required to ensure that works are not overloaded with orders requiring `export packing'.

With the exception of OCS orders and `packing units', Order Allocation seldom needs to consider these alternative units of capacity. When such consideration is required, the infrequent use of such units of capacity is itself a problem.

The Individuality of BSSP Plant

The author uses the word `individuality' to cover ways in which the abilities of mills differed. Most straightforwardly, different mills cannot produce strip of the same physical dimensions. For instance Shotton's cold mill cannot produce wide coil. Whether a mill can produce a particular order is not an operational consideration. Rather it is a technical decision. Abilities in terms of such things as width are set out in the Product Manual, a one hundred and twenty page document published by the Technical Department. This identifies an order as being feasible, infeasible or whether it should be referred to the Technical department for a decision.

Although a works' ability to process an order is a Technical concern, Operations can become involved. Coil weights are an example of this. Allocators are seen as the source of coil weight information despite coil weight being a technical concern.

More obviously operational decisions are the decisions to allow a works to specialize. An instance of this is the use of Port Talbot hot mill for long runs while Ravenscraig fulfils the more specialist orders and Llanwern carries out an intermediate role. Another instance is the Ebbw Vale hot-dipped galvanizing line which specializes in heavy gauge orders. However even these operational decisions can be initiated by technical requirements and the decisions concerning on which site to invest in production equipment to meet such technical requirements.

Even with specialization of this sort, certain orders are more difficult to produce than others. This is something not reflected in the Product Manual. Such orders have occasioned works schedulers to moan that they were being provided with all the difficult orders while other works were getting all the easy orders, a comment which is evidently untrue when viewed from Order Allocation.

An example encompassing many of these factors of 'individuality' occurred in 1989. Until 1989, Ebbw Vale was the only works capable of producing IZ, a type of hot-dipped galvanized coating. The lightest IZ gauge available was 0.55mm although the Product Manual stated that lighter gauges may be available on referral to Ebbw Vale. Shotton began IZ coating in 1989, the new facilities being designed to produce IZ for gauges between 1.2mm and 0.5mm. The operational decision was made that Shotton would take over the production of lighter gauge IZ leaving the heavier gauge IZ for Ebbw Vale. A problem arose because two customers were ordering 0.45mm gauge IZ from Ebbw Vale. It was called 0.5mm gauge as this was the gauge of the coated strip, not the substrate gauge as is normal practice. Shotton were unable to produce these orders thus Ebbw Vale had to accept them. This meant Ebbw Vale had also to be provided with a large quantity of other light gauge orders to allow the mill to be sequenced. The low yield of such orders resulted in remake orders being placed. Ebbw Vale again had to provide these. Order Allocation had to liaise between these mills to overcome the immediate problem of completing the orders in hand as well as to push for a solution to the reasons for the problem; orders were being accepted which BSSP was not easily able to satisfy. The solution was eventually that such orders would not be accepted, a decision of which the Commercial Department had to be convinced.

A further point of individuality is that a particular order specification does not result in identical products from different mills.

Such differences between steelworks are partially due to minute variations in the rolling processes. Perhaps a more understandable example concerned OCS. A great deal of work has been done to achieve standard colours for OCS coatings. Yet due to variations in the baking process used to cure the coating, the colours from different works are still noticeably different. Indeed even to achieve this level of colour match, the paints used by the different works are intentionally supplied with variations in composition.

The difficulty matching the colours within OCS stands as a good analogy for the rolling processes. Just as the composition of the paint, the exact chemistry of steel is not standardized between works. The standards by which steel is specified, for instance BS1449, does not specify the exact chemistry of the steel. Thus variations can exist between equivalent grade from different steel mills.

This individuality between mills can appear during later processing of the steel by a customer or by downstream processes within BSSP. In this manner, these differences could become a concern of the allocators.

An example of this could be seen at Shotton where Hot Rolled coil is sourced from three different works. The quantities sourced from each of these three mills are laid down by the Annual Operations Plan (AOP). Ordering coil from the Hot Mills to fulfil Shotton's orderbook is normally controlled by schedulers at Shotton who work within the AOP. However, on occasion, Order Allocation has reason to control the source of Shotton's Hot Rolled coil more closely. In such circumstances, differences between coil from different sources can affect the options open to allocators. For instance, `edge cracking' problems at Shotton when rolling with Llanwern-sourced coil has on occasions prevented an increase in the supply of Llanwern coil to Shotton. Thus allocators can find a solution to a problem impractical due to this form of individuality.

Bottlenecks in Production

Figure 2.2 shows the flow of strip steel through the production processes of BSSP. The capacity of a steel mill is effectively fixed in the short term providing little leeway for coping with variations from plan. It is then not uncommon for bottlenecks to appear which are not readily solvable. The allocation task is greatly involved with the identification of and solution of such bottlenecks.

It must be stressed here that the level of detail shown in figure 2.2 greatly simplifies the process with which Order Allocation has to cope. More representative of the level of detail is figure 2.4 which shows the facilities within a cold mill. If a mill is loaded with too many orders requiring, for instance, inspection, the result will be not only late delivery, possibly to another BSSP mill, but also possible storage problems in the overloaded mill. A minor change in the order mix can be the cause of such a bottleneck. Analysing the order book to identify all possible bottlenecks of this kind would be a laborious task due to complications. Complications include, for instance,

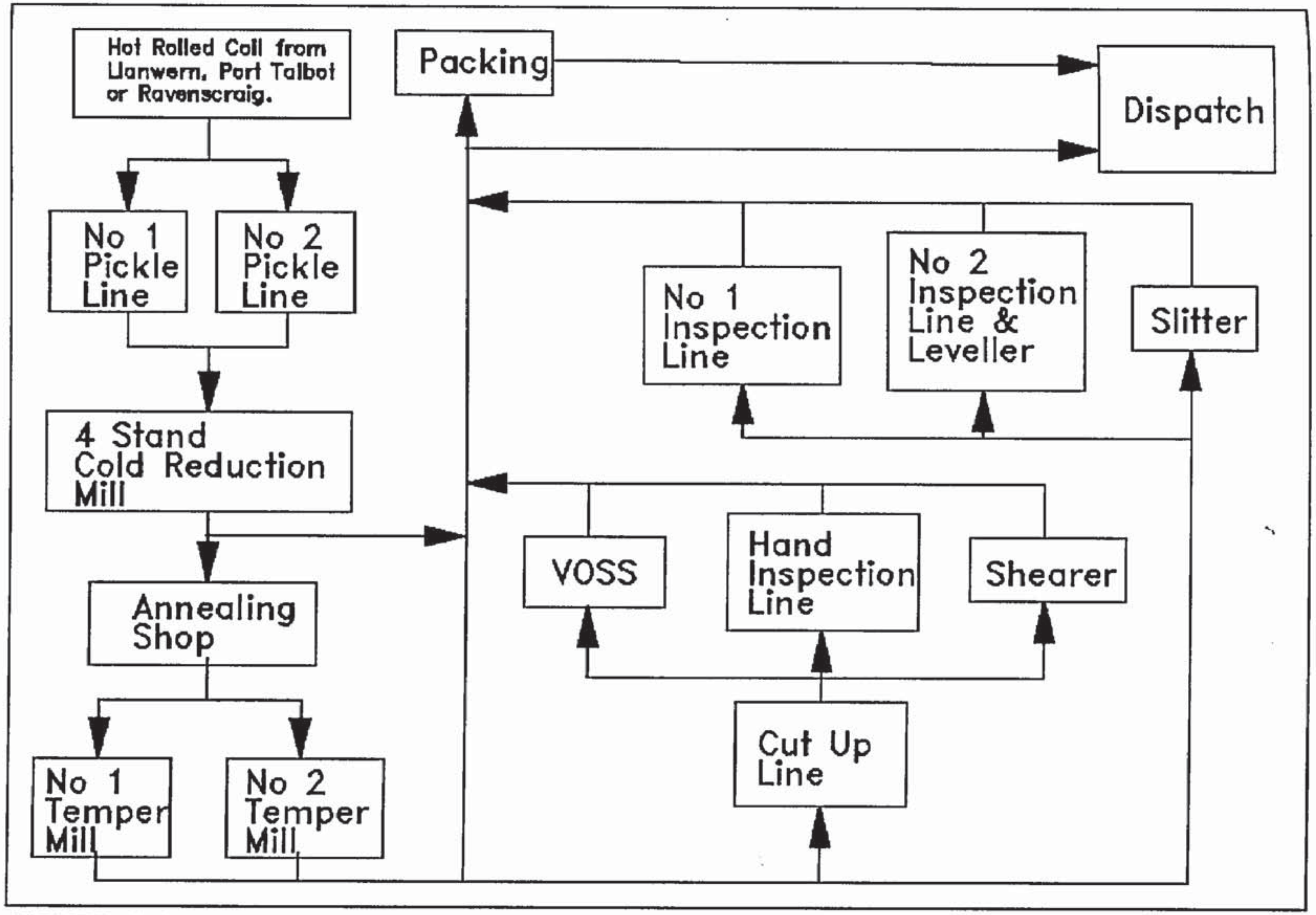


FIGURE 2.4 PRODUCTION FACILITIES WITHIN LLANWERN COLD REDUCTION MILL ILLUSTRATING THE COMPLEXITY WHICH CAN BE PRESENT WITHIN A SINGLE MILL.

difficulty identifying orders requiring inspection and converting ordered tonnes into packing units. Further, the capacity of the minor facilities can depend on manpower which can be common to more than one facility. Having been informed of a bottleneck, allocators will be involved in identifying the scope of the problem and ways of overcoming it.

A bottleneck may have been a problem over a period of time. An example can be seen in Port Talbot's cold mill inspection line. The order load on this facility has been monitored by allocators for more than three years. It appears that the `solution' is for Order Allocation to directly prevent the line from overloading.

Figure 2.4, in illustrating a cold mill shows the extreme case. However all mills include such `minor' facilities to some extent.

One major occurrence which further illustrates the problem of bottlenecks occurred in the years prior to 1988. At this stage Llanwern steelworks was unable to produce continuously cast steel, called concast. Concast is of more consistent quality than steel strip made from conventional ingot steel and this is very evident in the physical appearance of the steel strip. Customers prefer the concast steel. A trend was evident that customers were increasingly specifying concast steel and the possibility existed of the non-concast orderbook shrinking so much as to adversely affect the Llanwern order load. The rule was passed that any customer not having previously received concast would not be offered concast. Even then, orders from such customers were received from sales offices with concast specified in the miscellaneous instructions. The concast instruction was ignored by allocators if the order did not need to be concast and the customer provided with "concast equivalent" from Llanwern. Thus the Llanwern orderbook was sustained until Llanwern was able to produce concast in 1988.

Bottlenecks could have more obvious causes than changes in the order mix. Equipment breakdowns at mills are not unknown. An example of a major breakdown occurred in 1989. Problems with an electric motor in Port Talbot's hot mill lost some fifty thousand tonnes of production. In coping with profound losses of production, Order Allocation will be required to divert orders. In the case of the problem with Port Talbot's hot mill, the need was to prevent Port Talbot's cold mill from being starved of orders. In this, the individuality of the mills can be a major consideration.

Mills closed for holiday weeks but different mills close in different weeks. While an allocator would be forewarned of a standard closure, instances occurred where short notice is given of closures due to slack order books. In these instances, orders can be affected by a combination of closure and sequencing constraints. When such problems occur, Order Allocation need to check with the Commercial Department to ensure that the combined delay is acceptable to the customer.

Controlling Costs

Another consideration with a multi-site production process such as BSSP run and which should be mentioned is transport costs. Transport costs are high. An example of this is the benefit to be gained at Shotton where vessels can dock on site at certain times of the month. The cost of transporting steel strip to the nearest port would otherwise be something more than two percent of the sale price of the steel. Cost is of course part of the reason for the AOP. The AOP defines, for instance, quantities of steel to be shipped direct from Shotton's site.

So, while the allocation task is required to launch orders to be available for particular shipping dates, identifying such benefits is not an allocation function. The allocation task is not therefore concerned with cost saving per se. It is concerned however with loading orders on the works to allow the plans of others to be achieved.

The Allocation Task in the Context of Production Complexity

The overview of the allocation task given in section 2.1.2 described the release of orders to the works on the basis of weekly plans. By carrying out this task allocators involve themselves in a very complex production process. The process is both complex and difficult.

Being effectively responsible for the level and mix of work in the mills, Order Allocation needs to be aware of much of the detail of mill operations. Firstly, Order Allocation needs to understand the implications of its own actions on the mills. Secondly the task of allocation often results in the allocator becoming what could be called the interface between commercial and mill problem solving. It is hoped that these aspects of the allocation task have been adequately described and demonstrated within this section.

2.1.4 FUNCTIONS AND PROCESSES WITHIN BSSP

The organization of BSSP is first briefly described to introduce the functions of concern to the project. This is followed by accounts of the planning processes and the flow of orders within BSSP. Complications within the sales functions are then described and finally the allocation task is examined in yet more detail.

The Organization of BSSP

Figure 2.5 presents an organizational chart of BSSP. As can be

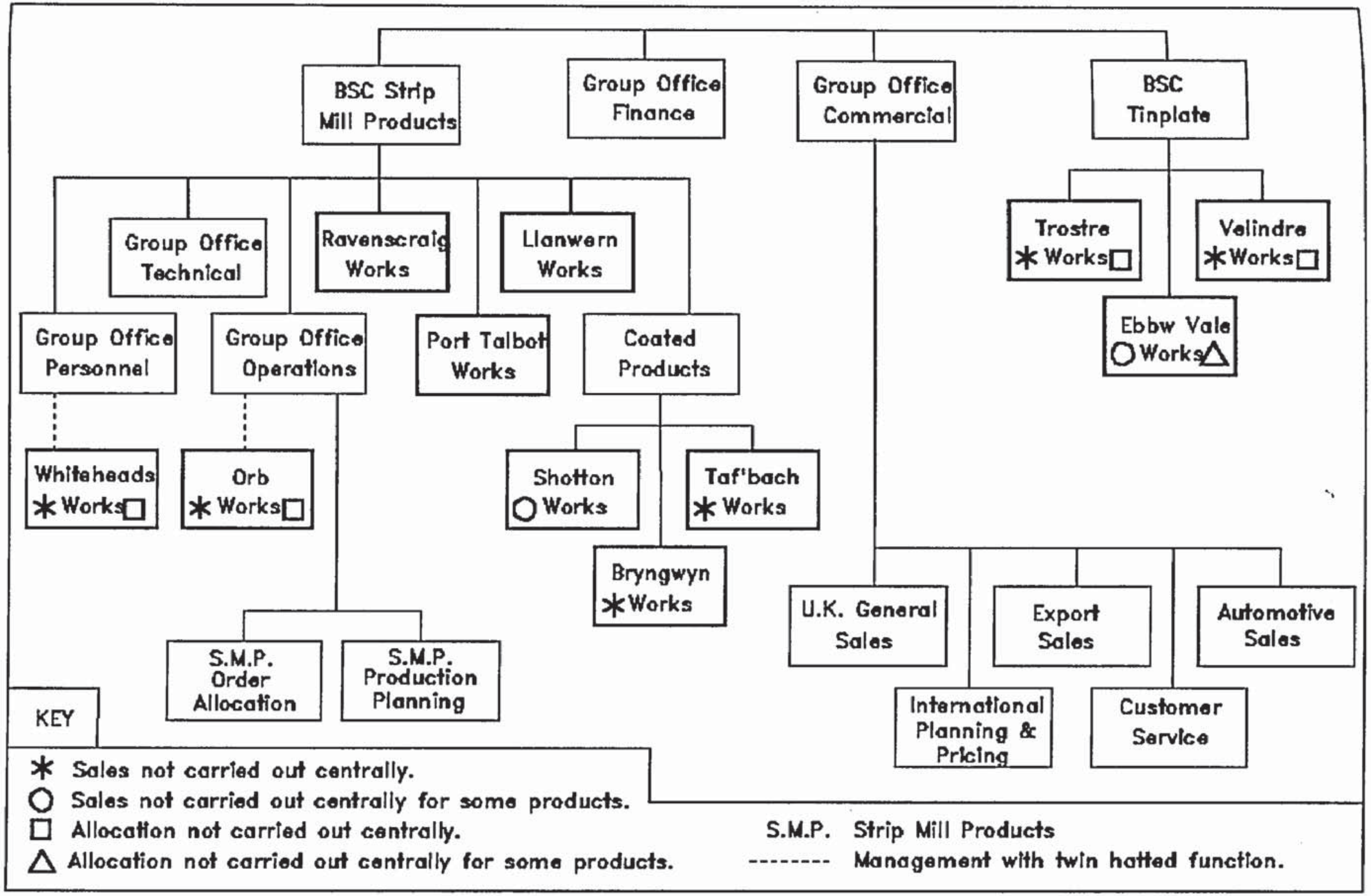


FIGURE 2.5 ORGANIZATIONAL CHART OF STRIP PRODUCTS GROUP
showing organization of central operations and sales functions. Chart correct in late 1986.

seen from this chart, the three works of BS Tinplate have no connection with the largest part of BSSP, Strip Mill Products, although they were supplied with substrate from them. Thus they are not a direct concern of the project. Similarly Orb and Whiteheads are of no direct concern. Further, the three works supplying Organically Coated Steel (OCS) while having their own sales function, have had their orders allocated centrally since a short time before the start of the project.

A complication to understanding the organization of BSSP is that Ebbw Vale, a Tinplate works, also produces Zinc-coated strip, orders for such products being obtained by central sales and allocated centrally. Likewise the majority of the products of Shotton works are not OCS and are sold centrally, unlike OCS.

Each works has its own Personnel, Operations, Technical and Finance functions. Indeed, there is little central control over the works. One area of control is the Annual Operation Plan (AOP) where the performance of the steel works is set out. Works are also accounted separately so their economic performance is also a matter of concern to them.

Figure 2.5 identifies three sales functions, UK General, Export General and Automotive. These deal with different areas of the market for hot rolled, cold reduced and Zinc-coated products. With the inclusion of OCS, this means that there are four sales functions providing orders which are allocated centrally.

Beyond these, two other Commercial functions should be mentioned. Firstly the planning function within International Planning and Pricing which identifies market size and projected market share. Secondly Customer Service exists on all sites to liaise between the customer and the site. Such liaison is not allowed before an order is allocated to that particular site.

Another point which should be made is that works themselves also carry out some sales functions. Customers are allowed to insist on the site which produces their order and many orders carry such a customer insist. Resulting from this, a particular site might arrange to carry stock for a particular customer or indeed promote itself to that customer.

In 1989, Order Allocation and the associated function Production Planning had a complement of ten, including five allocators and two planners. The function is located at the Group offices, on the same site as Llanwern steelworks.

The five allocators specialize by product; hot rolled, cold reduced, Zinc-coated and Organically Coated strip with the fifth allocator matching orders to downgraded stock. Also, each allocator is responsible for obtaining weekly plans from particular works and for generally liaising with those works.

The planning function is involved in creating the Annual Operation Plan (AOP) and the weekly plan (known as the Four Week Plan), as well as monitoring the achievement of the AOP.

Planning within BSSP

The longest term plan of concern to the allocation function is the Annual Operation Plan (AOP). This sets out quarterly outputs for the financial year. It can be converted to average weekly outputs as the number of weeks worked by a mill in each quarter is also specified. The AOP is revised two or three times during the year.

The starting point for the AOP is a sales plan. This is compiled by the planning function of International Pricing and Planning with inputs from the sales functions of BSSP including tinplate, electrical steel and narrow strip. This then represented a market requirement for all products.

The market requirements are converted into an operational plan by SMP Production Planners. This defines the quarterly interworks order levels and mill stock levels. The plan is then sent to the sites where more detailed plans are defined to meet the required output. These plans are then costed on site and returned to the centre for board approval.

The allocators are not involved in the development of the AOP. However the AOP has a large impact on the Four Week Plan (FWP) with which the allocators are involved. Also the AOP remains the only basis for planning beyond the horizon of the FWP.

The FWP is produced weekly for the next four working weeks. Its starting point is a plan produced by each site giving their expected production in those weeks. The production levels can be affected by stock levels on site and recently achieved production levels as well as the production levels set out in the AOP. The sites also provide data as to the level of arrears and the level of orders already loaded onto them. The allocators use the plans from site and add to this the unallocated order book. The SMP Production Planners take these plans to produce a first draft of the FWP.

The operational problems being tackled by the FWP often impose major constraints on the sales functions. Thus the draft FWP is negotiated within a weekly allocation meeting. Present at the meeting are representatives from the four sales functions and central customer service as well as the allocators involved. The meeting is chaired by the Allocation Manager.

The allocation meeting allows direct commercial input into the

FWP. However, it is little concerned with the levels of new orders entering the orderbook; more it is concerned with the allocation of the current orderbook. Explanation of operational constraints is sometimes difficult to follow and the focus of discussion is not always relevant to the problems at hand. This the author sees as understandable given the task which the allocation meeting attempts to carry out.

The FWP effectively set the capacity for each product from each site available for each sales function. Works capacity can be altered by the meeting but not to any great extent. The finalized FWP then defines levels of orders to be released to the works. This may then need further Commercial Department involvement to identify priority orders requiring release.

The achievement of the FWP is not considered centrally within BSSP other than its effect on the order load on the following week. The works' performance is monitored centrally week by week in terms of achievement of the AOP, a task which the SMP Planners are responsible for. The level of production monitored is not in terms of orders fulfilled or delivered tonnes. It is in terms of processed tonnes and, as explained in section 2.1.3, this does not directly relate to the quantities of orders fulfilled, the quantity used in the FWP.

Allocators monitor order release against the quarterly outputs laid out in the AOP. The works are in this respect not concerned with orders released to them, but orders delivered by them. Thus to achieve works quarterly targets, allocators may also be required not to release too many orders which would result in deliveries the following quarter. Such orders affect both quarterly output and end-of-quarter stocks. This work is not assisted by formal planning.

The Flow of Orders through BSSP

The work that is involved in obtaining and processing orders can be demonstrated by considering the flow of a UK General order through BSSP.

Generally order enquiries are received by a Regional Sales office. The customer will specify the required delivery week and in response the sales staff will `promise' a delivery week. The `promised' delivery week is determined from a set of rules, the leadtime rules displayed on the central computer. These rules specify the earliest Hot Roll week and delivery week for all basic categories of steel with further rules identifying whether extra weeks are needed. Some categories or types of order will be labelled `refer'. This indicates that the `promised' Hot Roll and delivery weeks can be obtained by contacting Order Allocation. The order will then be transmitted on paper to the Group offices at Llanwern where the order will be entered onto the central computer. On the central computer, orders reside in one of three databases depending on whether the order is a Home, Export or OCS order.

Any order which is new business is referred to the various works to identify which works can technically fulfil the order. This check occurs despite the steel having been ordered to a specification. The reason is that, despite the specification, the steel is still sold as `fit for purpose'. This can be a time consuming activity; the order cannot be released until a works has accepted it.

Order Allocation relies on a number of analyses of the orderbook databases. Listings of the unallocated orderbook and recently allocated orders are produced daily. Also run daily are programmes such as X-10, a programme which lists all unallocated UK General orders with incorrect leadtimes. (That is orders with incompatible combinations of `promised' Hot Rolling week and `promised' delivery week, while taking account of work's holidays and wide weeks.)

An order having thus appeared on the orderbook will normally be allocated a week or so before the `promised' Hot Roll week. The allocator will schedule the order by defining the `scheduled' Hot Roll and delivery weeks and releasing the order to a works. If the `scheduled' weeks are later than the `promised' weeks, the order would contribute towards the level of "unallocated" arrears, the levels of which are monitored by the SMP Planners.

The order is released to the works which will deliver the finished product to the customer. Once released to this works, the order will be routed; that is the processing required by the order will be determined. An attempt would then be made to match the order to existing downgraded stock. Failing this, works orders will be issued.

The order may involve supply of coil from a different site. For instance, Shotton works has to be supplied with hot rolled coil. To obtain this coil, Shotton would enter a second order on the UK General database specifying the supplying works and required delivery. Such an interworks order then does not require allocating centrally. Such interworks orders would be processed by the upstream works as though the downstream works were a normal customer.

The order will thus be processed by one or more works and delivered to the customer. Delivered quantities would be recorded on the central computer. Should the delivery date to the customer be later than the `scheduled' delivery week the delivery would contribute to "allocated" arrears. As with "unallocated" arrears, levels of "allocated" arrears are monitored by the SMP Planners.

The above account is still somewhat simplified. One area which has yet to be mentioned could be described as `commercial complications'.

Commercial Complications

The most prominent commercial `complication' is fluctuating demand. Historically, fluctuating demand has always affected the steel industry (7). It affected the various strip products very noticeably over the length of the project. An example of this was the disappearance of imported German Galvanized strip from the UK market in 1988 due to increased demand from German car manufacturers. The German demand was soon mirrored by UK manufacturers resulting in the use of overseas galvanizing capacity by British Steel. By 1990 however, the orderbook had weakened to the point where Galvanizing lines were being closed for extra holidays as a way of reducing capacity.

Fluctuating demand is exacerbated by the demands of steel stockholders and also by the use of leadtime rules by BSSP as a mechanism to prevent overbooking. Such use of rules has been criticized as a means of order entry control (8). However beyond the workings of the leadtime rules are other problems. Some customers bypassed the leadtime rules. In times of overload, orders are placed by some customers in excess of their actual needs, knowing that they can defer deliveries to a later date or suspend deliveries indefinitely or just cancel orders. Some customers are able simply to insist on a delivery inside the quoted leadtime; prior to privatization, there had been instances where a customer had used political power to expedite his orders.

The mechanism to deter such manipulation of the leadtime system consists of not allocating orders for any customer with excessive levels of suspended orders. Implementing this mechanism, a mechanism devised by the Commercial Department, is the job of the allocators.

Prior to the start of the project, the historical mechanism to control the orderbook consisted of turning down export business during an order overload. This mechanism clashed with a policy of expanding export business and was thus not used from 1986.

As well as the standard Make-to-Order business, other business is carried out by British Steel. Speedstock and Quickstock are names of schemes whereby ranges of standard hot rolled and Galvanized strip respectively are provided on shortened leadtimes. Similarly, Alpha Stock provides the same service as was provided by Alpha Steel before it was absorbed by British Steel. Prior to Galvanizing coming on stream at Llanwern, British Steel sells `Off-Shore Galve'. Although ostensibly the same product as standard British Steel produced `Galve', it was treated as a separated product commercially. Order Allocation had to push for action when the orders for standard Galve began drying up while orders for `Off-Shore Galve' remained strong. The situation was operationally nonsensical.

The Allocation Task

Much of the allocator's information is obtained through analysis of the three databases containing the orderbook. Most of this work consists of programmes written in C.A.Earl, a database query language. Maintenance of these programmes is carried out by the allocators and includes utilities such as order specification decodes. Such decodes are used by other functions within BSSP.

This maintenance became an Order Allocation duty unofficially. BSSP's central computer services historically were unable to provide the maintenance causing allocators to take on the work. Only very simple analyses, those which require no maintenance, remain under central computer services control.

Some analysis proved too difficult to maintain. X-10, a programme which identifies unallocated orders with incorrect leadtimes is considered a complicated programme by the allocators. Only two people in the allocation section ever carry out anything more than the most simple maintenance on it. Yet X-10 does not identify orders entered inside the leadtime. A programme to carry out this analysis proved too difficult to maintain and was dropped.

As well as programmes run on a daily basis, often programmes are written in response to a particular enquiry or problem. A typical example of this was an analysis of the orders placed by Ebbw Vale on upstream works to identify the volume of orders which would be requiring capacity on Ebbw Vale's pickle line. Such programmes may be developed on behalf of Production Planning and Control (PP&C) staff at the works or in response to some particular problem facing an allocator.

All plans produced by the allocators are compiled and documented using spreadsheets on micro-computers. Data obtained from the central computer is entered manually into these plans.

Allocators spent a significant proportion of their time answering enquiries which often concern only single orders. Such enquiries can originate from site or from a sales office. Sometimes answers can be found by accessing, for instance, the particular order record or the customer file on the central computer. Other enquiries are passed on, the allocator on such occasions sometimes acting as an advocate for the enquirer. Other enquiries are of a more general nature concerning operational information; leadtimes, works holidays, coil weights, etc.

Orders are actually allocated using a utility on the central mainframe computer. This is a rather primitive utility, the order number, the scheduled delivery week and the works involved all having to be entered. Releasing an order is not an infallible process. Thus it is necessary to check the orders released against listings of orders newly allocated to works on the day following release.

Orders can be amended by customers. However amendments can be overruled by Operations, Technical and other functions within BSSP. Amendments awaiting clearance are processed using a utility on the central computer. Allocators are expected to process amendments quickly.

2.1.5 DEVELOPMENTS IN COMPUTER SYSTEMS WITHIN BRITISH STEEL

This section describes a number of computer system developments either directly relevant to the allocation task or relevant to the project task.

A Previous Attempt to Aid to Allocation Task

In the mid-seventies an attempt was made to assist the allocation task directly. A Linear Programme (LP) was built to allow the best mix of orders between the different works to be determined.

The programme categorized an order using ninety-nine categories. The categories mainly defined the ordered product. For instance category 10 was defined as Hot Rolled Pickled coil with gauge greater than 6.35mm. The categories were not mutually exclusive. Category 10, for instance, would also appear as category 3, Hot Rolled coil, as well as category 1, Hot Rolled Pickled coil. Typically an individual order would be represented in some seven categories.

Orders in the orderbook databases awaiting release were identified by their `promised' Hot Rolling week and then categorized by the programme.

Allocators were required to provide a table of maximum capacities for each category and for each works producing that category. The values in such tables were not fixed week to week. Amending the table was a tedious task and was the major cause of the system not being used as intended. This table was the only facility allowing allocators to control the order mix between the different works. Within the constraints of maximum capacities, the LP would consider any order mix as allowable.

The programme was run overnight. This further constrained its use as overnight running prevented the allocation policy set out in the maximum capacity tables from being tested easily.

When run, the programme identified the relevant orders in the orderbook databases, divided the orders into released and unallocated and then categorized them. The LP then used the cost of releasing an order of a particular category to a particular works to minimize costs while constrained by previous releases and capacities in the capacity table. The costs only consisted of transport costs although the intention was to add other costs.

The programme provided suggested allocations for each order although the programme could not differentiate between orders with the same categorization. Listings or totals of the output from the LP could be printed by works or by categorization. The suggested allocation could also be used to generate release messages as an alternative means of releasing orders. These could be accessed as a list of order numbers, scheduled Hot Rolling weeks and works codes. Allocators could amend the works code prior to submitting the suggested allocation for actual release.

The programme proved unsuited to the task of allocation in its intended role. Within a month of its first use by Order Allocation the allocators had stopped using it due to the tedious task of filling in the maximum capacity tables and the restrictions imposed by the programme only being run overnight. The system was however used for a number of years to test the actual allocation and to provide a basic order load analysis. That is, the system was used to monitor the levels of orders released onto the works and to identify trends in the orderbook generally. In this role the allocators did not need to fill in the capacity tables. Such a use was a far cry from its intended purpose. Further it was at this time the only means of analysing the orderbook.

<u>Developments in Computer Systems Relevant to Allocation</u>

During the course of the project, new computer facilities became available. By the completion of the project some use was being made of these new facilities. However, none of them had by then entered mainstream use by the allocators.

One of these new computer facilities is called SURE 2 (SUpplier REsponsiveness system). This system was designed to allow customers to identify the progress of their orders through the works. It also provides allocators with the opportunity to identify the actual loads on the mills in terms of customers' orders.

Another new facility was the availability of the data analysis language, SAS. However having being available for more than a year SAS has yet to supersede C.A.Earl for any of the daily allocation programmes.

The allocators at the start of the project had between them five terminals for the mainframe and two micro-computers. By the end of the project, the decision had been made to provide each allocator with a micro-computer linked to the mainframe. Thus direct input of data from the mainframe into spreadsheets or other micro-based software will become viable.

Finally, most functions used by the allocators on the mainframe were part of what is called the Order Entry System. By the end of the project, there was a real possibility that this ageing system would be re-written providing new functionality. Such a change would greatly alter the allocation task. For instance the system, if implemented, will allow allocators access to the on-site routing information.

Computer System Developments within BSSP's Production Planning and Control Functions

The development of PP&C systems at the different steel works within BSSP is not controlled by any central function. Indeed it was policy that local solutions should be found to local problems (9).

PP&C systems which were being developed during the course of the project include a scheduling system for the Ravenscraig concast steelmaking process, simulation models of the docks and smelting beds at Port Talbot, a scheduling system for all processes at Shotton works and a production information system at Llanwern.

The development of these systems was carried out by different organizations. The first two, at Ravenscraig and at Port Talbot, were carried out by effectively Operational Research (OR) sections within BS and coincidentally both used Genetik, a simulation package. OR had historically carried out much work within BS. However, since the severe demanning of the early eighties, OR sections have not explicitly existed within BSSP although this is not so elsewhere in BS. The Ravenscraig development was carried out by OR staff seconded from elsewhere in BS while the Port Talbot developments were carried out by the Industrial Engineering department within BSSP Management Services.

The third system, at Shotton, may not be adopted as it is an adventurous scheme. It was originally proposed and was being developed by an external software supplier. It is based generally on Material

Requirements Planning methods. The fourth system, at Llanwern, was developed by Llanwern's own computer services department.

Expert System Developments Within British Steel

Much of the justification behind the project was the promise of expert systems as a problem solving technology. Within Port Talbot, where the project was originally conceived, expert systems were being developed within the area of process control and one of these systems had undergone extensive live trials (10).

Other expert system developments within BSSP included developments within the area of fault diagnosis at Ravenscraig (11) and the development of a system to provide technical expertise to sales staff based on the Product Manual.

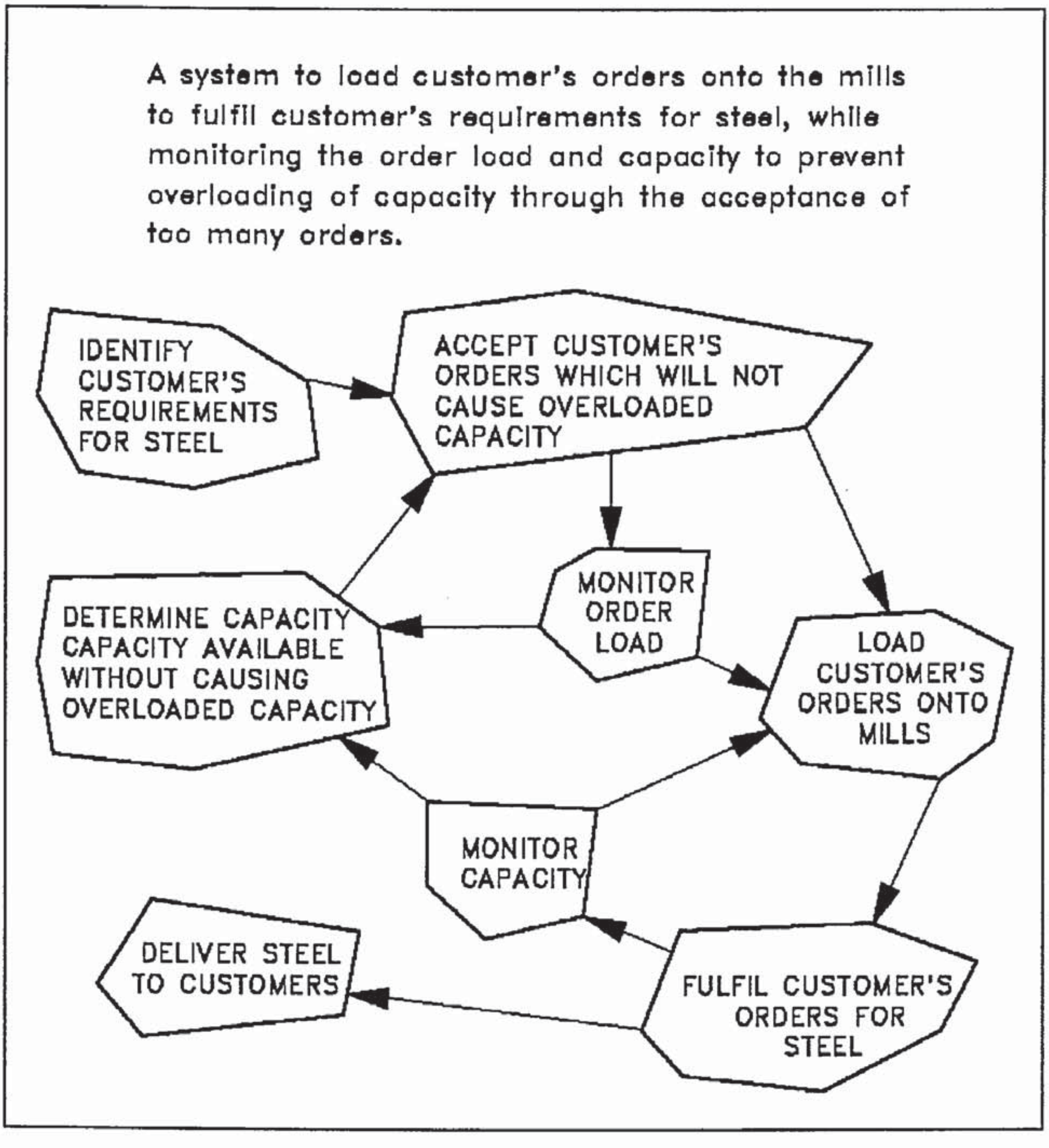
Within BS General Steels, a micro-based expert system to assist in the control a steel smelter had become truly operational while a second system to advise on the holding of slow moving spares for a steel works was undergoing trials. A third system had been used to capture the expertise of operators of a mothballed steel mill. By the end of the project the mill was still mothballed and thus the system has remained unused.

2.2 STRUCTURING THE PROBLEM

This section describes the finding out process carried out at the beginning of the project and the method by which the problem was structured leading to a definition of the task which the project would carry out.

Structuring the problem included the development of a measure of performance for the project; the level of unallocated arrears. From this point, attempts were made to identify the most significant causes

FIGURE 2.6 CONCEPTUAL MODEL USED TO GENERATE THE ISSUES SURROUNDING THE ALLOCATION TASK.



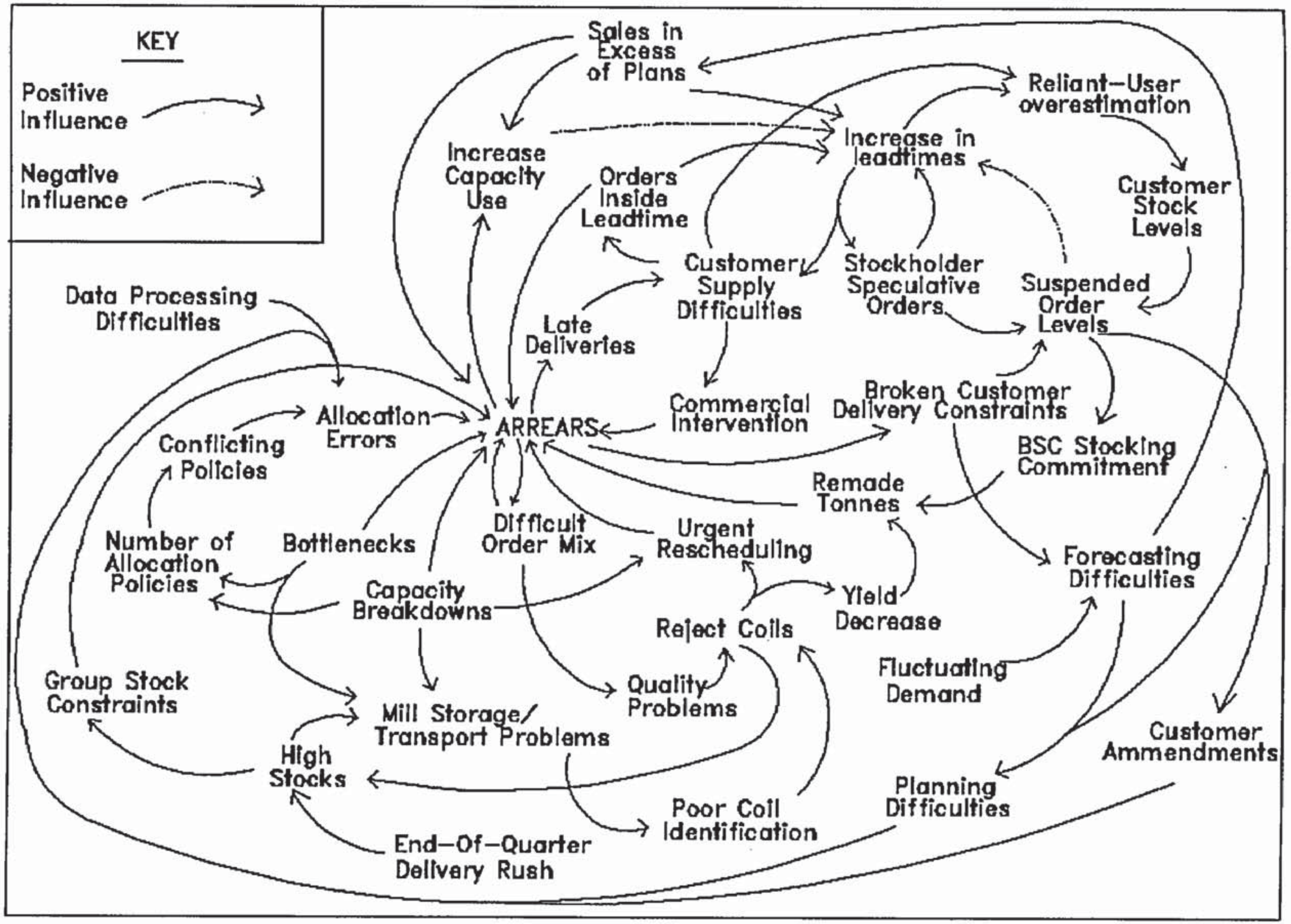


FIGURE 2.7 CAUSAL LOOP DIAGRAM OF THE ISSUES GENERATED FROM THE CONCEPTUAL MODEL IN FIGURE 2.6.

of unallocated arrears. As this proved difficult, identifying the significant causes became itself a major component of the project task.

The Finding Out Process

Most of the first six months of the project was taken up in an extensive finding-out exercise covering many functions within BSSP. Much of this period was taken in examination of the allocation task. Allocators were willing to describe their work but they often had great difficulty escaping from the description of individual cases. They appeared to find difficulty presenting an overview of their work. Also, understanding the allocation task was complicated by the use of much jargon within the task. Jargon included technical, commercial, computer services and operational terms often loosely applied or abbreviated.

The finding out included visits to two works, Llanwern and Shotton. The main reason for not including more works visits was the differences between the working practices evident from these two visits. Identifying further differences in working practice was not believed to be required at that stage given the open ended nature of the project remit.

Other functions examined included central sales functions, sales offices and accounts.

Determining a Measure of Performance for the Project

By the end of the initial finding out period, a great deal of detailed information had been collected. Many difficulties within the organization had been identified. Yet no central theme was apparent. The problem situation appeared to be best described as a "system of problems". Certainly no consensus of "the problem" appeared to exist except that the allocation function often encountered problems which it overcame only with difficulty. It was apparent to the author that what were perceived as `problems' and the `difficulties' involved in overcoming them varied from role to role within the problem situation.

Articulating this point could well have led to a consensus as to what the project should tackle. However, the author took a different course of action. The remit under which this project work was carried out had specified that:-

> One aspect of the project will be to decide on criteria to assess the success of the "solution".

In other words, a measure of performance for the project was required. The author saw such a measure of performance as providing a statement of the project task. He thus persevered to determine such a measure of performance without direct recourse to the stakeholders within the situation and through such work to achieve a consensus amongst them.

The technique used to determine a measure of performance for the project had been used previously by the author elsewhere (12). The technique consisted of using the constructs of Soft Systems Methodology (SSM) to provide a logical set of activities representing the problem situation in the form of a conceptual model. Comparing the activities with the real world situation allowed the significant problems or issues involved with those activities to be listed. The interactions between those issues could then be analysed by developing a causal loop diagram thus allowing the most significant issues to be identified.

The conceptual model and causal-loop diagram developed are shown in figures 2.6 and 2.7 respectively. The central issue within the causal loop diagram was quite evident and immediately provided a measure of performance for the project. That was "untimely steel" or in the jargon of BSSP, arrears.

As mentioned in section 2.1.4 above, arrears are broken into two types within BSSP, the levels of each being monitored by the SMP

Planners. "Unallocated" arrears occur when an order is scheduled for production by an allocator in a week later than that `promised' to the customer. "Allocated" arrears occur when an order is delivered to the customer later than the week scheduled by the allocator.

Two considerations pointed to unallocated arrears as a more relevant and useful measure of performance. Firstly the total levels of arrears pointed to both types being significant to the operation of BSSP. Calculation of the total levels of arrears was a straightforward process. For the calendar year 1986, the level of allocated arrears was some ten percent greater than the unallocated arrears giving no reason to prefer one above the other. Indeed, since that date, the level of unallocated arrears has always been of the same order and often exceeded the level of allocated arrears.

The second consideration was the causes of the arrears. Allocated arrears are ostensibly caused by difficulties on site. Throughout the finding-out process there was never any view expressed that allocated arrears were significantly caused by poor allocation of orders. Unallocated arrears, on the other hand, were seen as an inability to match the order load to the available capacity, something directly relevant to the allocation task.

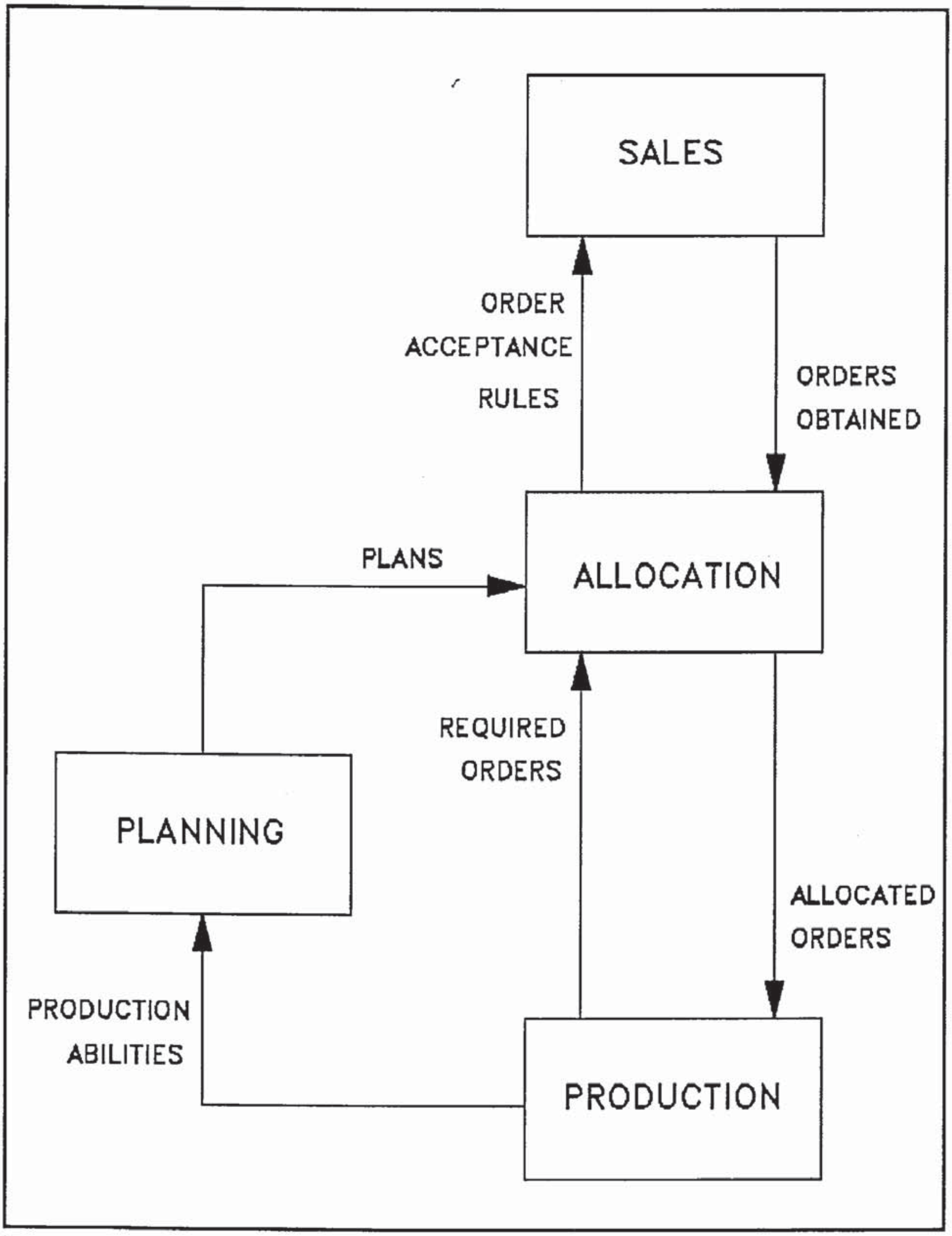
The use of unallocated arrears as a measure of performance for the project was thus agreed by the project team.

Analysis of the Causes of Unallocated Arrears

Having identified a measure of performance for the project, the author endeavoured to identify the causes of unallocated arrears and their contribution to the total. This task proved more difficult than initially expected mainly due to available data.

The figures used to calculate the total levels of unallocated

FIGURE 2.8 MODEL REPRESENTING THE PROCESS BY WHICH ORDERS ARRIVE AT WORKS.



arrears were obtained weekly by the SMP Planners by a simple analysis of newly allocated orders with the scheduled delivery week later than the `promised' delivery week. This total figure was broken down by product. No analysis was made within BSSP as to the causes of unallocated arrears.

Initially, the author considered that it would be possible to identify the most significant cause of unallocated arrears with relative ease. This belief followed from consideration of the causes of unallocated arrears for which seven categories were identified. Figure 2.8 provided a simplistic representation of the process by which orders were allocated. The causes of unallocated arrears could then be categorized as deficiencies in the outputs represented by the six arrows in the figure. For instance, orders contravening the `rules' would result in an overloaded orderbook. A seventh category can be seen as being due to the inflexibility of the process. That is adjustment of the `rules' due to any production shortfall would not be correctable over the length of the current orderbook.

Of these seven categories, the order entry `rules' and the level of orders obtained contravening the rules were considered by the author to be probably the major causes of unallocated arrears. It was considered that other causes could be dismissed lightly with the exception of arrears caused by poor plans and by the inflexibility of the system. It was felt that quantifying the levels of arrears caused in such a manner would be required.

Figures for planned capacity and output achieved over the calendar year 1986 showed long term variations. Such variations, when equated to shortfalls in production, appeared to give scope for some forty percent of unallocated arrears to be possibly attributable to poor planning and the inflexibility of the system. However such unallocated arrears would have to be accompanied by allocated arrears caused by the immediate shortfall in production. This corresponding level of allocated arrears was not evident. Actual levels of allocated arrears bore no relation to the levels expected.

The discrepancy between the expected and actual levels of allocated arrears was not due to allocators reacting to information other than plans and levels of allocated arrears. Examination of the data used pointed to the cause of the discrepancy. The production figures available represented tonnes of throughput. The production plans used resulted in tonnes of orders loaded. As discussed in section 2.1.3, these two figures did not necessarily equate. The apparent shortfall in production could be prevented from causing unallocated arrears by the use of downgraded stock and underproduction in fulfilling customers' orders. Such practice could have accounted for all of the discrepancy. Thus the levels of unallocated arrears due to poor planning and the inflexible system were not identifiable through analysis of this data.

A second attempt was made to identify the causes of unallocated arrears concentrating on the level of orders obtained which contravened the order entry rules. The analysis identified all orders on the orderbook with delivery leadtimes shorter than that stipulated by the order entry rules in force at the time the order was accepted. This data would not directly provide to the level of unallocated arrears caused by overbooking. The order entry rules are not applicable to a number of customers. However it was expected that it would indicate the order of magnitude of arrears caused in this manner.

The analysis was carried out on orders with Hot Rolling `promised' during the preceding ten weeks. The `promised' Hot Roll week was compared with the earliest week available according to the leadtime rules in force at the time the order was received. The results showed that the volume of orders breaking the leadtime rules far exceeded the level of unallocated arrears. This situation was possible due to the nature of the leadtime rules. The leadtime rules, being capable of responding only in terms of extra weeks in the leadtime, allowed such laxity to occur without significantly undermining the system as a whole.

The Decision to Build a System to Monitor the Causes of Unallocated Arrears

The leadtime system thus proved to be far from exact with regard to the levels of orders booked. Also the data available concerning plan fulfilment was not capable of confirming whether or not production plans were a significant cause of unallocated arrears.

This situation pointed to the need for a more detailed analysis. The order load would need to be matched directly to the planned capacity; in effect carrying out a roughcut allocation task.

Such a task suggested the building of a system to monitor the causes of unallocated arrears. Firstly the analysis could only be carried out using current production data as the required historical data was not available. The use of current data would require the analysis to be run over a period of time to cope with the intrusion of short term effects. Secondly the level of detail required by such analysis led to consideration of the task involved in carrying out such analysis. For instance, to identify bottlenecks in capacity, the analysis would have to consider minor mill facilities such as inspection lines.

The question remained whether or not the construction of a system to monitor unallocated stocks was justified. The main argument was that the only significant cause of unallocated arrears of which the author could be confident was the breaking of the leadtime rules. This cause lay beyond the jurisdiction of the project team. To tackle it, a greater burden of proof would be required than was then available.

A second reason was that a system to monitor unallocated arrears was seen as useful even if the project proved successful in reducing the level of arrears significantly. There was no reason to suggest that the problem of unallocated arrears would be or could be solved once and for all.

A third reason was found in considering possible project tasks given that different causes of unallocated arrears could be found as most significant. The roughcut allocation task to be carried out by the monitoring system appeared as a component in many of the possible project tasks.

The decision was thus made to build a system to monitor the levels and causes of unallocated arrears. The development of this system, to be known as UnAllocated Arrears Monitoring System (UAAMS), was approved by the project team. The development thus then became the first part of the project task proper.

2.3 ALTERNATIVE APPROACHES TO THE PROBLEM

Prior to embarking on the development of the computer system to monitor unallocated arrears, UAAMS, it is necessary to consider alternative approaches to tackling the allocation problem. This section first considers alternative approaches which could have been taken to structure the problem. Secondly, literature relevant to the project is considered as a possible source of alternative approaches to the problem.

Alternative Approaches for Structuring the Problem

The use of Soft Systems Methodology to stage a debate as to what the project should be tackling was one alternative which was mentioned above. Indeed, the author was quite prepared for the proposal to build the UAAMS system to be overruled by a consensus of the project team. It was considered that such a consensus could have pre-empted the results from UAAMS, dictating the cause of unallocated arrears which the project should tackle. Thus the proposal to build UAAMS was not isolated from the use of a debate as a means of obtaining a consensus within the project team. The proposal was made more in preparation for a debate than as an alternative to it.

Another approach to the allocation situation would have been to analyse the allocation task so as to identify which aspects of the task were problematical in nature. Such an approach was dismissed by the author. It has been argued by Checkland (1) that such an approach would restrict available solutions.

Possible uses of expert systems were also considered at this early stage in the project and can be seen as an alternative approach to tackling the problem. As a supporting system for the allocation task generally, an expert system would need a great deal of understanding of the production capacities and the production required. Development of such a system was seen as adventurous although the simpler UAAMS system, in carrying out a roughcut allocation, was not seen as alien to such an approach.

Another possible use of expert systems could also be seen in reviving the Linear Programme (LP) to provide suggested allocations. The major reason for its disuse was the need for allocators to fill in large maximum works capacity tables. As the allocators' only input into the programme, this had proved far from workable. Using an expert system to reduce the task required of an allocator in filling in such a table did appear to be a useful role for an expert system.

While expert systems did appear useful, the LP itself appeared to the author to suffer deficiencies beyond the immediate reason for its disuse. Firstly, its use of costs. Costs were not an input into the allocation decision. Costs were a part of the Annual Operation Plan within which the allocators worked. Thus the use of a LP to minimize costs resulting from allocation decisions did not appear sensible.

Secondly, maintenance of such a LP could have proved a limiting factor in the operation of the programme. The comparative relevance of the categories varied with time. Further, the categories would not remain fixed. Also the costs, even just transport costs, were an approximation based on percentage levels of interworks routings based on the AOP. To allow changes in categories without major adaptation of the LP the author believed that the number of categories would have to be increased. The LP solution, as discussed in section 3.4.6 below, relies on manipulation of aggregate order load which would be jeopardized by the use of too many categories.

The revival of the LP to provide assistance to the allocators was thus not seen as a necessarily useful approach to the problem. Indeed, it did not appear to be necessarily tackling the root of the allocation `problem'.

While the allocation function did not appear immediately suited to being supported by an expert system, this was not so of the SMP

Planning function. Consideration of this task suggested that the function then carried out by micro-computers in supporting the planning task should not be underestimated. An expert system supporting the planning function would either have to interface with spreadsheet software, spreadsheets which may change in layout, or would have to duplicate the spreadsheets' functions.

Consideration was also given to the decisions made within the planning process and whether they could be automated or only supported by an expert system. Through such analysis, the usefulness of expert systems as a support to the Planning function could be determined.

One point became apparent from considering the possible use of expert systems within the SMP Planning function. If considered seriously for development, the task set for this expert system appeared very difficult. Indeed, such a development appeared speculative in nature. While a means of analysing the efficacy of this particular expert system solution had been identified, no accounts of similar approaches to expert system development could be found by the author within expert system literature.

Literature Relevant to the Project Task

One requirement was to identify methods by which Production Planning and Control (PP&C) problems were normally tackled and to then consider the applicability of such methods to the allocation task. As well as this, PP&C literature obviously does not specifically mention the allocation of orders within a large multi-site strip steel manufacturer. Given this situation, PP&C needed to be confirmed as applicable to the allocation task within BSSP.

Much of the justification behind the project was the promise of expert systems as a problem solving technology. Thus the capabilities of such systems required consideration as did the methods by which expert systems are developed.

The appropriateness of PP&C methods and of expert systems to the project task is the subject of the next two chapters.

CHAPTER 3
TECHNIQUES AND APPROACHES TO PROBLEM SOLVING IN PP&C

The thesis considers the usefulness of expert systems to the allocation situation. To date, many expert system applications have been built and some are now in use within PP&C functions; some of these within the steel industry. However before reviewing expert system technology and its capabilities, it is necessary to consider alternative techniques for tackling the allocation situation.

This chapter considers the approaches to and the coverage of PP&C within the literature. A definition of what is meant by the term PP&C is followed by a discussion of the nature of the task carried out within PP&C.

As well as PP&C literature itself, the function of PP&C is also covered by the literature of Operations Management (OM) and Operational Research (OR). OM is a wider discipline than PP&C while OR can be described as a problem-solving discipline.

The literature of OM, OR and PP&C all act as an umbrella for a wide number of techniques, the techniques often being covered by more than one of them. The areas of study and the approach to PP&C advocated by them are shown not to have considered adequately the application of the different techniques and how to choose between them. Also discussed will be the concept of Computer Integrated Manufacture (CIM) and relevant work in this field reviewed.

The various techniques are then reviewed. The conclusion is reached that none of these techniques could be applied to the allocation task in a straightforward manner. Thus the focus of the thesis turns to the applicability of expert systems in chapter 4.

3.1 A DEFINITION OF PP&C

PP&C can be described as the function which provides a production facility with the plans and schedules required to achieve the desired sales output. It should be noted that these plans and schedules are not all destined for the production function; such plans can also be inputs into the sales function. Plossl, for instance, describes the objectives of most manufacturing firms as effective customer service, inventory control and plant operation. He goes on to describe the function of Production and Inventory Control as:-

> concerned basically with providing the information needed for day-to-day decisions required to reconcile these objectives in plant operation (13).

PP&C thus does not take customer service as given, which then leads to input into the sales function.

Another point illustrated by Plossl is the different terms used to describe PP&C. Plossl, in using the term Production and Inventory Control, simply emphasizes the control of inventory rather than the longer term planning aspects of the PP&C function; Plossl still covers the longer term planning aspects of PP&C. The term Production Control is used by other authors to mean PP&C (14,15).

Differences are not always confined to a difference in name. Koening's depiction of the organization of a manufacturing firm divides PP&C between Materials Management and Manufacturing Engineering (16). The author understands this split as being due to an extension of the role of Industrial Engineering, described by Prabhu and Baker as being responsible for the design of PP&C systems (15). Koening, in defining the scope of Manufacturing Engineering, has extended this Industrial Engineering role designing the PP&C system to include day-to-day control of part of it.

Plossl also attempts to make clear the divide between PP&C and the function of the "manufacturing people". In this, the author would contend that the divide is not easily placed. Wight, in stating that the limit of the PP&C function is to provide "valid attainable schedules" (17) skates over the issue Plossl wrestled with. The level of detail of the plans and schedules provided to other functions and the amount of pre-processing of information provided to PP&C is not definable.

Corke defines the objectives of PP&C, or as he terms it Production Control, as:-

> to enable good delivery dates to be offered, and to get customers' orders completed on time (14).

The divide between PP&C and the manufacturing and sales functions revolves around the words "enable" and "get". The author would suggest that the level of detail of this `enabling' will vary from firm to firm, the level being determined by custom and practice.

The author would like to emphasize here the effect of the placing of these divides on the conflict of objectives which is widely agreed to impose on the function of PP&C. The placing of the divides between PP&C and other functions will determine the ability of PP&C to manipulate these objectives as opposed to accepting them or negotiating changes in them.

3.2 THE NATURE OF THE PP&C TASK

PP&C is widely seen as a difficult but important aspect of any manufacturing organization. The benefits to such organizations from improved PP&C are large. Both Prabhu & Baker (15) and Plossl (13), for instance, see improved PP&C as "vital" as this can lead to very significant increases in company productivity. Others, for instance

Tompkins (18) and Fox (19), see the need for improved control systems to reduce the costs of PP&C itself.

But often these benefits are not realized. As Barber and Hollier state:-

> Many companies already using computer-aided production control techniques are not using them effectively and some have not improved on previous manual systems (20).

The difficulties engendered by the PP&C task are thus highly significant. Considered here is the description of these difficulties found within the literature. An underlying question is whether or not the PP&C literature is relevant to the allocation task.

Conflicts of Objectives within PP&C

Within the literature, many authors describe the essence of the difficulties which PP&C attempts to overcome. Galgut, for instance, is emphatic that the problem is due to imprecise information and inadequate management (21). Fox (19) sees the problem as the need to be able to adapt so as to cope with complexity and uncertainty.

The complexity and uncertainty are identified in some form or other by most authors. Some authors continue from this point to conclude that the PP&C "system" will never be able to cope with all the problems it encounters (14,15,22,23). Prabhu and Baker (15) suggest the handling of these problems which PP&C is unable to cope with shows the need for the "system" to have adequate authority within the company. Corke (14) similarly sees the need for PP&C to enable wider decisions and thus to prevent conflicting objectives; the same conflicting objectives presented above as Plossl's definition of the PP&C function. Prabhu and Baker (15) also mention the conflict between planning decisions carried out at different levels within the organization, a conflict in objectives identified elsewhere (24,25).

It is not made entirely clear by all these authors that they would actually endorse the proposition that the conflicting objectives are the major cause of difficulty within PP&C. However the problems of conflicting objectives are given prominence in their accounts. For instance, Prabhu and Baker (15), while describing the control aspects of PP&C, state that if this task is ensuring the right quality, cost and delivery, then the main problem is defining precisely what the word `right' refers to.

The prominence of these conflicts within this literature does match the author's understanding of the importance of such conflicts within the allocation function. The problems posed by such conflicts are seen by the author as unstructured or `soft' problems as described by Wilson (26).

In recounting these conflicting objectives, the literature has in the author's view fallen short of describing the full richness of the reality of PP&C. Scudder and Hoffmann (27) are alone in recounting the "end-of-the-month" syndrome where, towards the end of each month, the PP&C function is forced to maximize the current month's output at the expense of the following month. Customer indiscipline in changing its requirements or expecting preferential treatment is mentioned by for instance Corke (14), although the reviews of Ingham (28) account this issue some prominence. Corke alone countenances the existence of non-realistic internal delivery dates (14).

Such realities within PP&C can be seen within the allocation function. They are symptoms of the conflicting objectives discussed within the literature.

<u>The Nature of Complexity and Uncertainty within PP&C</u>

It was argued above that these `soft' problems encompassing the conflicts of objectives within PP&C follow from the inability of PP&C to cope with the complexity and uncertainty of the production facility. That is the required objectives could be more easily clarified if PP&C could maintain complete control over production. So what then is the nature of this complexity and uncertainty?

The literature does not always cope well with describing this complexity and uncertainty. Van Dierdonck and Miller (29) make an almost theoretical description. Complexity, they say, is a function of production volume, diversity, repetitiveness and task interdependence while uncertainty is generated from variations in supply, demand, production and objectives.

This treatment of uncertainty concentrates on forecasting problems almost ignoring the problems engendered in data collection. Galgut describes all such collected data as imprecise (21). Bestwick and Lockyer describe it as often grossly imperfect (22) while Prabhu and Baker describe the data as seldom available (15).

The uncertainty then revolves around the failings of data collection and the inability to forecast future events. Galgut has shown that the uncertainty is only a problem in that the complexity of PP&C requires precise data (21).

Turning then to complexity; in the author's view, discussion of the complexity of a production facility, for instance by Prabhu and Baker (15) or by Starr (30), suggests that there is a model of such facilities which the PP&C function should be utilizing; that the number of orders, items, processes, machines, interdependence between orders and stages in manufacture define <u>the</u> complexity of a production facility.

The author contends that it is more appropriate to consider complexity as the complexity PP&C has to handle to achieve an adequate level of control. For instance, the unit of measurement of capacity and work load is often artificial, selected to satisfy a number of considerations (14); it is not pre-ordained.

Further, the author would contend that uncertainty results in added complexity for the PP&C system to handle. For instance, the problems of plans diverging from reality with time, discussed by Corke (14), can be seen as being a factor of uncertainty. The problem would be overcome through more detailed or more frequent planning. This would add to the complexity of PP&C. Similarly a requirement to take account of arrears (21) or yields (31) can be seen as a factor of uncertainty causing an increase in PP&C complexity.

A further point with respect to complexity is the inter-related nature of PP&C problems. Delivery dates, Work-in-Progress levels, batch sizes, stock levels and throughput times, presented by Prabhu and Baker as typical PP&C problems (15), are all inter-related.

The complexity and uncertainty of the allocation task thus does match the descriptions of PP&C within the literature, although the literature often could be described as too clinical. Accounts such as Mather's (23) are perhaps more realistic, although his use of the word "chaos" is rather strong.

The function of PP&C can thus be described as often problematical. It should be mentioned that such situations are not unique to PP&C. The problems faced by Data Processing departments are widely seen in a similar light.

Reflecting on the Nature of the PP&C Task

The author has argued above that the conflicting objectives imposed onto PP&C functions can be seen as unstructured or `soft' problems. The author has also presented various aspects of complexity which PP&C also has to cope with. These aspects of complexity can be seen as presenting a structured or `hard' problem. The proximity of these `soft' and `hard' problems of such profundity within PP&C appears to the author to set PP&C apart from most other standard functions within organizations.

Despite this situation, and the importance of the task, there appears to have been little research into the real-world process of PP&C. Fox for instance, while investigating the scheduling task within a job shop within Westinghouse (32) identified that the schedulers spend some 80-90% of their time collecting data from around the plant. This figure, found to be mirrored in RAF aircraft repair scheduling by Grant (33), has not been highlighted by other authors.

In 1979, Gupta reviewed the research into flow shop scheduling. His conclusions, while rather scathing, are perhaps not inappropriate ten years on.

> The mathematical theory of flow shop scheduling suffers from too much abstraction and too little application. The practical use of flow shop scheduling techniques is rare. This questions their suitability. In spite of 23 years of research, we know very little about the practical flowshop scheduling problem except that it is an often occurring problem. (Research should) be inspired by more real life problems (34).

3.3 APPROACHES TO PP&C

The task of PP&C has been defined and the nature of the task described. Approaches to tackling the PP&C task are covered by such disciplines as Operations Management, Operational Research and Computer Integrated Manufacture as well as the literature of PP&C itself. These approaches are reviewed here. Not covered here are expert systems which are the subject of the next chapter although some expert system research is covered here as part of Computer Integrated Manufacture.

3.3.1 OPERATIONS MANAGEMENT

Operations management (OM) is concerned with the management of physical resources for production, whether the product be a manufactured item or a service (35). It thus encompasses PP&C.

By taking a wider view than PP&C, OM can be seen as a possible source of input into PP&C. Considered here are the strategic considerations of OM relevant to PP&C, classifications of production systems and discussion of the approach OM has towards the use of different PP&C techniques.

PP&C Strategies from Operations Management

It is widely agreed that the sub-division of the operations function results in a sub-optimal performance for the facility as a whole unless some central coordination is achieved. It is also widely agreed that this problem is central to OM.

While the problems of sub-optimization have been recognized for some time, recently more emphasis has been placed on the interactions between corporate strategy and OM (35,36). Thus sub-optimization is now being seen with regard to the whole organization rather than in terms of only the operations function. Hill discusses at some length the "great business divide" and the way objectives differ between manufacture and marketing (37). He sees the need for manufacturing to become less reactive and to promote a coherent "manufacturing strategy".

There is general agreement that the effectiveness of the whole is achieved through the linkage of strategic requirements to the lower level PP&C decisions. Buffa and Sarin see `making operational decisions strategic' as one of six basic components of a manufacturing strategy (35). They identify the development of Just-in-Time (JIT) manufacture as an example of this and describe the Operations Audit as a means of obtaining a realistic estimate of the operations function's effectiveness. Other similar work includes that by Voss (38) and Van Dierdonck and Miller (29), the latter specifically with regard to PP&C.

Others are less prescriptive but also identify the need for measures of performance to identify strategies or objectives (15,36). Constable and New temper the idea of such strategies with the comment that implementing strategies of this sort is not easy (36).

Miller tackles sub-optimization within OM by presenting a breakdown of the functions of OM for the purpose of identifying the performance of different aspects of the OM functions in a company (39). In such a way the effectiveness of PP&C functions could be identified and a decision made as to what areas need improvement.

A more specific view which has been proposed with regard to the sub-optimization issue is the effect of limitations of PP&C on a manufacturing organization. Prabhu and Baker identify the need to match the abilities of an operations function with the needs of the market, and extend this to matching the complexity of the production process to the abilities of PP&C to control production (15). They state:-

> If production is complex, so too will be the production control system and levels of staffing required. Production control procedures are often simplified considerably and staffing levels reduced through reorganizing what the manufacturing unit is doing.

Hill agrees with this stance and suggests that such strategy tends not to be considered during the design of PP&C systems (37). However, some types of production are linked with the simplification of PP&C. JIT and Group Technology are instances of this, although Dale and Russell (40) have concluded that Group Technology is not free from its own PP&C problems. It should be noted that balancing the complexity of the production system with the required complexity for production control exemplifies the cybernetic principal of `requisite variety' (41).

The status of the project was such that the idea of linking PP&C requirements to an overall manufacturing strategy, which itself required defining, was impractical. Further, the author agrees strongly with Constable and New; implementing such a strategy would be far from straightforward (36).

<u>Classification of Production Facilities</u>

Since the 1950's, attempts have been made to produce useful classifications of production systems (36). Consideration is given to attributes of the product (20,36), the production process (15,35,36) and stocking policies (15,35,42) to varying levels of detail. Schmitt et al present a variation on these more usual forms (43).

Most of these classification schemes are conceptually easy. However, the author should mention one that is not so straightforward. That is the dichotomy Push-Pull which is often introduced to explain the Just-In-Time system. The common differentiation between `push' and `pull' systems, described for instance by Buffa and Sarin (35), is as follows. Work within a `pull' system is driven by the requirement to supply down-stream processes. Work within a `push' system is driven by the presence of jobs waiting up-stream of the process in question. Todd (44) has described a successful Pull system as simply being adequately primed. Pull systems thus rely on a balance between the abilities of the plant, and the accuracy of the forecast of future demand. The author agrees with this view and would state that such discussion of the benefits of Push or Pull as given by Hill (37) are in the main misleading.

Some of the work classifying production systems is directed towards identifying the most appropriate production control systems. For instance the classification of batch production systems of Barber and Hollier (20) attempts to identify the most appropriate areas for computerization. Also Seward et al (45) compare the applicability of Closed-Loop Material Requirements Planning (MRP) and Hierarchical Production Planning (HPP). However this work is by no means conclusive. Barber and Hollier stress that their work does not provide the definitive computerization program, only an input into the decision-making process. Seward et al, while comparing only two of the available techniques, appear reluctant to delineate the appropriateness of even these two techniques positively.

The more usual coverage of such classifications does not consider the relative appropriateness of techniques. Within OM, the classifications are not fixed. That is, OM encompasses the design of the production facility which defines how the facility will be classified. OM thus considers the facility design options open to an organization and the general advantages of each (35,36,43) Within PP&C literature, a similar use of classifications can be seen except at a more detailed level (42).

Choice of Techniques within Operations Management

In the past, OM was technique-oriented (46). Such an approach has been defended by Bestwick and Lockyer (22). They consider the problems within OM as either 'programmable' or 'non-programmable'. The techniques of OM tackle the 'programmable' problems leaving management more time to solve the 'non-programmable' problems. Within this framework, the techniques are not treated in isolation. For instance, with respect to scheduling, Bestwick and Lockyer state:-

> It must carry with it the full weight and responsibility of top management, and consequently must be part of the

corporate policy of the organization. Scheduling is too important to leave to the schedulers.

Similarly Constable and New (36), in discussing the various sub-divisions of OM, consider the problems which are often encountered rather than prescribing suitable techniques. In this respect, the most relevant charge levelled at technique-oriented OM is that it does not attempt to identify the most suitable techniques, as levelled by Hill (46). Hill sees choosing appropriate techniques as the important part of OM. However, the choice of techniques made by Bestwick and Lockyer would be described by the author as pragmatic. The choice is based on what the authors see as being used and useful. Constable and New in presenting the problems, are perhaps attempting to structure such a choice.

Certainly these accounts do not deal with the new technique-based philosophies which Hill warns of. OM has yet to provide a structure through which such choices could be made. Treatments of these new techniques mainly describe only the individual techniques in isolation. Buffa and Sarin (35), for instance, describe four techniques; Materials Requirements Planning (MRP), Manufacturing Resource Planning (MRP2), Optimized Production Planning (OPT), Just-In-Time (JIT). But they give only brief consideration of the differences between MRP and OPT. The relative applicability of the four techniques is not discussed to the author's satisfaction. By no means does this coverage of techniques add to the coverage to be found within PP&C literature.

The Usefulness of Operations Management to PP&C

Given the argument presented above, the author can see little input into PP&C from OM which would have been useful to the project in hand.

It must be borne in mind that PP&C is a large sub-division of OM.

The divide between PP&C literature and OM literature cannot be drawn definitively. Considered here has been only the wider view afforded by OM.

3.3.2 OPERATIONAL RESEARCH WITHIN PP&C

Determining the approach which OR would make to tackling the allocation problem can only be understood in terms of the developments which have occurred in OR. Thus an outline of the history of OR is given followed by an attempt to answer the question `What is OR?' and coverage of the literature discussing OR methodology. This is followed by three sections discussing generally the OR techniques appropriate to PP&C; comparisons drawn between them, their usefulness and accounts of the work concerned with improving their usefulness. Two further sections cover `soft' OR.

The History of OR

The History of OR has been well covered in the literature. Recently, accounts have been given of early OR (47), its inception during World War Two and spread to civilian use (48) and the effects of computer developments on the later history of OR (49). The other major strand to the history of OR is the OR `crisis' of the late 1970s epitomized by Ackoff's paper "The Future of OR is Past" (50).

The first symptom of crisis within OR was the low implementation rate of OR work, which had become an issue as early as the 1960s (51). By this time, OR had become an academic subject, developing and teaching mathematical techniques, the relevance of which was questionable (52,53). Developments in computer technology allowed the increased sophistication of these techniques.

The crisis in OR was more profound in the USA (50) where Woolsey (54) saw the academic emphasis as the main reason for the crisis.

Ackoff's interpretation of the crisis was that OR practitioners were attempting to tackle problems rather than the `mess' faced by their clients, that problems were being distorted to fit solutions by for instance using optimizing techniques, and were not involving the stakeholders in the analysis. Ackoff was not alone nor was he the first to make such diagnoses.

OR has certainly developed since this crisis. However, OR is still seen by some as technique-driven and optimizing, such a view being used for instance by Jackson (55) to promote non-OR techniques.

One development within OR has been the adoption of OR techniques by non-OR practitioners (52,56-58) due in part to easier-to-use computing (56) especially micro-computers (57). In this respect, OR has begun to deliver Decision Support Systems (DSS) rather than answers (49,59).

Another development has been `soft' OR. This encompasses a set of approaches developed by, for instance, Eden et al (60), Bennett (61) and Mason and Mitroff (62) which attempt to cope with the `mess' and involve stakeholders in the analysis. A similar approach, though not integral with OR, is Checkland's Soft Systems Methodology (SSM)(1).

The use of prototyping within the development of computer systems is also seen as an approach of OR (49,56).

What is OR?

The OR Commission, reporting in 1986, decided that OR in practice should be taken to mean anything recognized as such by Society members (56). Their findings as to what was thus OR did not directly identify optimizing or cost saving as the aims of OR in practice. They found little explicit use of mathematical techniques, the approach or methodology used being described as `pragmatic'. The commission found little in the way of a definable methodology.

The main drawback to the health of OR found by the Commission was that the outside perception of what is OR is restricting the tasks which OR practitioners are called on to perform.

On the whole the commission suggests that OR in practice is not in crisis.

In 1953, Morse adopted a similar definition of OR to that used by the OR Commission. He added the rider that the methods of OR were those reported in the journals (63). Currently it is strongly evident that the findings of the commission as to the practice of OR only bear a passing resemblance to the heavily mathematical content of OR journals. But there is no reason to believe that practical OR has substantially changed since the crisis while the literature has not; Dando and Sharp had made an identical observation in 1977 (64) showing the fundamental `hard' precept of OR was even then a myth. The author would contend that while the introduction of such concepts as DSS and soft OR into the literature is a positive move for OR, the causes of the crisis still exist. That is the literature does not reflect the reality of OR. The OR Commission (56), in recommending a new magazine (OR Insight) aimed at OR practitioners rather than academics, appears to conform with this view.

OR Methodology

The traditional OR methodology, or the description of the process of OR, consists of a list of stages revolving around the use of a model through which a solution can be analysed (65). Such descriptions have been criticized.

Schwenk and Thomas (66) and Tait (67) suggest that the methodology should be used flexibly and iteratively; Ward describes this as a learning process (68). Tait also stresses the need to be conscious of the `weltanschauung' driving the analysis; generally that the constructs of soft OR should not be ignored within traditional OR.

Others have considered how to incorporate soft OR into the traditional methodology. Woolley and Pidd identified within the literature what appeared to them to be four `streams' concerning the first `problem formulation' stage of the methodology (65). The use of these `streams' they took to be decided by personal preference. One of these streams, the people stream, can be seen as using soft OR as a front end to the rest of the process.

Jackson, rather than accepting the methodology, criticizes it for not being able to cope with multiple perceptions or complexity (69). To overcome this he presents a framework of problem types (70). Thus categorizing the problem would appear to become the first stage in a wider methodology, this first stage leading on to use of a soft, hard or other appropriate methodology.

Robins (71) combines soft and traditional OR by considering three `perspectives' of the process of OR. One is a version of the traditional OR methodology. The second, the political view, presents what could be called a soft definition of the problems tackled by OR. The third perspective considers the models through which the problem situation is and can be viewed. However Robins (71) describes the process of OR as "iterative and somewhat messy sequence of steps chosen during the analysis to satisfy the immediate demands of the investigation." He is emphatic that OR methodology is not definable.

Description of the process of OR then has yet to fully incorporate either the soft approaches, or the concepts of prototyping or DSS within it. Nor is the difference between providing understanding to the client and providing a solution adequately represented; both are widely understood to be valid outcomes of traditional OR.

Choosing between OR Techniques within PP&C

While description of the process of OR may not have been well advanced, the other contribution which traditional OR could possibly provide is in choosing between techniques to use within PP&C. Generally this is not an easy task. Haley has suggested that there are too many OR techniques for an OR practitioner to cover in such a choice (72).

However some work has been carried out with regard to the application of techniques within PP&C. Eilon, for instance, has compared different aggregate planning techniques (73) while Alexander has identified several such comparisons of different priority rules for job shop scheduling (74).

The author's criticism is that the work is quite theoretical and thus not driven by the important aspects of real world problems. Work by Muhlemann et al (75) for instance showed that the frequency of rescheduling was a far more important factor than the choice of priority rule regardless of the criteria of evaluation. The identification of such an important factor in priority rules is seen by Fox and Kempf as profound and quite new (76). Such an occurrence however is not unique. Another instance concerns the Economic Order Quantity (EOQ) construct. Kingman identified that the assumptions concerning periods of credit did not match the real situation (77).

Ritzman et al (78) carried out simulation work in an identical manner to the more standard OR work except that they use a more detailed simulation and compared different techniques rather than different versions of the same technique. In doing so, Ritzman et al avoid the author's criticism of being too theoretical. Such work is rare.

Usefulness of OR Techniques within PP&C

Traditional technique-based OR has been criticized for being too narrow in outlook (79-81), unable to support problems that are not well defined (82). Thus it is seen as inappropriate for planning and other high level functions within a company (52,62,83). The literature suggests from this, and indeed a few authors state (69,84), that OR is a useful approach at only lower levels within the company.

Scheduling (52,62) has been specifically mentioned as being such an application. But scheduling applications of OR have also been described as non-existent (85), of little value (79) and quite restricted (25). Even one area of success, that of stock control, has been criticized as providing only coincidental improvement (77,86).

The Development of OR Techniques for PP&C

The techniques of OR are not then universally seen as useful within PP&C.

In 1977, Eilon discussed the different views held by academics and users with respect to PP&C. At the same time Eilon described the ineffectiveness of the practice of PP&C (87). Eilon's view of the problem was that PP&C "systems" ossify the process of PP&C, something not recognized then by academics. While discussing the scheduling task in 1978, Eilon (88) was less specific as to the reasons for "the glaring inefficiencies (that) are rife in production systems". He saw two possibilities for this; either the theoretical work was misplaced or the practitioners had failed to exploit this work properly.

There is acceptance within the literature that certain techniques are seldom used. One example of this is aggregate planning (89-91). While all of these authors agree as to the reason why aggregate planning systems `ossify' the task, they do not propose the same path to overcome this problem. Nor has the literature yet shown the relative efficacy of such paths.

A survey by Oakland and Sohal of UK industry (92) identified only forecasting and stock control techniques as being used widely within PP&C, a conclusion which had already been drawn (36). Oakland and Sohal identified the main reason for other techniques being little used as a lack of knowledge of the techniques within the companies. Oakland and Sohal conclude that the need is to overcome this "barrier to acceptance" of the techniques (85). However the linkage presented between the techniques and the benefits from their use is seen by the author as tenuous.

Heard (93) presents a different perspective of the gap between theory and practice within PP&C, something he describes as "a substantial schism". Heard puts the root of the schism as a problem of communication between practitioners and theorists. He sees the way forward through structuring the subject such that it can be understood by both parties. Such structuring of PP&C will be discussed in the next section.

While the gap, schism or lack of acceptance of traditional techniques is generally agreed to exist widely across PP&C, the literature shows widely differing responses to this situation.

<u>Soft OR</u>

The various strands of soft OR grew from the need to involve the client set in the analysis. Mason and Mitroff (62) for instance developed an approach to problems, called Strategic Assumption

Surfacing and Testing (SAST), to overcome disagreements concerning the basic definition of the problem. Checkland (1), reflecting on the limits of Systems Engineering rather than OR, found the concept `problem' itself needed to be refined. He identified what he termed unstructured or soft problems in which the required objectives cannot be articulated. Thus the resultant development of Soft Systems Methodology (SSM) does not aim to solve problems. Rather it aims to improve situations. Similarly, mainstream OR has also developed refined ideas of problem-solving, for instance by Eden (94).

The soft OR approaches can be seen to lie on a spectrum varying from `soft' to `hard'. Checkland describes such a spectrum (95). However he does not plot onto this spectrum the soft OR approaches, rather the various philosophies of approach advocated by the likes of Vickers or Churchman (1).

The softer approaches, as well as coping with the different views of the stakeholders, are also highly introspective. The process of intervention becomes, as Checkland describes it, systemic, and thus is able to cope better with the organizational `mess' (95).

The majority of the soft OR approaches are termed soft in that they explicitly involve the client set. An example of what the author would call the most basic of soft approaches is given by Butterworth (96) where effectively hard constructs are presented to the decision-maker who, assisted by the analyst, structures the solution path. The overriding problem being tackled by the analysis remained unquestioned but unlike traditional OR, the process goes beyond a straightforward assessment of alternatives (97).

The inclusion of multiple stakeholders within such processes leads to processes more generally seen as soft OR. The work of Mason and

Mitroff (62) and Bennett (61) allows reconciliation of differing objectives within an ostensibly hard, numerical construct. The majority of the approaches described as Soft OR (97) appear to conform to this description.

Ackoff's participative planning (98) can be argued as having not progressed beyond these approaches on the spectrum. While Ackoff presents many aspects that parallel SSM, he has been criticized for adopting a goal-seeking model (1). Ackoff in recognizing the soft 'needs', suggests that a participative approach would achieve a workable agreement.

The softest approaches are Eden's Cognitive Mapping (CM) and Checkland's SSM. Both centre around the use of a modelling technique to explore the problem situation and the views held by the client set. Both approaches are seen as being developed within the concept of action research (1,99).

Cognitive Mapping is a process in which causal loop diagrams are constructed from the views of stakeholders. These diagrams or Cognitive Maps are combined and amended by the stakeholders, this process being supported by computer tools. As well as facilitating the circulation of different views, the process can also be used to work towards a "consensual" view (99) which the author assumes to mean a workable agreement.

Much of the recent work regarding the use of Cognitive Mapping has been directed towards strategic planning within Eden's Strategic Options Development and Analysis approach known by the acronym SODA (100).

SSM, rather than using a causal loop diagram, uses activity diagrams. These are derived from a Root Definition which defines an objective (26) or purpose for the system represented by the activity diagram. Comparison between activity diagrams (called conceptual models) and the real situation allows learning about the problem situation. A number of uses for SSM have been developed (1,26).

It has been stated that the conventional description of SSM has remained less than precise (101). Certainly the wide ranging uses of SSM are not covered fully by descriptions of SSM within the literature (102,103). Such accounts Checkland describes as the generalized account of SSM (104).

The description of SSM is not straightforward. Checkland stresses that SSM is a methodology; "a set of guidelines or principles" which can be tailored to the characteristics of both the situation and the user. Yet even the highest level description of SSM (103) still appears in a generalized form rather than an all-embracing description. To the author, this generalized description of SSM appears deficient.

Concerning the Description of Soft Systems Methodology

In presenting his "bird's-eye view" of SSM (103), Checkland presents SSM as an evolving methodology. He sums up by presenting a number of points of overall learning from the development of SSM which the author now considers. The author would agree that SSM is "a systems-based learning system". However the author does not believe that its use "has to be participative" although he does accept that SSM does not constitute a body of "professional knowledge".

Considering these points further, it can be seen that one stage within the SSM `learning system' is the identification of feasible and desirable change. This stage is usually considered to be a debate, suggesting an increased level of participation compared with previous stages. That is an increase either in the number of personnel from the problem situation involved or in the detail of discussion with those personnel. That the stage is identifying organizational feasibility also suggests this.

For the use of SSM to be participative the author would expect more than just active involvement during the debate stage. To be described as participative, the author would expect the process of SSM to either involve stakeholders within the earlier stages of the approach, or present the earlier learning stages to the wider audience during the debate rather than just the learning from those stages.

These requirements, which the author sets to define a participative use of SSM, both become difficult when the use of SSM is to support some other body of 'professional knowledge'. An instance of this can be seen in the provision of computer systems. SSM can be used in a participative way to provide a primary task model (26) or to identify organizationally feasible and desirable systems. When however this also involves technical feasibility, the technical 'professional knowledge' is best coped with by the professional by considering himself as within his own problem situation. The organization then takes the role of an environment for the analysis. In this manner, SSM can be used to determine actions and involvements with the organization.

Such use of SSM remains non-participative simply due to the problem boundary enclosing so few stakeholders. Defining the problem boundary in such a manner is not considered explicitly within SSM (105).

The author had made such use of SSM prior to the inception of this project within the design and implementation of business computer systems, as mentioned in chapter 1. Such use of SSM is certainly not covered by the generalized description of SSM.

One description does encompass such use of SSM. Checkland (101) once presented two different modes of use of SSM. Mode 1 equates to the generalized version of SSM "used to carry out a project" or, in other words, to tackle a problem within an organization. Mode 2 Checkland described as the use of SSM to plan and "do thinking" about a project. This Mode 2 version of SSM can be otherwise described as using SSM within the problem solving system (1) or in other words the problem boundary encloses only the activities carrying out the project rather than the activities which the project is acting on.

As well as legitimizing a particular form of SSM, the author sees this description of Mode 2 use of SSM as of further significance. In extending the description of SSM beyond the standard `generalized' form, it can be seen as the first stage in the development of an all-embracing description for SSM.

Implications of OR for the analysis

Much of the OR literature discussed above provides little assistance to the project task. Description of the process of OR has developed little beyond the traditional pre-crisis description and the work considering the choice of technique is argued by the author to be too theoretical.

Soft OR did present constructs of relevance to the allocation problem. The conflict of objectives within PP&C after all has been argued above to be `soft' in nature. However the complexity of PP&C has been also argued to be `hard' in nature.

In such situations, the widely agreed role of soft OR would be to identify some workable consensus through which a traditional hard approach would be feasible. This is analogous to the generalized, Mode

1 version of SSM.

The traditional OR approach would be that the area of concern can be modelled regardless of what consensus is obtained. Indeed the model would allow the situation be understood and thus hasten a consensus.

The problem the author sees with the 'Mode 1' soft approach is that traditional models may be necessary to understand the implications of any discussions. And the 'hard' approach would require direction to prevent the modelling from becoming un-necessarily complex or solution-driven.

The mode 2 use of soft OR in which the soft constructs are used to plan and to assist the project does appear useful within PP&C where both soft and hard 'problems' exist side-by-side.

3.3.3 PRODUCTION PLANNING AND CONTROL

While there is general agreement that the task of PP&C is difficult, there is no consensus as to how to improve the operation of a PP&C system. Many authors are proponents of particular techniques; Goldratt with Optimized Production Technology (OPT) (106), Wight with Manufacturing Resource Planning (MRP2) (17), Plossl with his approach based on control of throughput time (8). Others consider that one single technique cannot be the solution to all PP&C situations. Reinfield, for instance, states:-

> Any single idea may be very valuable to a firm, or it may be harmful if used improperly, or even if used at all (107).

The question to be answered here is: how does the PP&C literature assist in identifying these 'valuable ideas'?

Firstly comparisons between techniques within the PP&C literature will be reviewed. Then literature covering the structure and classification of the PP&C task will be reviewed, a structure of PP&C

being developed from some of this work. Finally, the role of computers will be discussed.

Choosing between PP&C Techniques

The advice given by PP&C literature with regard to the choice of appropriate techniques is often contradictory. On the whole comparisons involve no more than two techniques.

MRP is stated as suitable to production with an "upside down bill of materials", steel production being stated as an example of this type of production (17). Indeed, MRP is stated as being "not universal but applicable to most manufacturing" (108) or suitable for any manufacturing involving dependent demand between production stages and involving discrete items (109). Steel production certainly conforms to this description. However Haglund concludes that MRP is not needed within primary metal production (110) while Fiora and Pitzer state MRP is not appropriate to steel production (111). They describe an alternative approach used in US Steel based on levels of Work-In-Progress (WIP) and calculated leadtimes, a scheme also used within BSSP.

Comparing the relative applicability of MRP and Hierarchical Production Planning (HPP), the production carried out by BSSP is certainly capacity rather than sales driven. Thus to BSSP, HPP is fundamentally more appropriate than MRP according the Seward et al. Indeed, BSSP has all but one of the seven attributes described by Seward et al as the attributes which identify HPP as being more applicable than MRP (45). Etienne, comparing the applicability of MRP with methods based on Economic Order Quantity (EOQ) states that EOQ is more appropriate to industries such as steel (112), contradicting Buffa and Sarin who see EOQ as only suited to independent demand (35).

Other authors compare such techniques as MRP, JIT and OPT in this same manner (113,114). The assumption is that the techniques are mutually exclusive. Within this assumption, concepts can be transferred from one technique to the other as Buffa and Sarin describe between OPT and MRP (35).

This assumption of mutual exclusivity is not made by all authors. Andersson et al investigate the usefulness of MRP within HPP (115). Billington et al see a similar coupling of the two techniques but from an MRP viewpoint rather than from HPP (116). Krajewski and Ritzman see MRP as a middle level technique with HPP techniques operating above and simulation below (117). Swann argues that the function of MRP is separate from the function of OPT (118). Leahy describes the integration of OPT and JIT techniques (119).

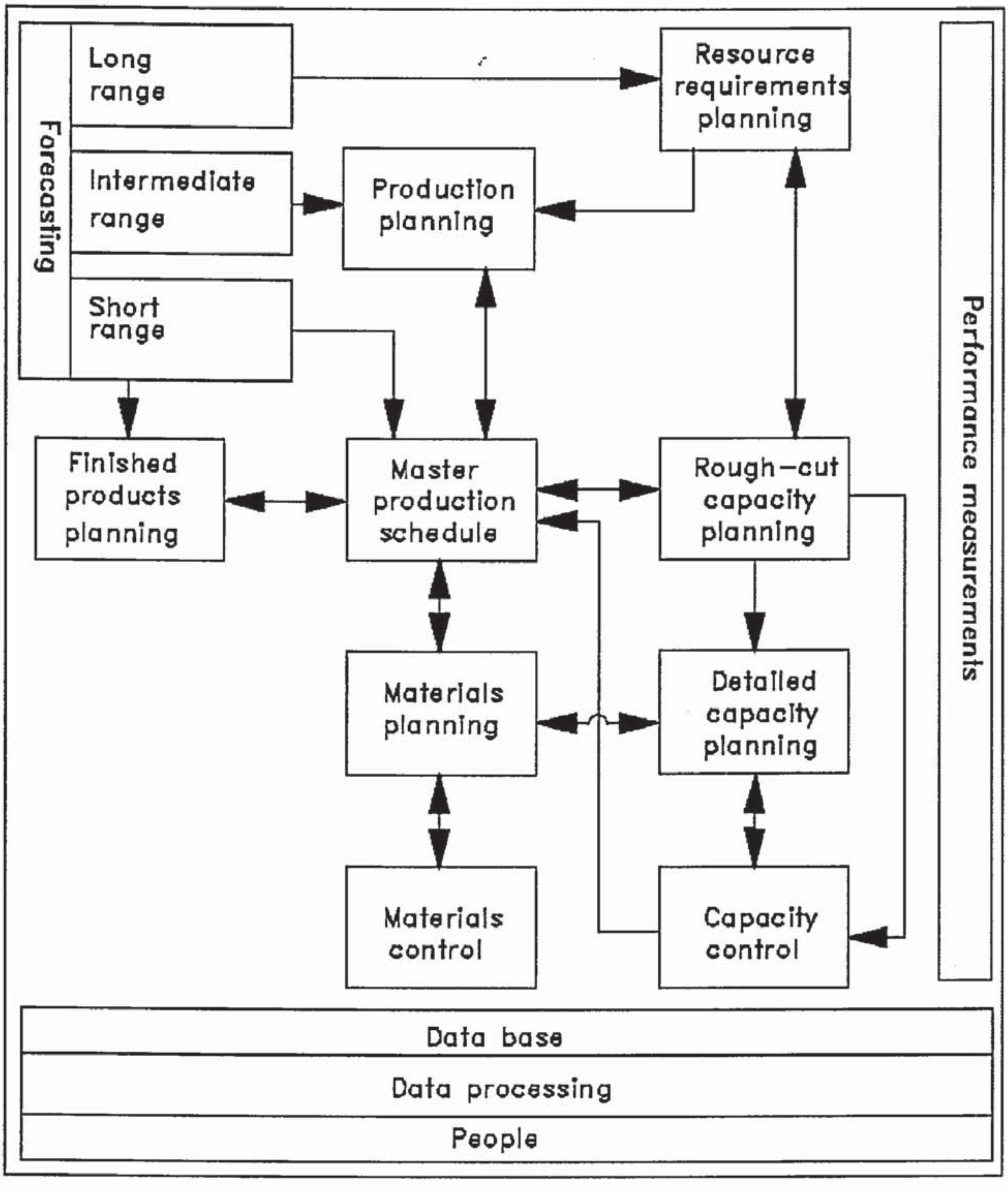
To make sense of such differing coverage of these techniques, the author would argue that there is a need to identify the structure onto which these authors are placing these techniques.

Structuring and Categorization of the PP&C Task

Large amounts of work have been published presenting structures for the PP&C task. There is however no standard presentation of this structure. Many different types of structure can be found. These include functional sub-divisions (120), taxonomies (20,93,121), lists of PP&C tasks (20,36,110) and diagramatic representations (13,14,25, 39,42). Other structures are presented representing the structure imposed on PP&C through the use of a particular technique, for instance MRP (17) or HPP (122).

If any of these structures can be accepted as a mainstream representation, one is presented by a number of authors (13,15,123) and reproduced in figure 3.1. Plossl has stressed that this structure is

FIGURE 3.1 THE STRUCTURE OF PP&C SYSTEMS AS MOST COMMONLY FOUND IN THE LITERATURE.



common to all manufacturing industries although the same elements in each control system will not always carry the same weight (13).

A more usual approach to this differing emphasis within PP&C is to categorize the different types of production system and the type of service offered the customer (28,42,88,124). Even these categorizations vary significantly amongst themselves.

The importance these authors impart into the correctness of their structures or categorizations varies. Some authors, for instance Plossl (13), Barber and Hollier (124) and Heard (93), suggest that they are presenting the structure of the PP&C task. Some authors may appear to have identified such a structure. Barber and Hollier do identify their six company types through the technique of numerical taxonomy. However the author would suggest that the variables significant within this analysis may not be those significant to the task of PP&C.

Other authors, for instance Eilon (88) and Miller (39), point out that the structures presented are not meant to be definitive.

Within the set of those authors presenting definitive models of the PP&C task, Corke (14) tackles the task of presentation in a different manner to the rest, although Bennett (125) almost achieves the same presentation manner. Corke does not present the structure of PP&C as a model of the PP&C task per se. It is presented as the minimum necessary set of activities required to achieve effective PP&C. This is very reminiscent of the use of a conceptual model within SSM, discussed above as part of `Soft' OR.

A Structure for PP&C

The author has developed Corke's `conceptual model', incorporating functions presented by Bennett (125) as well as those described by Buffa and Sarin (35) although the latter reference only discusses job

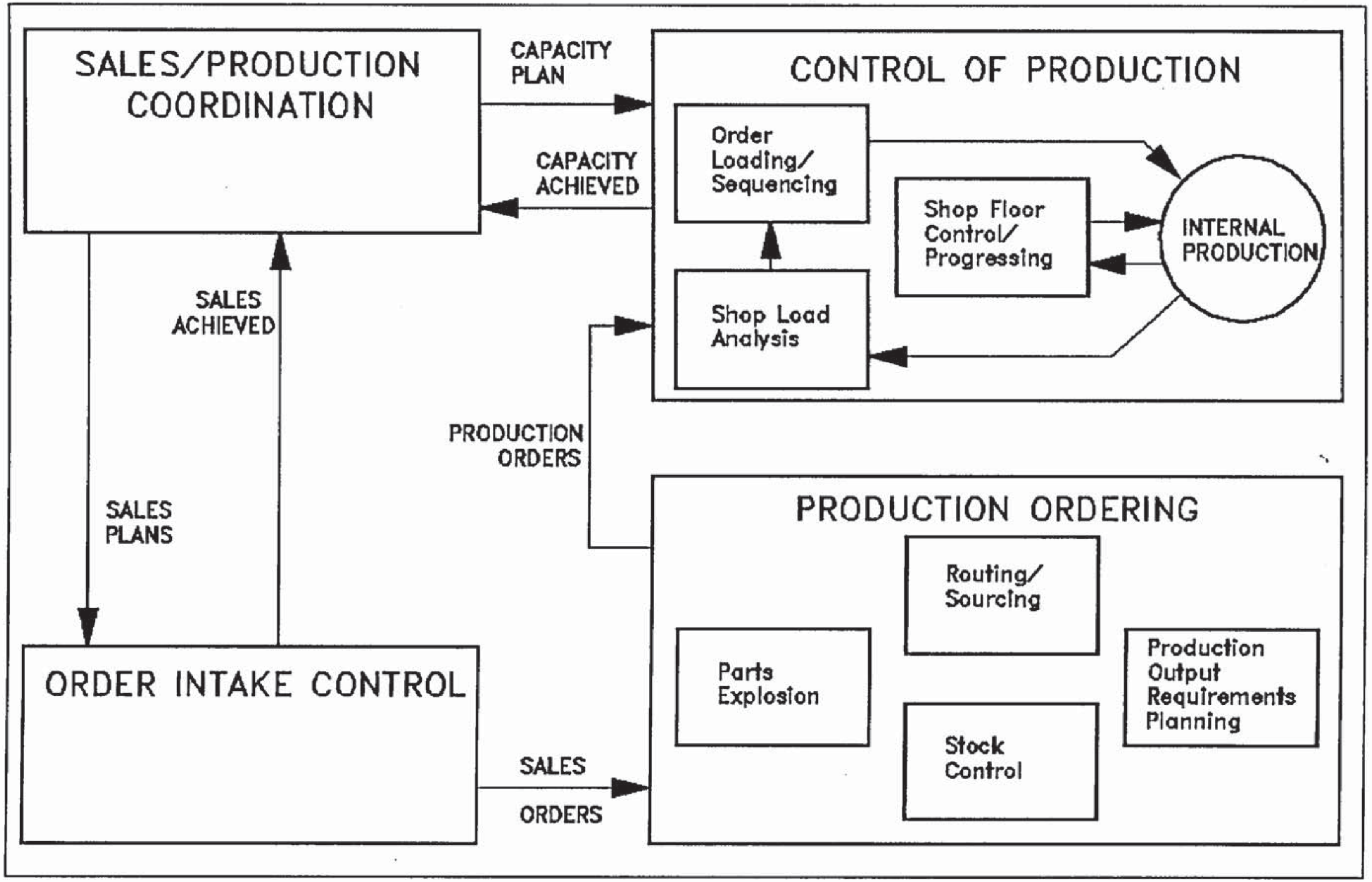


FIGURE 3.2 A CONCEPTUAL MODEL OF THE PP&C FUNCTION.

shop scheduling when describing such functionality. The resultant conceptual model (figure 3.2) demonstrates the profundity of the task faced when trying to structure the PP&C task. As Bennett states, production control systems are unique. The author would describe them as profoundly unique. Not only do the functions within PP&C vary greatly in importance, so does the connectivity between them, as does the extend to which PP&C is decentralized. This conceptual model then demonstrates one aspect of the diversity which the PP&C literature has been trying to capture.

Implicit within any description of PP&C is a structure of the PP&C task. Large numbers of authors present structures of PP&C yet their work does not appear to be cumulative. This suggests that the exact nature of any structure is not seen as greatly important. Certainly the differing views of authors as to the correctness of providing `definitive' structures for PP&C has not appeared as an issue previously.

The author above stated his belief that a discussion of the usefulness of the different PP&C techniques requires an appropriate structure of the PP&C task. The structure developed by the author will be used below for such a purpose. That such a structure is absent from most comparisons perhaps reflects an apparent low level of interest in extensive comparisons of the different techniques.

<u>The Use of Computers within PP&C</u>

Computers have been in use within PP&C since the 1950s and were being harnessed effectively by the mid-1970s according to Corke (126). Despite this assertion and the improvements in computer technology since then, Corke states that PP&C is still not widely successful (14). As discussed above he is not alone in seeing PP&C as widely ineffective. Corke however sees the role of the computer as making the PP&C task easier but he stresses the need to grasp the principles of PP&C. He does not see computers themselves as the answer.

In 1970s Stewart presented four case studies of PP&C computer implementations (127). Two of them had failed. All four cases demonstrated difficulties in implementing computer systems within PP&C. Stewart's conclusions were that computers were not the answer to PP&C. Options other than computer solutions should be considered as should the manufacturing policy. By the mid-seventies, Constable and New (36) saw capacity-free MRP and WIP and stock reporting systems as the limits to computer use within PP&C. Other applications, except in special circumstances, were "best left alone".

These two examples demonstrate the establishment of two issues with respect to the use of computers within PP&C.

The first issue, the advice as to how to overcome implementation difficulties, has generally achieved a consensus. Prabhu and Baker (15) and Rao (128) both state that gaining the support of top management is essential and identify the need to integrate the computer's functionality into the operation of the organization. Other authors echo elements of this view. Plossl sees the principal problem as obtaining the correct form of data (13) while Reinfield warns that implementation decisions are frequently driven by exigency and politics rather than sound judgement (107).

The other area, the understanding as to what is and is not usefully computerizable, has not achieved a consensus. Some authors imply that there is no limit to the computerization (13,15). Norquist (129) sees `number-crunching' and repetitive work as usefully computerizable. Further he states that many of the problems caused by the use of computers in PP&C could have been avoided through good planning and common sense. Van Dierdonck and Miller (29) suggest technical problems can combine with implementation problems, though without comment as to the technical limitations. Wight states that computers can best handle simple repetitive consistent logic. Sophisticated systems lack transparency, preventing people from understanding and thus from using the system well. To Wight, "the magic is in the people" (17) Reinfield sees computers as still causing an ossifying effect (107).

The Implications of PP&C Literature for the Analysis

The PP&C literature concerned with choosing appropriate techniques has been shown to be contradictory and not exhaustive in its coverage. Attempts to structure the PP&C task have been discussed. A structure of PP&C developed in this regard will be used for comparing PP&C techniques below.

Computers have become widespread within PP&C. While there is no agreement as to the effective scope of their use, it is widely agreed that implementation problems, which would be termed within OR as `soft', affect the usefulness of implemented PP&C computer systems.

3.3.4 COMPUTER INTEGRATED MANUFACTURE

Computer Integrated Manufacture (CIM) is not straightforward to define. Boaden and Dale (130) identified ten categories of definitions of which three seem defensible. These are:-

(1) The integration of Computer Aided Design and Computer Aided Manufacture (CAD/CAM),

(2) The wider integration of organizational functions,

(3) The computerization of organizational functions,

emphasis perhaps being placed on different words of the name CIM (131).

Many see the functions of PP&C as being incorporated within CIM (120,132). Also a survey has shown that industry also emphasizes the integration of PP&C functions within the context of CIM (133). However De Meyer has shown that, within Europe at least, this integration of PP&C functions, centering on MRP, is not frequently concerned with CAD/CAM (134). Prabhu and Baker state that developments in CAD/CAM have been completely separate from such systems as PP&C systems (15).

Of the three definitions of CIM above, the first can be quickly dismissed as not relevant to the task in hand.

Concerning the wider integration of organizational functions, the author has found little within CIM literature which goes beyond the integration of MRP covered by the MRP and Manufacturing Resource Planning (MRP2) literature.

Research into the computerization of PP&C has revolved around the scheduling task. Newman states that within scheduling, the goal of CIM is `autonomous scheduling systems' (135); that is, automatic scheduling. Newman sees expert systems as a necessary component in achieving this automation. Much of the work Newman refers to does not consider itself as research into CIM. However it will be reviewed here under this approach as the motivation for the research is overtly automation.

Two problem solving mechanisms have been used within the research to automate scheduling, Constraint Relaxation and what could be termed Distributed Control.

<u>Constraint Relaxation</u>

Constraint relaxation derives from the thesis that not all constraints and preferences can be satisfied within scheduling. So to produce a schedule, the constraints and preferences are `relaxed' until a solution is obtained. Fox (136) has researched into the automatic relaxation of these constraints to produce schedules for a job shop within Westinghouse, part of the Intelligent Management System project (19).

Fox's research has involved the development of a system called ISIS. Through this development process, the problem solving method has been greatly refined beyond the simple application of constraint relaxation. This research is apparently still on-going (136).

Two drawbacks to using constraint relaxation within Order Allocation are firstly, the constraints are provided by functions within the organization with conflicting objectives, a situation evident to the people working in such functions within BSSP. Secondly, the wide range of constraints and their relative importance can change with time; a conclusion made by Fox (136). The author can identify within the allocation task constraints which could suddenly, with no previous precedent, inhibit the efficacy of such a system. The author would suggest that the introduction of new steel-making rules, new products or new sales policies may necessitate major re-development of a system using constraint relaxation.

Distributed Control

Distributed control using multiple actors or agents has been argued as necessary due to the size and complexity of automated scheduling and because it leads directly to hierarchical control of the manufacturing process itself (137). It has otherwise been argued that distributed control allows reactive rather than predictive scheduling (138).

Buchanan et al (139) are carrying out research into such distributed control within the context of Alcan's Kitts Green works.

Three levels of actors maintain a workable schedule through bargaining and negotiating between themselves, rather than relaxing constraints. The main objective is to maintain a workable schedule.

Results from simulated running of the system have yet to be published. Buchanan has stated that highly coupled scheduling problems may not be suitable to the approach, suggesting that steel scheduling is an instance of such problems (140).

Usefulness of the CIM Approach

The research of both Fox and Buchanan et al effectively revolves around automating the scheduling task. It is thus why the author reviews them under the heading CIM. The research is as yet incomplete, with work examining the requirements of the interface between such systems and the rest of the organization as yet not reported on.

Should automation of the allocation function be seen as the approach to take, these approaches would be highly relevant. However, Greenwood's view that CIM is risky needs to be always borne in mind (141).

3.3.5 SUMMARY

The author has sought approaches through which to tackle the allocation problem. To this end the author has reviewed above the literature of Operations Management (OM), Operational Research (OR), Production Planning and Control (PP&C) and Computer Integrated Manufacture (CIM). He has considered approaches to designing the function of PP&C, the structuring of the PP&C task and methods for choosing between PP&C techniques. Much of this work the author did not find useful to the task in hand.

The approaches that have been proposed within the literature are the strategy-driven approach from OM, the model-driven approach and the

Mode 1 soft approach from OR and two approaches to automation from CIM. Within PP&C, the approach can be called technique-driven. All these the author has criticized.

The use of Mode 2 soft OR, specifically SSM, does seem appropriate though in this role SSM is not seen by the author as an approach. Also, the author has developed a model of the task of PP&C; a model of the minimum necessary activities. This model demonstrated the difficulty in structuring the PP&C task and was itself not seen as useful for analysing the PP&C functions of BSSP. However it will be used in the next section to structure the discussion of appropriate techniques.

3.4 PP&C TECHNIQUES

Reviewed here are the techniques associated with the task of PP&C. To structure the review, the techniques are first introduced under the heading of the functions they carry out within PP&C, these functions being taken from the conceptual model of PP&C developed above (figure 3.2). Those techniques applicable to more than one function are covered in a separate section. A final section draws together implementation lessons from the different approaches.

In discussing each technique, the author generally discusses the technique's track record and usefulness in a number contexts depending on the technique's applicability to the task in hand and on the literature coverage. Thus the techniques covered in the following sections are considered in terms of their use generally within PP&C, within the steel industry and within BSSP. The overriding question is, of course, whether the technique is an appropriate means of improving the allocation situation.

3.4.1 TECHNIQUES FOR `CONTROL OF PRODUCTION'

The function `Control of Production' loads orders into the production facility as well as determining the sequence in which the orders will be processed. Techniques developed to produce a detailed schedule of production thus carrying out both order loading and sequencing.

Much work has been carried out attempting to identify the optimal solution to the problem posed by the `Control of Production' activity. In the past, reflection on the usefulness of this work led some authors to see simulation as the technique through which to achieve `control of production' (30,142). As the use of simulation extends far beyond this, it will be reviewed in a later section. The techniques of Constraint Relaxation and `Distributed Control' are essentially techniques for `Control of Production' and were discussed in section 3.3.4 above.

Algorithmic and Enumerative Techniques

Algorithmic solutions to the scheduling task were developed in the fifties but despite a large amount of research have not evolved much since then (88,143). The limit of such solutions is represented by Johnson's algorithm (30). This provides the minimum make-span solution for a problem involving no more than two machines. Special cases do allow solutions to be found for more machines. However algorithmic solutions are widely agreed to be inappropriate to real-world scheduling problems.

Methods of searching for a solution rather than calculating the solution have employed Branch and Bound methods to increase the efficiency of the search but are still too inefficient for real situations (30). Other search methods are also of limited usefulness (142).

Optimized Production Technology

In the eighties, a new method for providing an optimal solution was developed. Originally called Optimized Production Timetable, it is now called Optimized Production Technology (OPT) and has been developed quite markedly since its inception (106). Unusually the method of optimization is hidden within a proprietary software package. As a result it has come in for some criticism (144). However the general method through which the optimal solution is achieved is well known. This revolves around the concept of bottleneck capacity. The method identifies bottleneck capacities then schedules forward and backward from the bottleneck.

OPT reduces the need to obtain accurate data and indeed the need to control closely all processes. OPT also reduces the problem posed by uncertainty (106). By concentrating on bottleneck operations, the amount of accurate data and control required is reduced, as are the effects of uncertainty. However OPT does not provide any counter to the other criticism of optimization; how to cope with different objective functions.

Buchanan stated that the usefulness of OPT within BSSP, where bottlenecks can be widespread and change with time, is small (140). Expanding on this comment, the fluctuating demand for steel and the relatively fixed capacities do result in overloads rather than distinct bottlenecks. Also the use of larger lot sizes, holding stock ahead of bottlenecks and reducing set-up times, strategies used within OPT, are not applicable within BSSP.

It is now generally accepted that sub-optimal PP&C solutions are acceptable. The author would contend that once sub-optimal solutions are acceptable, not only do the objective functions require definition, but the target performances also. Further, as the schedule is not achieving the optimal solution it becomes questionable whether it is scheduling or some other function that is responsible for any shortfall in performance.

The acceptance of sub-optimal solutions thus results in major restatement of the scheduling problem. Any restrictions due to the operation of an optimizing technique then appear to become highly undesirable.

3.4.2 TECHNIQUES FOR `ORDER LOADING'

Within the activity `Control of Production' some techniques allow for the decoupling of the `Order Loading' function and the sequencing function.

One technique which can be immediately dismissed is that of Just-in-Time (JIT). As outlined in chapter 2, the measure of performance for the project was not immediately concerned with allocated arrears and mill stock levels. JIT was thus outside the project remit.

Sequencing Techniques

Sequencing constraints imposed on strip steel production were fully described above in section 2.1.3. The mechanisms used to decouple the allocation task from the mill sequencing task involved the use of Work-In-Progress (WIP) held ahead of the relevant process, a technique also used by U.S. Steel (111). The allocation task then was required only to balance the order load to enable the sequencing task and also to launch `campaigns' for products for which the use of WIP was impractical. Thus the sequencing techniques such as priority rules and batch sizes are not applicable to the project task.

Order Loading

Two basic approaches can be seen within the literature, optimizing techniques and techniques which load the highest ranking orders up to the limits of capacity.

The optimizing techniques involve Mathematical Programming which has wider application within PP&C than just Order Loading. It is thus covered in a later section.

The techniques which revolve around the ranking of individual orders mainly relate to job shops (145-147). These all consider the delivery date as well as effect on the balance of the shop load. O'Grady and Azoza (145) are alone in not considering process leadtimes and thus the routing of production.

No reference has been found to the balancing of order loads in terms of total `route loads' as is practised within BSSP except as part of optimizing techniques where particular routes are equated to particular products. Nor is there comment that the ability to amalgamate individual orders into routes allows the use of optimizing techniques. The author has found no reference to the comparative efficacy of `ranking' and optimizing techniques.

The order loading methods carried out by allocation could perhaps be improved using either of these techniques. However, as discussed in section 2.2 above, it was still questionable whether this was the root of the allocation `problem'.

3.4.3 TECHNIQUES FOR `PRODUCTION ORDERING'

The `Production Ordering' function of course include such constructs as Economic Order Quantity and other stock control methods which are not relevant to the task in hand. The one technique of relevance is Material Requirements Planning.

Material Requirements Planning

Material Requirements Planning (MRP) relies on two sets of information about a production process. These are bills of material (BOM) and the leadtimes of the items in a BOM. Through these constructs, MRP can be used to convert a sales plan or load into the load imposed on production. The BOM and leadtimes are constructs of the `Production Ordering' activity, as is taking account of finished stock. MRP does not extend to the `Control of Production' or `Order Entry Control' activities per se, although MRP can be used by these activities. In other words closed-loop MRP, in which the material requirements or capacity are altered to produce a workable schedule, is part of these other activities. However, the process is unguided; MRP may provide data and structure to the task but balancing the material requirement and capacity is otherwise unaided by MRP (166,148). MRP is principally an infinite capacity scheduler (149) and as such is a `Production Ordering' activity.

Manufacturing Resource Planning (17), known as MRP2, extends MRP so as to consider resources other than capacity. The closure of the `loop' is still not assisted.

The level of detail which MRP utilizes can vary. As such MRP can act within the `Sales/Production Coordination' activity. Plossl (8,13) and Njus (150) both see the main benefit of MRP in its use to plan capacity roughly. Wight also mentions `rough cut' use of MRP. But he sees this as only a preliminary to the real detailed MRP which is used to monitor and thus to reschedule `open' works orders (17); that is, orders already placed. Thus Wight describes the use of MRP within all activities of the conceptual model. Plossl presents a contrary view; MRP should only be used as a `rough cut' planning aid (13).

MRP is a well known and widely used technique generally in manufacturing. Yet despite this and the recognized benefits from using MRP (151), many companies are not achieving them (152) despite the problems of implementing MRP being well known (153). Indeed the requirements to implement MRP are widely agreed (151,152,154).

However, beyond the standard "MRP requires top managerial support" requirement are stronger comments suggesting the `soft' aspect of PP&C affects MRP implementations. Plossl suggests that MRP has to take "pressure" (13). Krupp states that at the core of every working MRP system is a man who refuses to fail (155). Galvin mentions the support of top management for MRP collapsing under pressure (154). Wight strongly states that the top management should not just support the implementation of MRP but should understand the need for it (17). Others talk of the problem of the `informal system' (156,157) and how top management must overcome the resistance from this system (152) or more generally of `human problems' (153) and the need to integrate the human element (158).

The author interprets these comments as identifying the need for a soft approach to analyse the perceived and actual roles of PP&C within an organization. Thus it would be possible to identify the constraints that an implementation would have to overcome or that the system would have to work within.

The detailed use of MRP can suffer from what is called `nervousness' (149,159). The term `nervousness' describes the occurrence of numerous messages about production diverging from the detailed plan in some minor way. Plossl is quite definite in warning against such detailed use of MRP (13).

The equivalent of the BOM and product leadtime are used by the

Production Planning section of BSSP group office. The planning is not detailed. Simply, the logic of leadtimes and the BOM is used in a spreadsheet to ensure that the planned `market tons' of the different products does not overload any mills. Thus `rough cut' MRP was being used without knowledge of the existence of MRP.

3.4.4 TECHNIQUES FOR `ORDER INTAKE CONTROL'

The literature gives little discussion of the `Order Intake Control' function. Proud has stated that generally within industry, 90% of late deliveries are due to errors in order intake control (160); if this is true, the literature is truly out of step with real world problems. Most work considering the function of Order Intake Control identified by the author considers the use of mathematical programming. The area is reviewed by Abad and Sweeney (161). The author considers this work as too theoretical.

Plossl (8) states that there is danger in allowing `Order Intake Control' too much leeway for action when faced with overloads. He describes "tinkering" with leadtimes as "self defeating and deadly". Plossl's solution to overload situations lies outside `Order Intake Control'; to work off the overloads when they occur and to minimize their occurrence through the use of `rough cut' MRP.

Corke (14) provides the fullest description of this function. In most respects, his description of the processes involved are realistic. Corke does stress the difficulty in overcoming an overload situation. However the rationale that an overloaded facility is an inefficient facility is not so convincing in the case of steel production. Also, the concept that production problems are also commercial problems is presented only anecdotally. Corke only intimates that the method for resolving commercial and production differences is rational argument.

The author sees the production/commercial divide as more central to the problem and requiring more than rational argument to be overcome.

While the Annual Operations Plan provided some discipline in `Order Intake Control', the allocation function was required to determine and maintain order leadtimes to prevent overbooking of orders above the plan.

3.4.5 TECHNIQUES FOR `SALES/PRODUCTION COORDINATION'

`Sales/Production Coordination' must result in programmes to alter either capacity or order load. To significantly increase the capacity of steel production requires long lead times. The production capacity of the various strip products achievable by BSSP is strongly affected by major historic decisions rather than current policy (5).

Aggregate planning considers the compatibility of sales and production plans. The use of `rough cut' MRP within this function has been discussed above. Most coverage of aggregate planning per se considers the use of mathematical programming. Such use of mathematical programming has been said to occur seldom (89,90,91). The usefulness of mathematical programming requires more consideration than this however. Because such techniques are usable within other PP&C functions, the usefulness and limitations of mathematical programming will be discussed separately later.

One area within the literature which will be discussed here is the use of different levels of aggregation to facilitate coordinated planning and production control. This approach is commonly called Hierarchical Production Planning.

Hierarchical Production Planning

Hierarchical Production Planning (HPP) is widely associated with mathematical programming and aggregate planning although the author feels the term HPP has never properly been defined. Hax has stated that the higher levels of HPP should be optimizing if possible (162). If the definition of HPP does not include optimization then HPP would simply be the formal use of hierarchical planning.

When HPP and aggregate planning are said not to be in widespread use (89,91), this probably applies to the use of mathematical programming for this purpose. Of concern to the author is coverage of the formal use of hierarchical planning of production.

Hax, Meal and Bitran, widely seen as the pioneers of HPP, have presented a number of accounts of the design process behind HPP.

An early account by Hax in 1976 (162) presents the constraints which are coped with at different levels of the organization. In coping with them, the different levels of organization interact with each other but such activities cannot be combined in a monolithic manner. Hax then argues the basic characteristics of a PP&C system. Most importantly, he states that it is "mandatory to adopt an integrative and hierarchical approach for the overall logistics (or PP&C) system." The hierarchy is due to the difference in time span between strategic, tactical and operational decisions. Hax presented a large area requiring research to assist the design of HPP systems, research described as "exceedingly ambitious". Inputs into the design procedure were seen as the structure of decision-making carried out within the organization, the product structure, the available solutions and the interaction between levels. The interaction between levels included feedback to higher levels.

This account of HPP research is later quoted by Hax and Bitran as being "extensive justification" for the HPP approach (24,163). However the design process then described does not pay full regard to the earlier considerations. In these later papers Hax and Bitran describe the hierarchy as if based solely on product structure. However they state that the hierarchy can be "extended to different numbers of aggregate levels by defining adequate subproblems." Interactions between levels were mentioned but separately. There was no mention of operational structure. In 1984 Meal described the design process as difficult (148). Indeed the author has found no work extending the account of the design considerations made by Hax in 1976.

Work or comment which has appeared without reference to Hax considers aspects of rolling planning horizons (164), the need for feedback to higher level plans (165) and difficulties caused by high level assumptions (166). Such work does seem to draw into question Meal's comment (148) that the HPP hierarchy is "natural" arising from the different leadtimes needed to execute different decisions.

A number of accounts have been given of HPP designs. The industries involved are diverse; electric motors (122), paper production (167), multi-plant multi-product production subject to seasonal demand (168), aluminium smelting (169), interdependent chemical production within ICI (170). All but the first use Linear Programming for at least one level; the first uses Goal Programming at all levels.

Only two of these accounts cover more than the solution method used. Hax and Meal (168) discuss the use of prototyping, the consequences of the data requirements of the system and the effect of the system on the decision-makers. Stephenson (170) discusses the need to implement the system in stages and the need to tailor the system design to the needs of the organization's decision-makers.

In summary, the literature does not cover to the author's satisfaction the definition, the theory, the design process or the implementation issues of HPP. Excepting consideration of optimization methods (which with respect to mathematical programming are considered below) the author is unclear as to how to harness usefully the constructs of HPP.

3.4.6 TECHNIQUES APPLICABLE TO MORE THAN ONE FUNCTION

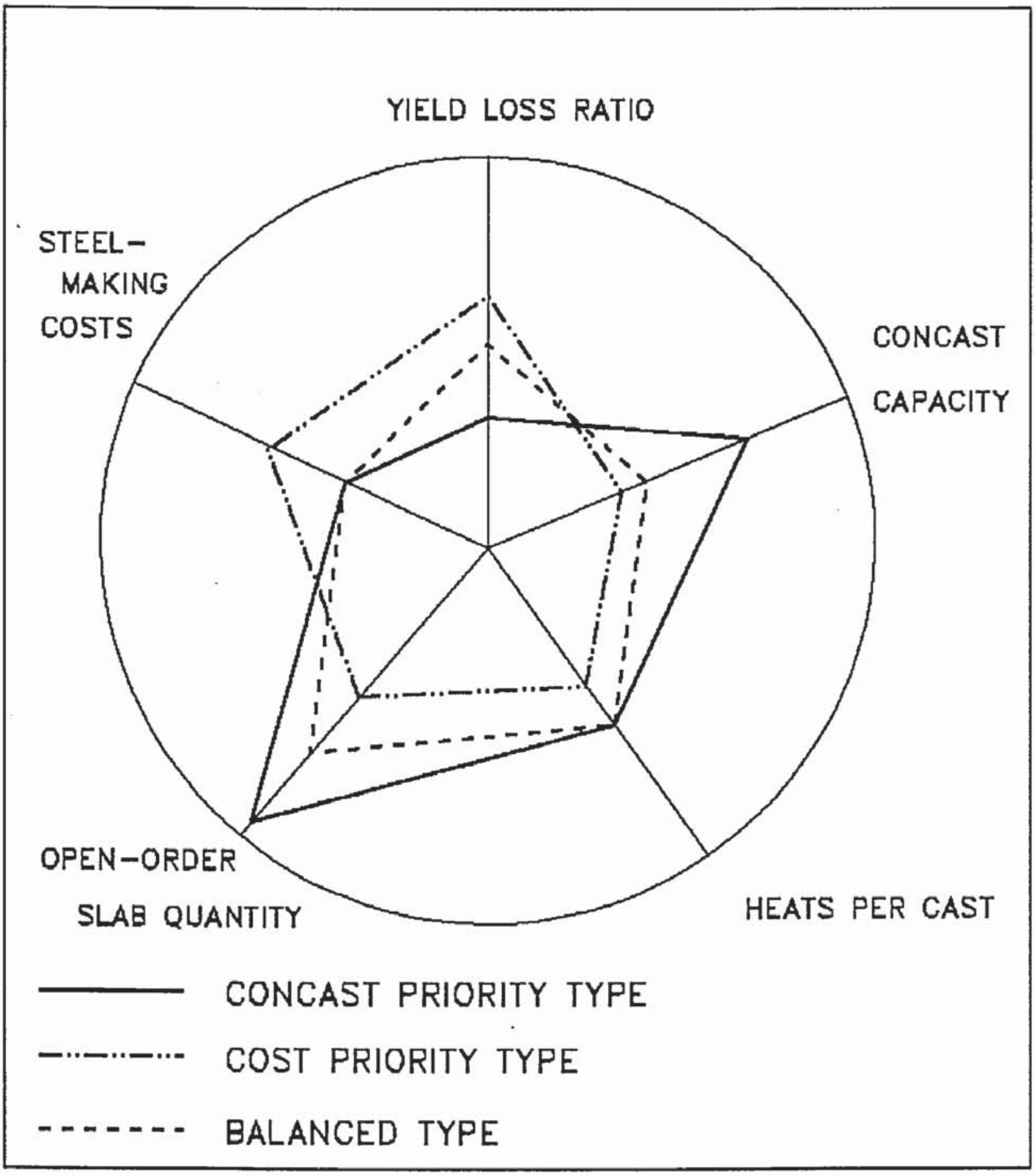
Two techniques applicable to PP&C find widespread application within the author's conceptual model of PP&C. These are Mathematical Programming and Simulation.

Mathematical Programming

Linear Programming (LP) is the most widely used mathematical programming method. Buffa and Dyer (171) state that there have been thousands of useful LP applications, being applied in virtually every organizational function. With regard to the abilities of LP, Starr demonstrates a wide variety of uses (30). However with regard to the actual use of LP, others have found little evidence to support Buffa and Dyer's assertion. Perry and Preston found the picture projected by literature to be that of non-use of LP in production planning outside the oil industry (172). From their survey of 131 United Kingdom manufacturing companies, Oakland and Sohal concluded that LP found very low levels of application (92).

There are a number of drawbacks to using LP within PP&C. The author considers two in particular here. Firstly, to use LP it is required that the orders are grouped into homogeneous sets. This is due to the assumption of divisibility made within LP. Thus LP cannot be used to sequence orders (142). Integer Programming can be used to overcome this problem. However, use of Integer Programming limits the

FIGURE 3.3 THE "RADAR CHART" USED BY OKINAKA TO DISPLAY THE SOLUTIONS FROM MATHEMATICAL PROGRAMMES.



size of the problem being solved (142,145,173).

A second drawback with LP is that the solutions from LP always lie on the boundary of the solution space. That is the equations or constraints are absolute. The idea behind Constraint Relaxation, discussed above in section 3.3.4, is that the inclusion of all constraints in a scheduling problem will result in no solution being possible. Mathematical Programming methods play no part in the research into Constraint Relaxation. A development within mathematical programming is Goal Programming. Linear Goal Programming, as with LP, solves the problem within linear constraints. However, rather than using an objective function which is optimized, Goal Programming considers the deviation from a number of goal functions and minimizes a weighted combination of these deviations. Thus goals can be used to replace absolute constraints. In so doing, Goal Programming overcomes the problem within LP of having to optimize a single objective function (174). Goal Programming has found application in PP&C, for instance by Tsubone and Sugawara (122). One objection made with respect to Goal Programming is the difficulty in determining the goals and weightings to use (172,174). Okinaka, although he does not make clear the exact technique used, appears to use Mixed-Integer Goal Programming to provide schedules for a concaster at Nippon Steel (6). The user was provided with five objective functions. Achievement of these objectives for each solution is displayed using a `radar chart' illustrated in figure 3.3.

A number of authors have commented on the efficacy of Linear Programming, Integer Programming and Goal Programming as well as other Mathematical Programming methods in the context of their applicability to specific PP&C tasks (142,145,172,173,175).

The author would suggest that the usefulness of Mathematical Programming within PP&C should be considered in terms of the role carried out as well as specific tasks. Here the author considers Mathematical Programming as a means of analysing a PP&C function as well as a means of obtaining day-to-day solutions. In the latter case, the difference between providing the solution and the role of a Decision Support System (DSS) is emphasized.

Mathematical Programming techniques have been used as a way of analysing a PP&C function; that is as a mode of enquiry. Examples of such use of Mathematical Programming have been described by Ackoff (176) and by Perry and Preston (172). In both these examples the author interprets the problem definition as rather narrow. It would appear that Mathematical Programming is only applicable to enquiries involving problems far narrower than that faced within BSSP's Order Allocation.

As a means of obtaining day-to-day solutions, the use of Mathematical Programming within PP&C recorded in the literature gives no indication of its use as a provider of solutions. Rather Mathematical Programming appears in the role of a DSS. Further, the use of different types of Mathematical Programming can be seen to have a significant impact on the functioning of such DSS. Tsubone and Sugawara (122) used Goal Programming to allow intangible costs to be included in the model. Stephenson (170) however provided systems using LP such that the user could weigh the effect of intangible costs against the their effect on the objective function by varying the intangible constraints. In this, Stephenson sees the importance of user understanding of the model. Other authors also hold this view (6,173) including Buffa and Sarin (35) who state that the optimal solution should not be seen as the final solution. Optimal solutions should be used, they say, for asking "what if" questions.

The development method used for such systems is relevant to this discussion. Buffa and Dyer (171) in discussing the usefulness of LP, stress the need to involve the user in the development of the model, a process they call `bootstrapping'. It is not made clear if such user involvement is required to facilitate user understanding of the model or to assist the developer in designing the system. The description of such a bootstrapping process presented by Perry and Preston (172) suggests to the author that this approach does not give much leeway in changing the problem definition. Rather, bootstrapping mathematical models appears to concentrate on making realistic the initially perceived required solution.

The usefulness of mathematical programming in providing optimal solutions appears to be in the role of a DSS rather than a provider of solutions. The author was dubious as to the ability of such modelling to cope with changes in problem definition, but did not rule out its usefulness to the problem in hand.

Simulation

Discrete event simulation has long been used within the steel industry. Tocher, a pioneer of simulation, worked within United Steel (177). As to its application within PP&C, BISRA, in 1965, described their simulation work (178); half of this was work used to develop PP&C rules, procedures and systems. At the same conference, Amiry described the use of a simulation to identify the information requirements of a centralized allocation function within United Steel (179).

While most simulation within PP&C remains a development tool, simulation has been used as part of the day-to-day process of PP&C. Dewhurst and Dale describe a simulation used by schedulers rather than

OR practitioners (180). More specific use of simulations include their use in manning level decisions (181), in obtaining accurate delivery dates (182), in order intake control (183), for testing schedules (173) and for identifying bottlenecks (184).

The last two examples are applications within the steel industry. Jain et al (173) describe the use of a simulation in a Bethlehem Steel machine shop used to manufacture the steel rollers used by the company. The simulation was used to test prospective schedules for the machine shop and thus ensure that a good schedule was achieved. Jain et al stated that large savings were associated with its use. There was one problem. The machine shop had no control over the demand for new rollers. In overload situations, use of the simulation was not effective. As a result Linear Programming was used to obtain a rough balance between load and capacity prior to using the simulation.

Narchal (184) describes the structure and uses of a simulation of a single steel works. The language, DYNAMO, prevented dynamic financial inputs into the model but allowed basic financial outputs as well as modelling the performance of each facility within the works. The model was used for such tasks as identifying bottlenecks and analysing of the outcomes from various scenarios and projects.

The advantages of simulation have been stated by Buffa and Sarin as being its suitability for situations with large numbers of complex interactions, the understandability of its solution and the availability of simulation programs (35). Reitman conversely sees three major classes of difficulty (185). The first is keeping the model from becoming too complex. The second is that, while agreeing with Buffa and Sarin that simulation is easy to understand, Reitman sees a need to keep the client involved in the design of the simulation to ensure the correct problem is being solved and that the solution remains credible. Thirdly, Reitman sees problems caused by constraints imposed by the choice of simulation language.

Reitman, in demarcating the limits of simulation applicability suggests to the author that a simulation-driven enquiry would not be advisable. Indeed, the complexity of BSSP is far greater than previous steel production systems modelled using simulation (173,179). Also Bethlehem Steel's experience with simulation of such facilities (173) proved impractical in overload situations.

The author also sees problems with the user interface if simulation were considered as a day-to-day decision making tool within PP&C. The user interface has been one area in which simulation has been criticized. The development of visual interactive simulation (186) overcomes much of this criticism but not all. Robinson (187) described the need for `experienced' users to ensure the user did not invalidate the model using the interactive interface. Also a number of authors have proposed the use of expert systems to achieve such things as providing reasons for a simulated event occurring (188), allowing the complex decisions found in PP&C to be embedded within the simulation (189) and allowing a user access to the entire representation of the problem (190,188). Indeed expert system techniques have been used to build simulations for the purpose of process control (191).

To sum up then, the decision to use simulation as a modelling technique remains open. However, carrying out a simulation-driven enquiry is seen as problematical.

3.4.7 IMPLEMENTATION ISSUES OF PP&C WITHIN THE LITERATURE

Implementation issues within the literature can be found with respect to Material Requirements Planning (MRP), Hierarchical

Production Planning (HPP), mathematical programming and simulation.

Material Requirements Planning

The technique with implementations covered most widely in the literature is MRP. It was mentioned above that the requirements of implementing MRP were well known and widely agreed. These requirements are as follows; the support of top management, correct data, training and educating users, creation of realistic Master Production Schedules. (The last requirement can perhaps be considered as a requirement for correct data.) Even with these requirements known, MRP implementations were disappointing. The evidence within the literature was shown above to suggest that the `informal system' or `soft' problems were preventing effective use of MRP.

A second area of problem was that of MRP nervousness; the occurrence of numerous messages warning of slight deviations from plan.

Other Techniques

Other techniques have implementation issues covered in the literature but less definitive and detailed than MRP.

As with MRP, Hierarchical Production Planning (HPP) imposes a structure onto part of the PP&C function. The structure of HPP is perhaps more flexible than MRP; HPP can adapt to suit the organization. Certainly prototyping the system, suggested by Hax and Meal (168), implies this. Hax and Meal also see the information requirements of HPP as an implementation issue, perhaps matching the `correct data' requirement of MRP. Consideration of the effect on decision-makers (168,170) as well as implementing HPP in parts appear to equate to obtaining support from top management.

The literature covering Mathematical Programming and simulation applies only to single decision-makers. Implementation issues revolve around prototyping being used to ensure the correct problem is being solved and that the solution is understood by the user.

These requirements for success in implementing PP&C techniques bear a marked similarity to those pertaining to PP&C computer systems, as discussed in section 3.3.3 above.

Soft System Methodology (SSM) is not specifically referred to as a means of guiding the implementation of such techniques as, for instance, simulation. It is classically seen as identifying feasible `solutions'. Such a role would tackle some of the implementation issues. However, the use of SSM in a more flexible, Mode 2 manner would in the author's eyes tackle all of these implementation issues.

3.4.8 SUMMARY

The above review has identified many difficulties with the use of the techniques generally viewed as usable within PP&C. The domain-independent techniques of mathematical programming and simulation alone are considered by the author as techniques of possible usefulness within the project task.

A further finding is that all accounts of implementations in the literature report implementation issues. These issues cover the techniques of MRP, HPP, mathematical programming and simulation. The author sees SSM as a means of tackling these issues.

3.5 CONCLUSIONS WITH REGARD TO PROBLEM-SOLVING WITHIN PP&C

The nature of the PP&C task has been described, the author arguing that PP&C can be viewed as a combination of `soft' and `hard' problems. The nature of the allocation `problem' is not seen as in any way unique within PP&C.

Within the literature, approaches to PP&C have been discussed. These included the strategy-driven approach of Operations Management, the model-driven approach of classical Operational Research (OR), the approaches of `soft' OR, approaches within PP&C itself and within Computer Integrated Manufacture. Much of this the author has argued as unhelpful to the task in hand. The exception was the use of Soft System Methodology (SSM), which can be called `soft' OR, specifically in its Mode 2 role as described above.

With regard to techniques used within PP&C the author presented the case that only simulation and mathematical programming, modelling techniques not confined to PP&C, remained appropriate to the task in hand. Within PP&C, although both techniques have been used on a day-to-day basis, their use is predominantly within development work. Also the use of both techniques engendered problems such that their use at this stage in the project could only be seen as a possibly. The implementation of techniques such as these within PP&C can be anticipated to be accompanied by implementation problems. The author proposed that the use of SSM would tackle such problems.

CHAPTER 4
EXPERT SYSTEMS AS A PROBLEM-SOLVING TECHNOLOGY IN PP&C

Chapter 3 has reviewed the approaches currently advocated within to PP&C. This chapter reviews the expert system literature to identify the applicability of expert systems within PP&C.

Expert systems are a new technology. Unlike the approaches and techniques discussed in Chapter 3, it is necessary in the case of expert systems to understand what is meant by the term "expert system". This is covered in Section 4.1, a broad characterization being adopted.

It is also necessary to consider the capabilities of expert systems in general. Section 4.2 discusses problems in identifying these capabilities. Due to these problems, in Section 4.3 the author is only able to present a personal view of these capabilities based on the track record perceivable within the literature. In light of all this, Section 4.4 reviews discussion of the usefulness of expert systems within PP&C, while Section 4.5 considers the methodology used to develop expert systems.

Issues identified within the chapter concerning the usefulness of expert systems and its methodology are summarized within Section 4.6.

4.1 WHAT ARE EXPERT SYSTEMS?

There is no universal agreement on the precise definition of an expert system (192). The author believes that any definition of an expert system would draw criticism. This is not a unique situation for new conceptual types of computer system. The definitions of Computer Integrated Manufacture have been reviewed by Boaden and Dale (130) at some length. Stabell (193), likewise has reviewed definitions of Decision Support Systems, criticizing their proliferation.

However, while Stabell may suggest that an exact definition of DSS is not necessary, it would not appear desirable to discuss the capabilities of expert systems as a problem-solving technology within PP&C without first presenting an understanding of what the author means by an expert system.

4.1.1 EXPERT SYSTEMS, KNOWLEDGE-BASED SYSTEMS AND INTELLIGENT KNOWLEDGE-BASED SYSTEMS

The terms expert system, Knowledge-Based System (KBS) and Intelligent Knowledge-Based System (IKBS) have often been defined. But the existence of three terms to describe similar types of system does appear to confuse the defining of them. Holroyd et al (194), for instance, end their paper with asking the question "What is an expert system?"

Firstly, the term IKBS can be dismissed. The term appears to have been coined by the Alvey directorate. Logically, IKBS are an `intelligent' subset of KBS. But as the term `intelligence' is imprecise (195) there seems to be no advantage in considering such a subset (196).

What is meant by KBS and expert system respectively is not clear cut. Feigenbaum et al (197) state that the distinction is that an expert system contains knowledge from an expert, while a KBS contains knowledge from other sources. This demarcation is not universal; Bobrow et al (198) do not consider expert systems and KBS as mutually exclusive while Addis (199) makes clear that expert systems are a subset of KBS. However, the two terms are more often than not used synonymously (197,200).

The distinction between expert system and KBS has further bearing on the definition of "expert system". In this respect, the term "expert system" will be used throughout this thesis to refer to both expert system and KBS. When it is necessary to refer to a system containing the knowledge of an expert, this will be referred to as an expert system in the 'narrower' sense.

4.1.2 ASPECTS WITHIN THE DEFINITIONS OF EXPERT SYSTEM

Feigenbaum et al (197) defined expert systems as:-

> AI programs that achieve expert-level competence in solving problems by bringing to bear a body of knowledge.

This provides three aspects to the definition of expert system; their origins in Artificial Intelligence (AI), their performance criteria, their use of knowledge. A fourth aspect is the techniques used within expert systems.

Expert Systems as an off-shoot of AI

It is widely understood that expert systems are an off-shoot of AI. This would be helpful in the definition of an expert system except that the definition of AI is also difficult (201,202).

Performance Criteria for Expert Systems

The emphasis on performance criteria within expert system definitions was identified by Hayward as part of the difficulty in providing an adequate definition (203), although Hayward also stated that this was:-

> ..inevitable and proper since clearly the point of expert systems is that they exhibit "expert" behaviour.

Hayward's objection was that such emphasis on performance excluded any definition of the techniques used in expert systems. While such an emphasis has been continued by, for instance, Buchanan (204) and Waterman (205), techniques are not ignored and presumably could have been covered more fully had it been felt necessary. However, the emphasis on "expert" behaviour within the definitions of expert system remains.

Considering what is meant by "expert" behaviour, both Buchanan and Waterman mention only one way to gauge such behaviour; that is against the performance of an expert. Such a situation must strongly suggest that expert systems are seen as replacing experts in some way and may even suggest that expert systems only work in a consultative mode (206).

Including performance criteria of this sort within the definition of an expert system, then, implicitly constrains the methodological debate. What is evident here is that this replacing of experts applies less to KBS than to expert systems in the narrower sense of the term. By this narrower sense, it will be remembered, the author means a system which contains the knowledge of an expert rather than a system capable of "expert" behaviour.

The question yet to be asked here is why should expert systems need to exhibit "expert" behaviour? A computer system after all need only be fit for its intended purpose.

Hayward's reason for describing the expert performance criteria as "inevitable and proper", is that trivial systems, possibly developed using simple expert system tools, will then be termed expert systems. Hayward sees the essence of expert systems to be the non-trivial search. In this he means that either the search technique or the development of the search technique is non-trivial.

This then becomes a definition of expert systems by technique. Such a stance must diverge from the performance criteria on which Hayward insists. Expert behaviour is only expert relative to the abilities of those benefiting from that expertise. In this respect, the author agrees with Hartley (207) and Davis (208); such relative expertise simply equates to a system being fit for its intended purpose. However, the author disagrees with Waterman (205); being applied to a `toy' problem or not being fit for purpose does not mean a system cannot be an expert system.

In the author's view, then, performance criteria only cloud the issue of a definition for an expert system.

The Use of Knowledge by Expert Systems

The use of knowledge (197), symbolic knowledge (204,205), separate domain knowledge (209), explicit knowledge with some flexibility (198) is included in most definitions of expert systems in some form or other and knowledge is explicitly part of a KBS. While even dimensions of knowledge (210) or the importance of knowledge is discussed (211), what is actually meant by the term `knowledge' appears to be mainly taken as understood; the term `knowledge' is not defined in the literature using the term.

Debenham addresses the definition of what is meant by `knowledge' (212). In essence, the author agrees with his description of `knowledge'. That is that knowledge is an emergent property of the search technique used. However, the author finds his hierarchical relationships between objects, information, knowledge and meta-knowledge unconvincing. Meta-knowledge is knowledge used to direct the use of knowledge in a `solution' (197,205). The author cannot agree that such a solution-driven relationship exists between objects and information.

To define knowledge to the author's satisfaction, the common, every-day usage of the word needs to be reconciled with the structure of knowledge-bases commonly used by expert systems.

The definitive meaning of every-day word use can be found from a dictionary. This source (213) gives two relevant meanings. The first meaning defines knowledge as `being known'; "The fact of knowing a thing, state, etc," or "Acquaintance with facts". The second meaning defines knowledge as "The sum of what is known" or more clearly "all that is known or may be known" (214).

With the first of these meanings the question needs to be asked; "What does the knowing?" The content of a clause stored in a knowledge-base may be said to be `known' by a system which could access that knowledge-base. However the author would suggest that it is less superficial to consider the content of a clause as `known' only when accessed by the system.

Given the above then, expert systems obtain what they know through search and seldom explicitly maintain this knowledge except as an integral part of such a search. Thus the abilities of the search are an integral part of knowledge within an expert system.

Illustrating this point, while a human expert will know that he knows, for instance, all the Hot Strip mills in BSSP prior to recalling their names, most current expert systems do not maintain such an understanding of their knowledge. They would only be able to provide such a list by searching their knowledge -base. Most current expert systems do not consider their ability to solve a task a priori, but initiate a search for the solution immediately.

Most current expert systems then only have knowledge of that which has been found through search.

The second meaning summons up the vision of an encyclopaedia or, in the case of a particular domain, a database containing all the information that is useful. Such a database will be far `richer' in content or structure that a normal database. To utilize such `richness', search would have to provide a high level of support in accessing the data. This again points to search mechanisms as being an integral part of knowledge within an expert system.

The search methods coupled with `rich' databases thus satisfies the author's understanding of systems which use knowledge. The inclusion of the search method as an ingredient of knowledge was also concluded by Debenham (212).

The author's understanding of systems which use knowledge needs now to be reconciled with the knowledge-bases commonly used by expert systems. That is, the term `rich' needs defining as do the abilities of the search. It is evident that the search in most expert systems approaches transparency; the intricacies of the search need not be understood by a user. As for `richness', expert systems do not simply sift and sort the content of the database; the database is `rich' enough to allow a search to determine significant amounts of new data.

The author is thus satisfied with the proposition that expert systems use knowledge. However, the defining of knowledge in terms of search and `richness' of data structure suggests that knowledge may possibly be related to the techniques employed in search and to the data structure.

Expert Systems Techniques

Chandrasekaran (215) found it impossible to define an expert system from its use of any single technique. Search, uncertainty, symbolic knowledge structures, rule-bases, separate Inference Engine and Knowledge Base; in all of these he identified significant exceptions or problems. Chandrasekaran concluded that this:-

> points to the multiple dimensions along which expert systems can be viewed and to the need for more careful analysis of much of the terminology that is used in discussing expert systems.

One of these techniques, search, was discussed above as part of the author's understanding of what constitutes knowledge. As the author is proposing that expert systems use knowledge, Chandrasekaran's objections to using search to define expert systems need further examination.

The objection to search as a defining technique was that XCON does not carry out search. This is not totally correct. XCON does carry out search. OPS5, the language in which XCON is written, chooses the most specific but valid rule to fire first (216). OPS5 has to carry out a search to identify this rule. In this XCON still conforms to the author's understanding of knowledge. Chandrasekaran's objection appears then to be that OPS5 does not carry out search by speculatively firing rules. The author does not share Chandrasekaran's objection.

It is worth noting that search carries out the same role that meta-knowledge does, given the definition of meta-knowledge in current literature (197,205). That is, meta-knowledge represents what could be called meta-search. However, meta-knowledge has met with little serious application in expert systems according to Gensereth (217) and was absent from some of the early examples of expert systems (203). Thus meta-knowledge cannot be seen as a definitive form of search.

Chandrasekaran also finds difficulties in using `symbolic knowledge' as a defining technique for expert systems. The author here sees difficulties in the use of the adjectives `symbolic' and `numeric' to describe what he understands as knowledge. Thus he can do no more than agree with Chandrasekaran's comment that the terminology requires careful analysis.

4.1.3 CHARACTERIZATION OF EXPERT SYSTEMS

From the above, the author produces his characterization of expert systems. That is, what the author implies by the use of the term expert system. It ignores the origins of expert system technology within AI as being unhelpful. Performance criteria are likewise not used. The characterization stems from the proposition that expert systems use knowledge and thus from consideration of what constitutes knowledge.

An expert system can be viewed as a system which in the main operates on a knowledge-base. A knowledge-base is defined as a database of complex structure or content, on which can operate a transparent search. This search achieves more than sifting and sorting of the content of the database; the structure and content of the database and the search method is such that significant amounts of new data are created during the search.

This characterization of an expert system is not proposed as a definition. Its purpose is to allow the reader to understand exactly what the author means by the term expert system and appreciate why such a view is held.

Many would see this result as characterizing KBS rather than expert systems. The author thus here restates that by expert system he means also KBS. His reluctance to use performance criteria to characterize expert systems has put less emphasis on the narrower sense of expert system which was described above.

The above characterization includes the ability of expert systems to determine significant amounts of new data. Such an ability to determine new data suggests that expert systems could carry out functions not previously considered as applicable to computerization. The next section will consider how much this perceived usefulness has been realized.

4.2 IDENTIFYING THE CAPABILITIES OF EXPERT SYSTEM TECHNOLOGY

The capabilities of expert systems are clouded by a number of issues. In the author's view, these issues have not been addressed fully enough within the expert system literature despite being contentious for some time.

Problems within the expert system literature make difficult the identification of the "track record" of expert systems.

It is widely accepted that there is a shortage of working expert systems reported. However, different reasons for this shortage are proposed within the literature. The author contends that these differing reasons result from differing underlying assumptions held by those working within the field of expert systems. Such a situation suggests that the track record of expert systems then becomes the only literature source for determining the capabilities of expert systems; a track record which is difficult to identify.

The author contends that only a personal view of the capabilities of expert systems can be presented owing to this situation.

4.2.1 CONTENTIONS WITH REGARD TO THE TRACK RECORD OF EXPERT SYSTEMS

In 1983, Reitman concluded (218) that with "persistence, patience, and an adequate supply of salt" it should be possible to determine the usefulness of AI and thus, the author adds, expert systems. Reitman had identified what is now widely termed expert system `hype'. However his belief that the usefulness of expert systems can be identified that easily shall be shown, seven years on, to be overly optimistic.

In a panel discussion at the AAAI-84 conference (219), the panellists were asked to identify any operationally successful expert systems, as the one system the questioner was familiar with, XCON, was "fast becoming ancient history". In 1986 Buchanan wrote:-

> Depending on which speakers you believe it has been suggested that only one expert system, at most, is `really' working (namely XCON) or that there are hundreds of operational systems (204).

In 1987, Smith wrote:-

> an observer might be forgiven for wondering whether there are any live KBS at all from which their users can claim to be delivering measurable benefits (220).

This graphically illustrates the lack of expert systems universally accepted as `usefully working' and that the situation has been a major issue for a long time within expert system literature. While in the author's view there is certainly more than one system which should be accorded the distinction of being universally accepted as `usefully working', lists of such systems are not appearing in the literature. Lists that do appear contain systems which are not described as `usefully working'. Further the systems mentioned repeatedly within the literature remains small. As an example, of the fifty-nine systems listed by Waterman (205), only six are recognizable within Harmon's list (221).

The author would contend that the list of universally accepted `usefully working' expert systems would appear short to many. Mingers and Adlam, for instance (222), identified only ten systems reported as usefully working within a literature survey covering the years 1984-88. While the survey ignored earlier "traditional expert systems such as MYCIN and XCON" and the literature surveyed has been criticized, the conclusion that the reported track record of expert systems is small was not ostensibly refuted by their critics (223). Given such a situation, the issue still exists.

There are three possibilities which could explain this shortage of expert systems universally accepted as working:-

(i) The literature does contain reports of the `usefully working' systems but due to the `hype' surrounding expert systems, it is difficult to separate fact from fiction.

(ii) `Usefully working' systems are not being reported in the literature.

(iii) The small number of universally accepted `usefully working' systems accurately reflects the real situation.

The first possibility is that usefully working expert systems are hidden in the literature. It has never been proposed that the literature disguises such systems as non-working. However it will be shown that the ambiguity in the literature does prevent any list, other than a very short list, being classed as universally accepted.

4.2.2 DIFFICULTIES IN IDENTIFYING EXPERT SYSTEMS' TRACK RECORD

The accomplishments of expert systems have been shown above to have been a contentious issue for some time. Yet, despite this, the expert system literature lacks rigour in identifying the level of use and the benefits gained from expert systems. In its coverage of the domain of PP&C, the literature will be shown to suffer inconsistencies and poor reporting of system status. The author also sees problems with terminology used within the literature as well as the problems caused by use of the trade press to publish such literature.

Inconsistencies in Expert Systems' Track Record in PP&C

Within the domain of PP&C, several expert systems have been described as working, presumably usefully. This, however cannot be taken at face value.

The most promising candidate within PP&C for a usefully working expert system is ISA, an expert scheduling system developed within DEC which is described by a number of authors as `working' (204,224-226). Yet Kraft, when discussing expert system implementation within DEC (227), describes ISA as "just about to go into production"; Kraft's presentation post-dates all the other references. Further, Kraft also describes IMACS, another expert scheduling system within DEC, as being used as a prototype. McDermott, three years earlier had described IMACS as "now being used in a manufacturing plant" (219).

The author concludes that both ISA and IMACS are probably `usefully working'. This is an assumption though. No report has been found to confirm that they are working and useful. Indeed, the omission of such well known systems from Harmon's list of expert systems in use (221) suggests otherwise.

ISIS is another system described as working by a number of authors (204,228,229). ISIS is a job shop scheduler designed to provide a schedule through `constraint relaxation'. In 1987, Fox, the developer of ISIS, wrote that ISIS had been tested within Westinghouse though not operationally (136); Fox's account post-dates all the references above.

In the case of ISIS, the author contends that ISIS is not `usefully working'. Firstly, ISIS is also absent from Harmon's list (221) though it is perhaps more well known than ISA. Alexander (230), for instance, takes only two examples of expert systems within production management; XCON and ISIS. If ISIS were being used, the author contends that there would be more reason for it to be present in Harmon's list than for ISA. Secondly, the author's personal view is that to harness `constraint relaxation' for scheduling, the main problems to overcome would be organizational. It is evident from Fox's work (136) that very little attention has been paid to this aspect of implementation.

Thus at the time of writing, a universally accepted list of `usefully working' systems has not yet appeared in the domain of PP&C. This is not atypical for expert system domains.

Poor Reporting of Expert System Status within PP&C

Some reviews of expert systems within PP&C make little effort to report the status of systems. Often the status of a system is simply not mentioned. Kusiak and Chen (226), for instance, do not identify the status of fifteen of the twenty-two systems reviewed. Such a complaint can be made about other reviewers (135,224,231).

Accounts of single systems also can omit the status of the system being covered. An example of this is Feigenbaum et al's (197) account of Northrop's ESP system. While they are adamant that the system is useful, never do they state that it is actually being usefully worked. Whatever the status of ESP, such coverage should have made clear its operational status. The sceptical conclusion is that ESP is not usefully at work.

Overcoming the problems posed by literature in which the status is not made clear has been has been attempted. Steffen, in his survey of scheduling systems (225), presents a five-level implementation perspective or framework. Into this he is able to fit fifty-one systems, two of which he categorizes as `operational', those being ISA and DISPATCHER.

Not everyone would agree with the detail of Steffen's categorization, or even that he has presented the most useful categories. However, the author sees such categorization as far more useful than a total lack of comment which is more usually encountered. Development of an expert system from prototype to `usefully working' system appears to be a long process (222). Thus systems which are undergoing live field trials may not be adopted for a long time, if at all, yet they could be described as `working usefully'. A categorization of development stages may prevent ambiguity in the description of development progress. Illustrating this point is the different description of the operational status of two systems, British Steel's multiple burner controller (232) and American Express's Authoriser's Assistant. These two systems have been described while apparently at a similar stage in their development; undergoing live testing. Yet one is described as "now being tested at Trostre Works" (10) while the other is described as though full deployment was as good as achieved (197).

Problems with the Terminology Used in Expert System Literature

The contradictions and ambitious claims that have been made regarding working expert systems in the past, may be due to ambiguity over the meaning of words used.

Kidd and Cooper (233) noted that the word `working' did not necessarily mean `being used' but meant that the system worked as intended. Kusiak and Chen's review (226) provides good examples of the use of the word `application' to mean applied research.

One of the results of such ambiguity is the creation of a new terminology. The adjective `serious' has been used to describe systems designed to tackle real-world problems (195,210). However the word `serious' is also used to suggest that trivial systems, unlike `serious', more complex systems, are not applicable to real-world situations (203,234,235). Another word, `fielded', has been used to describe the status of systems, for instance by Harmon et al (236). Yet a `fielded' system or a system installed for use does not mean that it is necessarily being used, as O'Neill and Morris found (237). How much of a problem this ambiguity in terminology has caused within expert system literature can only be speculated on. The author believes that the ambiguity allows a single piece of literature to be interpreted in many ways, thus simultaneously strengthening the position of different views as to the capabilities of expert systems.

The Use of the Trade Press to Publish Accounts of Expert Systems

Within PP&C, another possible reason for discrepancies in identifying working systems is the quantity of systems being reported in the trade press and company publications rather than in expert system literature. Of the systems classified by Steffen as industrial prototypes (225), 40% were referenced to such publications. The majority of sources from which the author identified what he takes to be `usefully working' expert systems within PP&C are company publications.

While the credibility of such publications may be questioned by some, the author sees the main problem being the difficulty of covering all such literature. Such a situation must result in reviewers producing widely differing lists of working systems.

Criticism of Expert System Literature

The author has presented a number of problems with expert system literature. At time of writing, very little criticism of this sort has appeared in the literature itself. The reasons for the failure of the literature to present the track record of expert systems, then, appear not to have been investigated. Due to the problems above, the author believes that the only way to identify the status of a system with confidence is still, as in 1984 (219), personal involvement.

4.2.3 REASONS FOR THE SHORTAGE OF REPORTED WORKING EXPERT SYSTEMS

It was argued above that problems with the expert system literature made the reported track record of expert systems difficult to identify. Yet even when interpreted liberally, it is widely held that there is a shortage of reports of 'usefully working' systems. Many comments have been made to explain this situation. These comments appear to fall into four categories as follows; successful systems are not being reported, the new technology will take time to be harnessed, there are problems to be overcome within current expert system technology, expert system technology is not as useful as expected.

Successfully Working Expert Systems are not Reported

It was argued by Steels and Campbell (238) that the commercial success of expert systems has caused successfully developed expert systems not to be reported in literature. Secrecy certainly may prevent details of systems being published. Steffen states that expert systems used to supplement MRP systems are unlikely to be reported (225), though he does not suggest this would be due to them being successful. Also Hewitt, when surveying expert systems in North America (239), came up against secrecy about the function of systems under development. But he found that companies would talk about the status of systems under development; he found none that were operational. Schindler (240) describes working systems as "zealously guarded" due to the costs of development which were far higher at that time.

D'Agapeyeff & Hawkins (241) proposed another reason for non-publication revolving around the size of successfully developed expert systems. It has been stated that most usefully working systems are small (242) and not particularly ambitious (243). The developers of such systems may feel such systems do not merit a publication.

Delays in Harnessing Expert System Technology

The lack of literature could be explained by a delay in harnessing expert system technology. Rauch-Hindin (216) and Bramer (192) both suggest this to be the case but give no reason. Others suggest that such a delay is inevitable as with all new technologies (234,244,245). Merry (246), puts the case for the `gestation period' of new types of software to be ten to fifteen years and suggests that such a time scale would be appropriate for expert systems.

Other more specific reasons which have been suggested include the temporary high cost of `serious' expert system development tools (247), the problem of legal responsibility (242) and the shortage of personnel with knowledge engineering experience (204,248).

Deficiencies in Expert System Technology

Deficiencies in expert system technology have been talked of in the literature; deficiencies which would have to be surmounted before expert systems can come into widespread use. Steels (234) and Buchanan (249) see the need for second generation expert systems, while others are more specific as to the failings of expert system technology. Failings have been identified in the user interface (206,250), in the knowledge representation techniques currently employed (251-253) and also the ability to represent procedural schemes (254,255).

Other deficiencies are seen in the development process of the technology itself. It has been commented that much literature describing applications only describes research systems (205,250) or "toy" systems (248) and that the lessons learnt from building such systems are not applicable to real-world applications (256,257).

It should be noted that these views are almost always given as a warning or as a reason to use a new method. That is these views are seldom arrived at through consideration of specific problems with specific applications.

The Limited Usefulness of Expert System Technology

A minority view is that expert system technology has limited applicability. Both Davies (258) and Martins (259) are very unforgiving of expert system `hype'. Both agree that current usefully working expert systems are applied to relatively simple applications. Davies suggests the future of expert systems lie only in training systems, user interfaces, user application builders and rule induction. Martins, while not suggesting that expert systems are totally unuseful, points to the need for "cleverness and substantial experience with complex real-world computer applications" and that only then would expert systems be capable of "prodigious achievement".

Partridge (260) divides Current Expert System Technology (CEST) from AI in general and states that "CEST is not a raging success." He sees CEST as mainly small-scale rulebases and as readily applicable only to "the domain of abstract, technical expertise."

O'Keefe suggests a similar scope for the usefulness of expert systems (261) and, writing from a background in OR, states:-

> Few problems can be solved (or their solution aided) by backward chaining a few rules, consultation and some measure of uncertainty (262).

Implications for Identifying the capabilities of Expert Systems

The four explanations for the shortage of literature suggest a very broad spectrum of views as to the capabilities of expert systems. Yet apart from the sceptical view of expert system technology, and comment that the limits of the technology are not yet identifiable (205,216), there is very little comment concerning these limits. Indeed, there are difficulties the author sees as yet unarticulated in the literature. Both O'Keefe and Partridge broach the capabilities of expert systems in the context of what is being applied to provide a solution. However, also required to be put in context is the problem being thus solved. While there are so few exemplary working expert systems, terms like `problem' and `solution' may hide assumptions. The author contends that such assumptions are profoundly different and that it is they that result in the contradictory conclusions (263) and the "divergent attitudes" noted by Hewitt (239).

4.2.4 THE RANGE OF ASSUMPTIONS SUPPORTING VIEWS OF EXPERT SYSTEM APPLICABILITY

The four categories of reasons for the shortage of reported working expert systems proposed above can be shown to depend on a number of assumptions:-

(i) The capabilities of expert system technology.

(ii) The needs of the real world.

(iii) The significance of any gap between (i) and (ii) for the technology.

Consider the four categories. The first see expert systems working usefully; there is no gap. The second sees the gap as inevitably being closed with time. The third sees the gap as closable but only by developing some vital aspect of expert system technology. Finally, the fourth category sees the gap too wide to be closed. What is thus evident is that the underlying assumptions directly relate to the assumed usefulness of expert systems.

It would be possible to devise a more sophisticated construct of assumptions but it does not appear a worthwhile activity. The author finds that reviewing the literature to even roughly identify assumptions of this sort is in the main highly tenuous. The author will here consider only the issue of the applicabilities of rule bases and identify the difficulties resulting from underlying assumptions.

Assumptions Underlying the Applicabilities of Rulebases

The applicability of rulebases to real world problems has been considered by some authors in the literature. There are those who see such trivial systems as not applicable to the real world; that is only `serious' systems are applicable (195,235) and `serious' systems are non-trivial (203,234). The other extreme, perhaps most famously propounded by D'Agapeyeff (241,264), views rulebases as useful and applicable.

That most working systems are stated to be small rulebases (260) perhaps supports the advocates of rulebases. Conversely, failed applications are not well covered in the literature (265) nor easy to uncover (237). Further, even with six usefully working expert systems, all of them small rulebases, Kawasaki Steel (266) described expert system technology as "yet unproven". However, this perhaps highlights yet another assumption with regard to the applicability of expert systems; the assumed usefulness of expert systems within Kawasaki Steel is very great. Expert systems are expected to significantly influence the profitability of a steel works (266).

Two problems with rulebases are mentioned within the literature. The first of these is the limits of the knowledge representations in such systems. This is overcome by hiding implementation devices within the rulebase (267). Such inadequacies of the representation within rulebases is the usual reason given to support more complex or `serious' systems (237). The second problem is the limits of the explicit nature of the rules in rulebases. Rulebases have been compared to machine code (255) and legal small print (259). That is they are not easy to understand (268) especially if the rule base is large (269,133).

The limits to which working expert systems using rulebases are affected by these two problems is not evident from the literature. The one working system covered to this depth by the literature, XCON, is by no means small, yet has a simple representation structure (270). Despite the system maintenance problems directly relating to both the two problems above, XCON is still in use (271,272); the task carried out by XCON is important enough to warrant the maintenance problems. The applicability of XCON then depends on the its usefulness as well as its capability.

Thus when not related to a specific real world problem, the perceived applicability of rulebases must be dependent on the assumptions made about real world problems.

The literature has paid little attention to this. Bramer (192,273) stated that, to some, the idea of a rulebase comes as something of a revelation, explaining perhaps a naive view of real world problems held by some.

Kluzniak and Szpakowicz (274) propose a number of reasons why PROLOG mechanisms can be seen by some as <u>the</u> solution to problems. These reasons stem from different assumptions regarding such things as conventional software, the difficulties of modelling a problem, the usefulness of search for finding a solution and the power of logic. As an example of this, for instance, Kowalski suggests that logic is a vital ingredient of expert systems as he sees the use of logic inexorably bound to the verification issue (275). Conversely, Bramer (276) sees the vision of logical programming as far from current reality. The author would propose that such assumptions would also affect judgement as to the usefulness of rulebases.

It has been proposed here that even when an issue of expert system applicability is comparatively well covered in the literature, the underlying assumptions make a useful conclusion very tenuous.

The only evidence left on which to base conclusions as to the applicability of expert systems then appears to be the set of systems which are reported as usefully working. Difficulties in interpreting such reports have been discussed above, with the conclusion that the track record of expert systems could not be identified with confidence.

Carrying out enquiries to clarify these reports would have been extremely time consuming and was thus judged beyond the scope of this project. Thus the author believes that only a personal view of the track record of expert systems can be presented.

4.3 A PERSONAL VIEW OF THE TRACK RECORD OF EXPERT SYSTEMS

It has so far been argued that the only evidence on which to base conclusions as to the applicability of expert systems is the set of systems which are reported as usefully working, despite these reports being not straightforward to interpret.

The author's belief is that reports of working expert systems are in the main representative of actual working systems. Also that the literature itself, while not straightforward to interpret, does yield enough information to draw useful conclusions as to the track record of expert system technology. The main worry the author has over such conclusions is that it is possible for them to be overtaken by the use of more sophisticated tools; given the difficulties in interpreting the literature such developments may prove their efficacy over a very short period. The author has however identified no trends which would greatly alter the following coverage.

Within the literature, the reports of usefully working expert systems very often carry out tasks which fit into certain categories. The standard categorization of this sort encountered within the literature, for instance used by Goodall (242) and by Bader et al (196), appear to originate from Waterman et al (277). This earlier coverage was more concerned with methodological considerations and, the author contends, is inappropriate for recording the track record of expert systems. As Goodall stated, these categories are not mutually exclusive and categorization will probably be due to a `dominant characteristic' (242).

The author therefore attempts to fit the reports of working expert systems into his own categories, with little recourse to other categorizations.

4.3.1 FAULT DIAGNOSIS

The author identifies the most common category as fault diagnosis. This category diagnoses faults in `human designed artifacts' (278). Moralee suggests that fault diagnosis is the "easy" application (279). Both Hayward (278) and Efstathiou et al (280) suggest that this is because rule are an apt formalism with which to encode fault diagnosis expertise. Perkins et al (281) make the same connection but identify the similarity between the workings of rule bases and the workings of fault trees. All these comments were made in the context of real world developments of such systems.

Perkins et al reported their fault diagnosis system, SHEARER, as working. Many other fault diagnosis expert systems now appear to be working (11,197,221,222,236,237,240,241,266,282-287). Of all the expert systems believed to be working by the author, a substantial minority of them fall into the fault diagnosis category.

Identifying in more detail the tasks such systems cope with is difficult from most of these accounts. SHEARER (281) uses a rulebase and works in a consultative mode to diagnose faults in a range of coal cutting machines; automatic diagnosis was said to be not possible due to instrumentation difficulties. The complexity of the task was not mentioned other than in terms of the number of rules in the system.

More typical of the coverage is POLAPRES (282), a system used to diagnose faults in a steel rolling process within MMRA of Luxembourg. The system "speeds up the finding of a remedy for faults that have developed"; the author concludes from this that the system works in a consultative mode. Also rule-based, the complexity of the task is again only mentioned in terms of the number of rules contained in the system.

The software languages or tools used by fault diagnosis systems range from simple tools like SAGE to sophisticated tool kits like LOOPS and KEE. Also used are the languages PROLOG, LISP, OPS 5 and FORTRAN. This spread, as Waterman stated (205), may not be due to a variation in language requirements of the development but due to the developers familiarity with and the availability of software tools. Perkins et al, for instance, found simpler tools not as restrictive as they had been led to believe (281). The author however would add that requirements of interfacing with instrumentation and user interfaces may also be affecting choice of software, though there is little evidence to support this.

Most systems in this category appear to be stand alone consultative systems.

4.3.2 PROCESS CONTROL

Alexander (288) in presenting a framework for the application of expert systems to process control, identifies the need for detection or prediction, and diagnosis of process instabilities in manufacturing. As such, process control can be seen as an extension to the fault diagnosis category with no clear cut divide between them. The author would suggest the difference between the two categories is best seen as the real-time concerns of process control; the timing of the corrective action is important to process control. This conforms to Alexander's view of process control not always being automatic; that it `assists' in taking corrective action.

The advantage expert systems bring to process control systems is the ability to model aspects of the process which would be very difficult (289) or impossible (290) using mathematical models. Knight et al (291) see this as particularly relevant to industrial processes where control involves long time intervals. The author sees the ability to model the process as the same advantage as allowing easy computer system updates, an advantage mentioned by others (266,292). All accounts agree that the resultant systems can provide consistent control twenty-four hours a day, enhancing or replacing human experts. Such advantages have been equated to measures of efficiency in running a plant (10).

While less numerous than fault diagnosis systems, process control systems represent a major category of expert systems with a number of systems now reported as usefully working (197) often within large process plants like chemical plant (235), cement production (222), glass annealing (191) and steel mills (266,289,292). An illustrative exception is the use of an expert system developed by Hitachi to control the applying of brakes on trains (197).

The software language used by these systems is mentioned by just over half of the systems mentioned above. One system used a tool tailor-made by FMC (221). The other systems used LISP, PROLOG, C or FORTRAN. This differs from other categories where the tools used tended to be proprietary tools. The author believes that this is due to the instrumentation interface requirements of process control systems.

4.3.3 THE `ELECTRONIC RULEBOOK'

A third category of expert systems being reported as `usefully working' is that of the `electronic rulebook'. A typical application are the loan or capital investment advisors which are in use by, for instance, Kredietbank, Belgium (241), American Express (197) and Texas Instruments (283). Feigenbaum et al (197) identify a similar number of `electronic rulebooks' as they do process control systems. They apply mainly to financial decisions or to decisions dependent on the law, some of the latter being reported as for sale rather than being reported as `usefully working'.

When `electronic rulebooks' are applied to domains other than finance and law, it is only the need to adhere to regulations which identify them as `electronic rulebooks'. Two examples of such systems are a system to determine required fire-fighting equipment developed by Shell International (241) and a system to advise on the handling and labelling of hazardous chemicals listed by Harmon (221). The author thus sees the presence of man-made regulations as differentiating an `electronic rulebook' from other categories of systems.

The advantages of expert systems over other computer solutions in this category is not mentioned; previously computerization was not considered in this role. The main ingredient of success the author assumes to be the explicit format used to encode the rules in the system.

With the exception of American Express's AA, all `electronic rulebooks' appear to be stand alone and consultative. Mainly, the software tools used could be described as tools of `modest' sophistication; tools like `Personal Consultant Plus' and `Xi Plus'.

4.3.4 TWO TYPES OF `CONFIGURER'

The author has proposed the three categories of expert systems above as being areas where expert systems have demonstrated their applicability (293). While the majority of systems reported as working can be thus categorized, a large minority of systems remain uncategorized.

All but a few of these remaining systems are applied to technical design problems. The author uses the term `configurers' for this category. Such systems can configure a design, as Optex does for lenses (197) or XCON does for computers (270), or may select the best from a number of predefined configurations, as Weld Selector which selects the best type of welding electrode (197).

Other expert systems check some aspect of a configured design rather than configure the design itself. CORPS which predicts the design life of a gas pipe due to corrosion appears to work in this way (241). The author combines these two types of expert system into one category because they are integral parts of the same type of real-world tasks.

The limits of the category `configurer', as with the other categories, are not clear cut. The constraints the system copes with are in the main technical. But technology exists in the real world, and so physical constraints are also imposed, implicitly or explicitly. Thus UFEL (241), which predicts firedamp levels in coal mines, could be seen as a `configurer' or a predictive diagnosis system. If the constraints are man-made regulations, both types of `configurer' would become `electronic rulebooks'.

The number of `configurers' working appears to be only slightly fewer than the number of fault diagnosis systems.

As with `electronic rulebooks', configurers were previously not considered for computerization. The functionality of many `configurers' is not evident in the literature. The author would suggest that the majority appear to be consultative and stand alone.

The software tools used are similar in spread to fault diagnosis systems, although a use of C and a greater use of PROLOG increases the proportion of languages as opposed to tools used.

4.3.5 SCHEDULERS AND ROUTERS

Another numerically significant category of working expert systems are schedulers and to a lesser extent routers. Routers can be seen as configuring an activity, rather than configuring things. Two examples, ESP written in KEE, and CDS written in OPS 5, are reported by Feigenbaum as if they were working (197).

Schedulers have also been reported working (197,221,222,236,266, 283,295) and appear as numerically significant as process control systems. Drawing the line between scheduling and process control is difficult. The author would propose that scheduling is concerned with determining the best timings of a number of occurrences, while process control is concerned with maintaining the process. Even then CASS/X (221), FRESH (197), and ERSTRAC-II (222) could be categorized either way, depending on the statement of problem.

Scheduling has been suggested as a useful but difficult application for expert systems because the use of heuristics can be used to reduce the search space in generating the schedule (55,283).

One of the working schedulers was built using a Texas Instrument's tool kit known as MASS, a tool kit developed specifically for the scheduling task. This tool kit hides the expert systems operation from the user who is provided with interfaces to model the plant being scheduled and to input solution heuristics (283). This suggests the modelling ability of expert systems utilized within process control is also important to scheduling. Similar tool kits for PP&C were described by Rauch-Hindin as Knowledge-Based simulation programs (216) although significant evidence of their use is lacking.

Of the software tools used in these expert scheduling systems, half of them are known. Two systems were developed using specially made tools while three others were developed using LISP, C or PASCAL. Two final systems were developed using OPS 5 and ESHELL respectively. The tools used for these systems generally match the sophistication of the tools used by expert process control systems; the preponderance of proprietary tools found in the other categories is not apparent with scheduling systems. However, unlike process control, this cannot be due to the requirements of scheduling systems to interface with instrumentation. This sophistication of tools used perhaps adds weight to the argument that with the scheduling category, the advantage of using expert systems lies in their modelling abilities.

The complexity of the task undertaken by schedulers and routers is not clear. NKK's and IBM Japan's manning schedulers (197) may be worthwhile systems but they work on what can only be described as peripheral scheduling tasks. One of Texas Instruments' scheduling systems is implemented at their Temple works (283) and is said to have reduced inventory significantly. Such a claim is probably true but this would almost always be the outcome if inventory had not been tackled before, expert system or no expert system. Another account of this system (295) states that "traditional techniques" had been tried before. Strangely, the task carried out, as described in this account seems little different from that carried out by traditional MRP systems. No account of this system has mentioned MRP as an alternative approach.

The author believes that given the current coverage of the use of expert systems as schedulers, the comparative efficacy of such systems cannot be accurately determined. Except in the case of the task having previously been done by an expert (266,294) no consideration is given in the literature to how the task was coped with before and what alternative solutions were considered.

4.3.6 OTHER `USEFULLY WORKING' SYSTEMS

A small number of working systems diagnose faults in things other than `human artifacts'. The DHSS Performance Indicator Analyst (241, 296) diagnoses what could be called `faults in human organizations'. Harmon lists other diagnostic systems as in use (221,236). For instance Pulmonary Consultant which diagnoses lung diseases.

Another area of application is as a front-end to other computer systems (258). One example reported as operational is AIGLON, a system developed to front-end a finite element analysis package (297).

Beyond these diagnostic systems, front-ends and those categorized above, the author has identified only one other system reported as working and described well enough to categorise. That system is Dipmeter Advisor (298) reported as working by Feigenbaum et al (197) which interprets oil-well logs.

4.3.7 CONCLUSIONS FROM EXPERT SYSTEMS' TRACK RECORD AND CATEGORIZATION

The author is reasonably confident about the above categorization.

The number of organizations involved with working systems is large enough to negate the problem of "champions" whose presence is required to promote the development of expert systems within the organization (197,239). The main problem stems from there being so few schedulers and routers reported as working; the systems relevant to the domain of PP&C.

It is evident from the above that the majority of fault diagnosis, `configurers' and `electronic rulebooks`, and as a result, the majority of expert systems, are small and stand alone systems. This agrees with the conclusions of others (189,242,243).

The difference between tool types used by process control and scheduling systems and those in other categories partially agrees with the findings of O'Neill and Morris (237). The categories fault diagnosis, `configurer' and `electronic rulebook' fit well the areas described by them as "favour(ing) micro-based systems". The author identified too many non-micro-based systems to be able to make their assertion. This difference may reflect O'Neill and Morris covering the United Kingdom only. The areas of application of the more `advanced' systems are less in agreement. O'Neill and Morris present far more areas than the process control applications identified by the author in the literature. While reasons for this difference can only be speculated on, it should be mentioned that in this part of their analysis O'Neill and Morris do not differentiate between systems being developed and systems being used.

The systems believed to be usefully working mainly confirm O'Keefe's view; expert systems have yet to tackle the broader, messier management problems. Indeed the categories point to the use of expert systems mainly in problems pertaining to human-designed artifacts or constructs, be it designing, repairing, using or controlling them. The borderline case is that of scheduling where the complexity and conflicting objectives make the total scheduling problem both messy and broad. What is not yet determined is how much of this mess expert systems can cope with; the currently working scheduling systems may or may not be avoiding this `mess'.

Hewitt, in 1986 (239) stated that expert systems had only proved themselves in such areas "as fault diagnosis and configuration of equipment." The author's conclusions suggest little has been added to this; if there are many more areas of application there is less than half of Hewitt's five years left to identify them.

4.4 THE USEFULNESS OF EXPERT SYSTEM TECHNOLOGY WITHIN PP&C

The author's view of the track record of expert systems was presented above. It had previously been argued that this was the only source from which the capabilities of expert systems could be determined. This is certainly a more sceptical view than most. Other sources through which expert system usefulness within PP&C could be identified are in the literature covering systems still under development. Also there is literature which discusses the usefulness of expert systems in PP&C.

While the basis of these sources is not known and the conclusions of these authors are thus not seen as reliable, this literature cannot be ignored.

PP&C Expert Systems In Use and Under Development

With respect to the currently recorded track record, this has been shown above to be restricted to a small number of systems carrying out possibly peripheral scheduling or routing tasks of unknown complexity.

The number of systems under development, as discussed above, is large compared to the number of `usefully working' systems. The range of applications is also large, including the integration of previously separate scheduling activities (299), routing orders (300), monitoring stock levels (301) and acting as advisors (302).

The developments covered in chapter 3, ISIS (136) and DAS (139), directly tackle the scheduling problem, but are not seen by the author as being currently more than applied research.

This amount of development work indicates a perceived usefulness of expert systems within PP&C. Reports from industry appear to confirm this (239). However, the uses for which expert systems are being developed are diverse. So, what is not evident yet within the literature are the particular purposes for which expert systems are seen as being useful or the aspects of expert system technology which are seen as providing this usefulness. Nor is it clear how this usefulness is harnessed.

Reasons Given for Using Expert Systems in PP&C

Many authors have stated that PP&C is a good application area for expert systems, both from the view of expert system technology (19,239,303) and from the view of PP&C (120,304,305) It has even been suggested that expert systems are necessary within PP&C (306). Other authors suggest that expert systems are applicable to the scheduling task (55,74,135,231,283,307).

The author identifies the following themes within this coverage.

(i) The failure of OR to be widely used in PP&C within industry is highlighted by some authors. They conclude that expert systems are more capable than OR techniques and so are applicable (120,283). An alternative view sees expert systems as complementing traditional techniques (55,74,270).

With one exception, the integration of expert system techniques with traditional techniques has not been identified as part of the track record of expert systems. As the expert system would be functioning within the constructs of the traditional technique, the author sees a reasonably straightforward specification process for the expert system component. The difficulties may revolve around the interface between the expert system and the conventional system.

(ii) More useful comment is found with respect to the actual failings of traditional techniques. In this there is more of a consensus. The traditional techniques are seen to restrict the solution. The advantages of expert systems are then described as the ability to model the full complexity of the situation (19,55,120,135,185,283) and the ability to adapt the system to cope with the changing problem (19,283,304,308).

These advantages conform to the conclusions from the identified track record of expert systems.

(iii) Other areas of usefulness are seen as the ability to monitor the production process and transfer relevant information to the appropriate part of the organization (19,305) and the ability to cope with multiple objectives (231).

While such systems may be viable, these authors make clear that they are discussing future usefulness rather than current usefulness.

The second of these themes conforms to the argument that the modelling abilities of expert systems would be beneficial within the domain of PP&C. The rider is added that expert systems are thus more adaptable.

The author however feels obliged to emphasize comment made by some of these authors. Newman states with regard to scheduling:-

> The knowledge required for accomplishing the task of scheduling must be complete, well-defined and carefully structured (135).

Phillips et al stated that building expert scheduling systems is more difficult that building configurers which are themselves more difficult than diagnostic systems (283). Finally, Mayer (229) makes more emphatic comment with regard to the development of expert systems within manufacturing generally. Mayer states that such applications are limited by current expert system technology. In particular, the task characteristics are important, which includes requirements for the task to be bounded, frequently performed and of high value to the organization. Mayer only identifies one PP&C task within his "most promising" tasks for expert systems in manufacturing, namely scheduling bottleneck operations.

This advice conforms to the view suggested from the wider track record that expert systems have yet to be shown applicable to broader messier problems. This suggests a possible limit to their adaptability.

Conclusions as to the Usefulness of Expert Systems within PP&C

The conclusion as to the usefulness of expert systems within PP&C, then, is that they should be able to model the complexity encountered within PP&C more easily than more conventional techniques and thus be more adaptable. A possible use may be as a supplement to a conventional technique. However a constraint on their use appears to be that the problem being tackled needs to be well defined.

These conclusions remain tenuous. How best to utilize expert systems within an organization's PP&C function has yet to be identified.

4.5 THE METHODOLOGY OF EXPERT SYSTEM TECHNOLOGY

Mingers and Adlam (222) concluded that only one percent of systems reported in the literature were working usefully. Certainly such systems are elusive within current literature (223). While many of the reported systems will have been research vehicles and not intended for real world use, many systems must have failed to achieve an intended real world use. While the reasons for such failure have been postulated (309), actual failure of expert system developments are little reported. Indeed the failures themselves are elusive (237). Thus the lessons from such failures are less than even anecdotal.

Also, with so few working expert systems reported, few methodological lessons are to be learnt from the successes of expert systems. In other words, current expert system design methodology must be subject to some doubt.

Here the author attempts to identify the scope of expert system methodology so as to be able to judge its applicability within the domain of PP&C.

4.5.1 A SPECTRUM OF METHODOLOGIES

The author has identified within the expert system literature a spectrum of views as to how to go about developing expert systems. There are three main positions held within the literature; exploratory, solution-driven and problem-driven.

Prior to describing these approaches, it should be noted that much literature does not make clear the stance taken with respect to these approaches to expert system development. Also, multiple positions are implied by some.

The Exploratory Approach

This approach lies at one extreme of the spectrum. In this approach, the entire approach revolves around exploring the problem by developing the expert system. Real world use of the expert system is anticipated only; it is expected that "spin-off" applications will result in useful systems and thus justify the development.

Within the literature, a number of authors have offered a set of development stages necessary for expert system implementation (198, 217,236,277,303,310-314). The majority of these see the first stage as identifying the application followed by identifying the required knowledge or expert. This coverage suggests the exploratory approach. There is little consideration of the user or the environment that the system will be working in.

The author sees two assumptions which he would describe as naive within this approach. Firstly, the assumption that, given a problem whose solution would be a likely application for expert systems, an expert or body of knowledge exists capable of solving that problem. Secondly, the assumption that developing an expert system using the knowledge will naturally lead to a useful application. If there is some part of the process overlooked by the author, it certainly is not mentioned by the literature referenced above.

This criticism can be described otherwise as a lack of specification as to what the expert system is being developed to do. This situation appears to be acceptable to some because the view is held that the specification of expert system applications is impossible (315); the exploratory approach certainly follows from this acceptance.

It can be seen that the exploratory approach is the main stream approach within expert system literature. There is little explicit consideration of how to identify the benefits to the organization. Instead, selecting or bounding the problem's scope is seen as an important (205) if not the most important step in expert system development (198,316). As such, it can be seen that defining the problem's scope replaces any requirements specification. Specifying requirements merges with design, forming a single activity (199,315). The technology has been described as "solution-driven" (211,229, 255,317) but more correctly, should be termed exploratory (286,318). Without any form of specification, the process cannot be anything else but exploratory.

The Solution-Driven Approach

The centre of the spectrum consists of overt searching for expert system applications. Examples of this approach examine the needs of the organization to identify the most useful expert system application. But the raison d'être is still to develop expert systems. Alternative methods of helping the organization are not considered.

The author's criticism of this approach is that any analysis of the organization may identify problems which would not be applicable to expert system development. While the most organizationally advantageous expert system application may be identified, it is assumed firstly that problems not applicable to expert system development can be ignored, and secondly that the chosen problem cannot be tackled more usefully some other way.

Examples of the solution-driven approach in the literature are few. Beerel proposes one such approach (211) based on the analysis of Critical Success Factors. These are used to identify "meaningful applications" which are both applicable to expert system development and important to the organization. There is no consideration of alternative solutions, thus the approach can be seen as solution-driven.

Stow et al present another example of the solution-driven approach (317). They advocate the use of Business Analysis to identify applications. Their main concern is the integration of expert system applications with conventional computer systems. As such, they see expert systems carrying out the less structured tasks unsuited to other computer techniques. It is the insistence on this combination which the author takes as solution-driven.

The Problem-Driven Approach

The other extreme is the use of expert systems to tackle the most organizationally advantageous problems if and only if expert systems are deemed the most suitable technology to achieve this.

The wide range of factors affecting choice of approach in analysing an organization, as well as choice of solution technique, have been shown elsewhere (65,319). The central concern the author sees is how to specify an expert system to allow expert systems to be compared to other techniques.

Gaines and Shaw (320) as well as Moll (321) have suggested the use of Checkland's Soft System Methodology (SSM), and Moll also suggests the use of cost-benefit analysis as a means of identifying the value of expert system development. Moll's use of his method included consideration of alternative methods of solution and thus can be considered problem-driven.

Beyond this, little has been written on how to develop a specification. Gudes et al suggest the use of formal terms of reference agreed by users, experts and developers (322). Others suggest an initial informal specification which is formalized gradually during the development (323,324). Kraft describes the use of informal specifications at DEC (227). Kraft makes clear that the specification process is accompanied by assessing the usefulness of the system and a decision to continue the development or not.

The Three Approaches

The main differences between the three approaches will now be summarized by considering how the extreme approaches vary from the central solution-driven approach.

At one end of the spectrum lies the exploratory approach. While the extreme of this approach will give no consideration to organizational requirements at all, the main stream approach does carry out at least cursory consideration of the function of the required system. The telling characteristic of this end of the spectrum is that there is no specification guiding the system development.

At the other end of the spectrum lies the problem-driven approach. Here the telling characteristic is that there is significant consideration of the alternative solution methods available. Such consideration is likely to entail a more detailed specification.

Reasons for Using Different Approaches

The author has criticized the exploratory approach and the solution-driven approach above. Here, complications which may cloud the choice of approach are considered and possible reasons for considering these other approaches are listed.

The idea of expert systems being used to teach novices is neither obscure (205) nor without working examples (241). Beyond this, however, O'Keefe has described instances where the process of development acted as a vehicle to promote learning (223). O'Keefe's examples are evidently exploratory in approach. However the author can see little reason why, even with such outputs, expert system developments could not be problem-driven.

It is often the case that decisions to build computer systems are taken by clients who have no technical input into the development. When this decision specifies the type of system, that is an expert system, the author can see no reason where for ignoring the use of a specification or ignoring alternative solutions. If the client insists on an expert system the author feels he/she should at least be informed if there are alternatives.

There are in the author's view only weak reasons for considering the solution-driven approach. Firstly, if the object of the exercise is to introduce the technology into an organization. Secondly, if the technology is so superior to the alternatives that a comparison would not be necessary.

The author can see stronger reasons for adopting an exploratory approach. Preliminary investigations may be superior if unhindered by a specification. Also the technology, if flexible enough, may be easily adaptable to any required purpose. Further the complexity inherent in expert system development may render specifications impractical. The author's view here is that neither these nor any other reason have yet been adequately demonstrated to prevent a problem-driven approach.

4.5.2 ISSUES AND CONSTRUCTS OF THE MAINSTREAM APPROACH

Almost all literature remains within the mainstream exploratory approach. However deficiencies in the approach have been discussed in the literature. The reliance on prototypes to specify a system has often been noted as a problem, as has choosing appropriate expert system techniques. Other aspects of the mainstream approach consider the process of choosing an application and also reviewed are various expert system categorizations.

The Relationship between Prototype and Specification

The standard view within expert system literature is that the prototype can act as the system's specification. In conventional software, prototyping is a problematic concept (325). The concept of `horizontal' and `vertical' prototypes is widely understood; whether a prototype roughly models the entire system or whether a small part of the system is built in detail. Also accepted is the decision as to whether to develop the prototype into the working system or throw it away once it has served its purpose, though Parbst (326) suggests that this choice concerns not two but a spectrum of alternatives.

When the prototype is used to present the users with an illustration of what the system options are, these two concepts are not seen as adequate by some (325-327). The rapid prototyping used in expert system development goes further than illustrating system options. It includes the design process as part of prototype development.

However the prototyping concept is seldom considered in detail within expert system literature. The need to make the prototyping cycle explicit has been stressed (311,328) as has the need to consider the purpose of the system (298,310,329). Further to this, others see the prototype as an inappropriate specification with domain expertise being easily confused with programming devices (267,272).

Matching Techniques to Tasks and Choice of Software Tools

A dilemma when developing an expert system is the choice of expert system technique to use and thus the choice of software tools.

In 1984, Smith (298) identified an `oscillating focus of attention' during expert system development. At points in the development of Dipmeter Advisor, the effort had to be focused onto the selection or development of new tools as the old tools proved of insufficient power to allow development. Smith saw the cause of this oscillation as the inability to identify the required tools. He stated:-

> As experience with expert systems grows, ..developers will start with a better understanding of the set of tools that will eventually be required to solve the problem.

This experience has not yet appeared in the literature. Work in mapping techniques to tasks has been carried out by Stefik et al (330), Chandrasekaran (255), Reichgelt and van Harmelen (331) and Kline and Doilins (332). However, it is yet to be usable (333). Waterman stated in 1986 that most existing expert systems were built with tools chosen due to familiarity or availability (205). Whether this remains true cannot be determined by the author. The variations in tools used within the fault diagnosis and `configurer' categories described above suggest that it may be true.

<u>Choices of Application</u>

Extensive coverage has been given as to how to identify possible expert system applications. The most developed of these is that given by Prerau (334). The methods used identify attributes which are indicative of expert system applicability. Moll has criticized this scheme because it only rates candidate applications; it does not identify applications (317). The author would add that it only identifies the best candidate and does not show whether an expert system is actually a good solution.

<u>Classifications of Expert Systems</u>

Other common constructs classify expert systems by domain (205), by task (277,335), by role (251) or along some important spectrum like, for instance, the system's contribution to the solution (208,321) or the number of solutions (332). Some of these constructs certainly seem useful for identifying possible expert system applications. However the constructs are little developed for this purpose. Of the references above, Waterman (205) and Davies (208) are considering general expert system applicability and Stefik et al (277) and Kline and Dolins (332) are considering the applicability of expert system techniques.

Unfortunately, the most developed of this work which would perhaps otherwise be useful, that of Clancey's taxonomies of tasks (335), the author sees as being flawed. Clancey misuses the systemic constructs of Vemuri (336); he ignores the effect of the system boundary being arbitrary which would result in differing classifications for the same situation.

4.6 SUMMARY

In attempting to identify the applicability of expert systems within the domain of PP&C, a characterization of expert systems has been presented to allow the reader to understand what the author means by the term `expert system'. The author then described a number of difficulties in identifying the track record of expert systems. These difficulties allowed only a personal view of the track record of expert systems to be presented.

The track record of useful expert systems within PP&C is shown to be small, though many consider PP&C as a useful application area and many expert systems are being developed within this domain. It has also been shown that most opinions see expert systems in PP&C as more suitable for modelling the complexity of PP&C and are possibly more flexible than more conventional techniques. Further it is believed by some that expert systems are applicable within PP&C only where the task is well defined.

From these conclusions, the issue of the appropriate areas of expert system applicability within PP&C remains.

The mainstream approach has been described as exploratory in nature. It has been shown that it gives no guidance as to how to identify useful applications. The mainstream approach also has been criticized with respect to its use of prototypes and its support to the decision regarding what software tools to use.

Given these criticisms, the approach taken to expert system design becomes an issue; in particular, the applicability of the mainstream approach within the domain of PP&C.

CHAPTER 5
A PROBLEM-DRIVEN APPROACH FOR EXPERT SYSTEM DEVELOPMENT WITHIN PP&C

The first section of this chapter discusses the issues identified in chapter 4 as affecting the development of expert systems within the domain of PP&C. This led to the experimental development of an expert system to test whether or not the reasons for an exploratory approach to expert system development could be substantiated.

The experimental development tackled the problem of encoding the rules used to define the leadtimes of orders. Eventually two different systems were developed to carry out this task. The two systems were developed using different software. The conclusions from this experimental development support the case that the exploratory approach is inappropriate within PP&C. Another issue was highlighted of considerable importance; the choice of software required for the development of a suitable human-computer interface.

With the conclusions from chapter 4 and the conclusions from this experiment, it is then argued that a problem-driven approach to expert system development is both possible and preferable to an exploratory approach.

The chapter continues by describing the constructs and methods developed to enable a problem-driven approach. An account is then given of the testing of these constructs and methods within the confines of conceptual design work carried out on expert system "applications" previously identified within BSSP.

A final section draws together the various different threads from the first five chapters of this thesis which are relevant to the use of the approach within the remainder of the thesis.

5.1 FACTORS TO BE CONSIDERED IN DEVELOPING AN EFFECTIVE APPROACH WITHIN PP&C

Chapter 4 reviewed the evidence within the literature to show to what extent expert system technology had been and might be a useful technology within the domain of PP&C. The evidence however proved tenuous.

In this section the different strands of this evidence are drawn together and considered in the context of PP&C. Such consideration adds weight to the argument for using a problem-driven approach. Given this, the reasons for using an exploratory approach are then examined in more detail and the added complications of using a problem-driven approach considered.

5.1.1 CONSIDERATION OF THE APPLICABILITY OF EXPERT SYSTEMS IN PP&C

Small numbers of expert systems were identified in chapter 4 as usefully working within the domain of PP&C. The vast majority of these PP&C systems were categorized as schedulers while the remainder, both routing systems, were categorized as `configurers'. A greater number of usefully working systems in other categories were also identified beyond the domain of PP&C. Further, chapter 4 reviewed the reasons given in the literature for using expert systems in PP&C.

These separate strands from chapter 4 are here drawn together to consider the applicability of expert systems within PP&C.

Consideration of the Reasons for Using Expert Systems

Section 4.4 above reviewed reasons given in the literature for using expert systems within the domain of PP&C. Of these reasons, the two most relevant were firstly that expert systems allowed the complexity of PP&C to be modelled and secondly that such expert systems were adaptable.

All the authors proposing this reason start from one of two positions. The first is that conventional techniques rely on strong assumptions about the modelling of the problem (19,120,283,304,308, 313), and the second that conventional techniques cannot cope with the number of prospective solutions (55,135). Conventional systems thus cannot cope with the complexity or are unable to adapt well enough to fit the problem. The complexity and need for adaptability is then overcome in one of two ways; by using expertise within a computer system (55,120,135,304,308,313) or by using expert system techniques to model the organization (19,283). In effect, these two proposed uses of expert systems respectively can be seen as amplifying the solution capacity and attenuating the problem complexity, thus conforming to the principle of `requisite variety' (41).

The first of these ways of coping with complexity remains the central question. Directly applying to this is Newman's warning (135). Newman, alone amongst these authors describes the real world `mess' of PP&C. In particular, he describes the scheduling task as "not an isolated function" and "dependent on decisions made elsewhere." Despite this, Newman continues then to describe scheduling systems without discussing the "isolation" and the "decisions elsewhere". Newman does not emphasize the point of his warning; that developing an expert scheduling system is no an easy task.

Mayer (229) does emphasize this point, but with regard to applications of expert systems within manufacturing generally. Mayer points to complications in the adaptability of expert systems and serious problems with the exploratory approach of expert system development.

The harnessing of expert systems of significant usefulness within

PP&C, then, may not be a trivial task. If this is so, it suggests that a greater importance needs to be attached to the approach taken to develop such systems.

As to the second way of coping with complexity, the author would suggest that using expert systems to develop a model of the organization would not escape all the problems associated with simulation which essentially attempts a similar task. As discussed in section 3.4.6 above, expert systems have been proposed as a means of reducing the constraints imposed on modelling by conventional simulation languages with respect to the user interface. Indeed Goodman et al (189) consider extending the functionality of simulation rather than improving the functionality of the user interface, the concern of other authors (188,190).

However Goodman et al's work is currently research. It does not yet consider the adaptability of simulation to fit different problems. Thus the author dismisses the idea of modelling BSSP using expert systems for the same reasons as simulation was dismissed in this regard in section 3.4.6; the author sees such a model-driven enquiry as problematical.

Differences between PP&C and other Categories of Expert System

The author contends that the tasks involved in process control and scheduling should be considered separately from the other categories.

It was demonstrated above in section 4.3.7 that the expert system tools used to develop process control and scheduling systems differ markedly from those used for other categories of expert system. Further, `electronic rulebook', `configurer' and fault diagnosis systems can be said to tackle well defined problems with what could be called `simple' solutions. An `electronic rulebook' provides a solution defined by the context of the rules it contains, a `configurer' simply provides a configuration, while a fault diagnosis system identifies the required repairs. Such `simple' solutions may be supplemented by `how', `what if' and `why' questioning but added to the `simplicity' is the point that rules and fault trees are comparatively static.

Such simplicity is not generally present in PP&C or process control systems. Differences between these two categories then bear closer consideration.

Reasons for using expert systems for process control were given in section 4.3.2. These were that expert systems enabled models to be built which were very difficult using mathematical models and that expert system models were easier to update. This is in general agreement with the reasons given with regard to PP&C except that PP&C does not explicitly use mathematical models. However, the author would suggest that the conflict of objectives, the interactions with other organizational functions, and the complexity and uncertainty of PP&C exceed that of process control. As a result problems within PP&C are less stable than those of process control. Profound PP&C problems may change radically in nature or disappear totally. Thus the requirements for adaptability within PP&C would tend to exceed those of process control.

Possible Applicability of Expert Systems to PP&C

There is no reason to believe that expert systems will not find some application in PP&C carrying out well defined tasks. Possible instances of this are the routing systems described in section 4.3.5. Beyond this is the prospect of expert systems tackling the `mess' of PP&C or finding some role within it. To illustrate this prospective use of expert systems, consider the manning schedulers of NKK and IBM

(197). It is unclear whether or not these systems extend their scope to cope with mismatches between required and available manpower, the resolution of which may be subjected to the `mess' which is so often an attribute of PP&C. Such criticism is not confined to these examples; all systems the author considers as in use within PP&C are likewise poorly described. Thus the usefulness of expert systems within PP&C is not shown by the track record.

However, whatever the abilities of these systems, the author would expect that their development would require to be problem-driven to ensure adequate linkage with the rest of the organization and to ensure the problem is not transitory.

Given this expectation, the author now turns to consider the reasons for using the exploratory approach so as to identify if there are any reasons for its use within PP&C. It is, after all, the mainstream approach.

5.1.2 THE CHOICE OF AN EXPLORATORY APPROACH

The exploratory approach to expert system development differs from the problem-driven approach in that it uses rapid prototyping to develop the system, the prototype acting also as a specification. Also the development is embarked upon with little concern for alternative methods of solution.

The author has criticized the exploratory approach above in section 4.5.1 but identified three possible reasons for not considering a problem-driven approach.

The first of these was that expert systems, through carrying out what is effectively a finding-out exercise, would be assisted in this work if not constrained by a specification. The second was that expert systems are so adaptable that they can be converted to suit a required purpose without the guidance of a specification. The third was that the complexity of an expert system is such that no specification would be possible.

This section now considers the validity of each of these possible reasons.

<u>The Limits of the Finding-Out Method</u>

The development of a specification for a system is generally seen within Software Engineering as a finding-out exercise while software design is a process which determines how to realize the specification. When incorporated in this process, prototyping is usually seen as a tool of specification; an aid to the finding-out (337). The rapid prototyping used by the mainstream approach to expert system development combines both specification and design into a single activity (199,315). The author has identified only one reason given directly in the literature for this situation. That is that the efficacy of the system depends on the format of the rules and facts which cannot be determined prior to implementation and testing (315).

Two points are not answered by this argument. Firstly it is not clear what the efficacy is defined against. Secondly, the exploratory approach is effectively carrying out `evolutionary' prototyping. In Software Engineering, the usefulness of such prototyping has been strongly questioned (337). Evolutionary development works under the assumption that unplanned evolutions can be accommodated. It also makes the maintenance of modularity difficult (338).

Given that expert system development involves the use of prototyping, the alternative to `evolutionary' prototyping should be considered; that is throwing away the prototype after it has served its purpose in the specification process.

Decision Support System (DSS) development has been described as "evolutionary" (339,340). However, as Keen stated, expert system development is "very different" from the development process of DSS (341). DSS development uses prototypes to determine user requirements, part of the specification process. Sprague (342) states that such prototypes differ from the normal use of prototypes. The initial system within a DSS development is a "real, live and usable system". Alavi and Napier (343) describe such a development. After first using a conventional throw-away prototype, the DSS development gradually adds functionality to a "first version" of the system. What is clear from this example is that while specification and design are interspersed, they are separate activities. The "first version" is designed having identified user requirements. The design then remains reasonably static.

Typically a DSS is first developed using high level tools. Inefficient operation may result from this. Thus throwing away the prototype is recommended for a DSS but it is seldom done (340). It has been shown that not doing so may result in maintenance problems (344). Also the architecture of DSS is described as "critical" as it is this which provides the flexible design of DSS (345). However, it is the separation of the finding-out and design process within DSS development which appears to allow the critical architecture to be designed.

The use of prototyping within OR as described by Ranyard (49) is very similar to this description of the "evolutionary" development of DSS. Accounts of prototyping within OR, for instance that given by Perry and Preston (172), illustrate that the use of prototyping within OR does not extend beyond the specification process.

The Adaptability of Expert Systems

Descriptions of the development of Dipmeter Advisor (298) and XCON (271) identified a dramatic increase in size of the system during field trials. This final stage of system development was seen as being as big a task as the rest of the development process put together. Smith (298) describes the iteration between developing the knowledge content of the system and developing the tools to contain the knowledge. Similar rebuilding of tools can be detected within the development of XCON (271,346,347), though over a longer timescale than with Dipmeter Advisor.

While such findings will be software dependent, these accounts do add weight to Mayer's contention that expert system adaptability is a fallacy (229). Indeed, within the Dipmeter development, 42% of the effort was expended on the user interface. This suggests that almost half the development effort could have been problem specific, although it must again be stressed that such a finding will be software dependent.

As well as seeing software dependence, the author also sees such conclusions being dependent on the knowledge used and problem tackled; effectively there is dependence on the category or domain of the expert system.

Adaptability at this stage was considered in the following terms. An adaptable system will be able to accept changes in the knowledge-base without redesign of the tools that contain the knowledge. An adaptable system will be easily adapted to new or changing problem situations. Sharpe (348) provides a good illustration of a development where the system was not adaptable to new or changing problem situations; a system of the `electronic rulebook' category.

The Complexity of Expert Systems

While adaptability has been seen as a contentious attribute of expert systems (348,349), the complexity of expert systems has not. Problems due to software causing complexity (255,347) the author sees as a problem of adaptability. The specification, our reason for considering complexity, is independent of software. Beyond all this, consideration of complexity has only arisen as a factor of the problem rather than a factor of the expert system, as described in section 5.1.1. That is the task of PP&C is described as complex, expert systems applied to this task are not.

The author suggested in section 4.5.1 that expert systems may themselves be complex such that their operation cannot be specified. Possible instances of such situations can be seen in the literature where an expert system achieve only a percentage of correct solutions, as is the case with XCON (271).

5.1.3 COMPLICATIONS OF A PROBLEM-DRIVEN APPROACH

The previous section has discussed two reasons for using the exploratory approach. Firstly, expert systems may be so adaptable that developing the operational system is a simpler task than developing a specification first. Secondly, expert systems may have such complex functionality that a specification is difficult to write. A third reason, that expert system development was essentially a finding-out exercise, was considered to be dependent on the other two reasons.

This situation then raises the question - how valid are these as reasons for using the exploratory approach? As the first of these reasons appears to be dependent on the category of expert system developed, the domain of application and the software tools used, the decision to carry out a trial development to test out these reasons for using the exploratory within the domain of PP&C seemed a natural one.

In section 4.5.1 it was suggested that the exploratory approach was inappropriate within PP&C and that a problem-driven approach would be more appropriate. Given this, a trial of the exploratory approach was seen as a possible opportunity for testing the problem-driven approach in some way.

As described in section 4.5.1, a problem-driven approach differs in two ways from the exploratory approach. Firstly it is driven by a specification which links the design work to the organizational requirements. A problem-driven approach would be required to identify the organizational requirements. Secondly a problem-driven approach requires ways of deciding how to tackle different aspects of the problem which includes methods other than expert systems.

Consideration of mixing the trial of the exploratory approach with that of the problem-driven approach however suggested that the trials may interfere with each other. The exploratory approach in benefiting from the initial specification and design of the system of the problem-driven approach may have affected the testing of the adaptability of expert systems during the development. Thus it was decided to disregard all aspects of the problem-driven approach during the trial development of an expert system using the mainstream exploratory approach.

5.2 TRIAL DEVELOPMENT OF AN EXPERT SYSTEM - THE LEADTIME SYSTEM

The trial development was intended generally to test the usefulness of the exploratory approach within the domain of PP&C. In particular, the trial was intended to highlight issues of adaptability of the developed system and to highlight any problems in specifying its final functionality.

A second but by no means insignificant reason for the trial was to gain hands-on experience with different expert system software tools, looking forward to future decisions as to what software to use for more serious developments.

5.2.1 THE LEADTIME SYSTEM

This section describes the trial development. Later sections will concentrate on the conclusions reached from this development.

The Choice of System to Develop

It should be stressed that the choice of system to develop was not driven by organizational needs. Rather the idea was to choose a system which could be developed reasonably easily. The first consideration, then, was to consider the expertise and rules available for such a trial system.

Within allocation, four general areas of expertise or rules were considered; allocation constraints and methods, routing rules, codes and rules for interpreting computer records, leadtime rules.

The first of these, allocation constraints and methods were seen as too transient and, to reflect reality, would have to be elicited from the allocators. These factors, coupled with the size of the system if completed properly even for one product, led this option to be dismissed.

The second area, that of routing rules, was also seen as too large. Not only would the rules include the Product Manual, a 120 page document, but also unwritten extensions to it as well as the custom and practice on each site.

The two remaining areas both appeared suitable; the leadtime rules, in particular the leadtime rules for UK General orders were

FIGURE 5.1 PART OF THE LEADTIME RULES FOR U.K. GENERAL CUSTOMERS SHOWING LEADTIMES FOR BASIC PRODUCT CATAGORIES AND TYPES REQUIRING REFERRAL TO OPERATIONS.

APPLICABLE FROM 9.00 03/03/87

DELIVERY DATES FOR USE IN NEGOTIATING NEW BUSINESS

Based upon the present ORDER-LOAD situation the following Delivery Dates may be quoted for all HOME ORDERS, GENERAL TRADE only.

PRODUCT		CSHR	CSDP	DELIVERY
HOT ROLLED COIL	:DRY	716	718	02-May-87
	PICK	716	719	09-May-87
HOT ROLLED C/L	:DRY	716	719	09-May-87
	PICK	716	719	09-May-87
COLD REDUCED	:COIL	713	717	25-Apr-87
	:C/L	713	718	02-May-87
GALVANIZED	:COIL	713	716	18-Apr-87
	:C/L	713	717	25-Apr-87
	:CORRUGATED	713	718	02-May-87
	:I.Z. COIL & C/L	713	716	18-Apr-87
	:COIL > 2.0 MM	714	717	25-Apr-87
	:C/L > 2.0 MM	714	718	02-May-87
	:ZALUTITE COIL	715	719	09-May-87
	:ZALUTITE C/L	715	720	16-May-87
ELECTROZINC	:C.R.COIL(AGREED	713	718	02-May-87
	:C.R.C/L RATES)	713	719	09-May-87
LACKENBY :	H.R.COIL	713	715	11-Apr-87
DURBAR(NON STD)	:H.R.COIL	713	719	18-Apr-87
TENSILES	:H.R.COIL	716 (R.W.L Telex 12/3/85)		

DATES QUOTED ARE WEEKS ENDING BUT DO NOT APPLY TO RESERVATIONS.

ALL ORDERS FOR THE FOLLOWING MUST BE REFERRED TO O.A.C. PRIOR TO ANY DATES BEING QUOTED OR ACCEPTED.

(1) GALVANIZED Z4,Z5,Z35,Z40 OR Z55 & ALL H2,H3 COATINGS

(2) H.R.PRODUCTS REQUIRING RESHEARING AND RESQUARING.

(3) WIDE ORDERS WITH LLANWERN OR PORT TALBOT PREFERENCE OR INSISTENCE (REF MEMO JCR/SEPT.83 FOR DEFINITION OF WIDE).

(4) SHOTTON INSIST H.R.C.PICKLED ORDERS.

(5) H.R.C.PICKLED ORDERS FOR SAE 1020 & RMS 207.

(6) ALL VACUUM-DEGASSED ORDERS REQUIRING TITANIUM TREATMENT.

FIGURE 5.2 PART OF LEADTIME RULES FOR U.K. GENERAL CUSTOMERS SHOWING ADDITIONAL RULES APPLICABLE TO THE DIFFERENT CATAGORIES OF PRODUCT.

DELIVERY WEEKS FOR STANDARD PRODUCT IF HOT ROLLED IN WEEK 1.

HOT ROLLED COIL (DRY)	2,3,8,9,12,14,17,20	3
HOT ROLLED COIL (PICK.)	2,3,9,14,17,20	4
HOT ROLLED C/L (DRY)	1,2,3,7,8,12,14,17,20	4
HOT ROLLED C/L (PICK.)	1,2,3,7,8,14,15,17,20	4
COLD REDUCED COIL	4,5,6,8,9,11,12,14,17,18,19,20	5
COLD REDUCED C/L	1,4,5,7,8,11,12,14,17,18,19,20	6
GALVANIZED COIL	5,6,8,10,13,14	4
GALVANIZED C/L	5,7,8,10,14	5
GALVANIZED CORRUGATED	8,10,14	6
GALVANIZED I.Z.	7,8,10,13(COIL ONLY),14	4
GALVANIZED COIL >2.0 mm	5,8,10,13,14	4
GALVANIZED C/L >2.0 mm	5,7,8,10,14	5
GALVANIZED ZALUTITE CO.	5,6,8,10,13,14	5
GALVANIZED ZALUTITE C/L	5,7,8,10,14	6
ELECTROZINC C.R. COIL	5,6,8,11,13,14,16,17,21	6
ELECTROZINC C.R. C/L	5,7,8,11,14,16,17	7
LACKENBY H.R. COIL	2,3,8,14,17,22	3
DURBAR H.R. COIL	8,14,22	3

NOTES:

1.For widths less than 710mm–ADD 1 week.
2.For KHR(40/30,43/25,43/35,46/40,50/35,50/45 or 54/35)BHR(43/25 or 54/35),J55 950,JTMP 12 or 13, of foreigh equivilent–ADD 2 weeks.
3.For BHR 50/35,all BS1501,BS4360 specs 50 and above,all BS5045,all API or foreign equivalents–ADD 1 week.
4.For UNANNEALED–DEDUCT 1 week.
5.For FULL or EXTRA–SMOOTH finish – ADD 1 week
6.For SIDE TRIM LAST with GP finish ONLY–ADD 1 week.
7.For RESHEARED AND RESQUARED–ADD 1 week.
8.For EXPORT PACKING–ADD 1 week.
9.For widths less than 710mm–ADD 2 weeks.
10.For Z4,Z5,Z28,Z35,Z40 or Z55(Inc DISCUS)–ADD 1 week.
11.For gauge less than or equal to 0.65mm–ADD 1 week.
12.For CORTEN–ADD 1 week.
13.For widths less than 660mm–ADD 2 weeks.
14.For INDEPENDENT Inspection–ADD 1 week.
15.For Pickled and Oiled with gauge greater than 4.875mm–ADD 1 week.
16.For DIFFERENTIAL Coating,ONESIDE Coating or Coating thickness greater than or equivalent to 5 micron–ADD 1 week.
17.For "TENFORM" & its equivalents–ADD 2 weeks.
18.For VITREOUS ENAMELLING–ADD 3 weeks.
19.For CR(34/20 or 37/23)–ADD 2 weeks.
20.For VACUUM DEGASSED TITANIUM TREATED–ADD 1 week.
21.For OILED COIL–ADD 1 week.
22.For CUT TO LENGTH–ADD 1 week.

OPERATIONS 02–Mar–87

eventually chosen. These rules define the earliest week in which an order yet to be placed by a UK General customer can be delivered to that customer. The choice was mainly due to an external event. Thought was being given to increasing the complexity of these rules and the point was made to the author that expert systems may provide a means of using more complex rules than with the current method.

The drawback to using the leadtime rules in this trial was that they appeared to be atypical of the domain. They were completely defined and written down. Indeed, as mentioned in chapter 2, they could be viewed by customers and sales staff through access to the central computer. The other areas considered for these trials were by no means so well defined. Conversely, the leadtime rules were continually updated and often changed. Indeed, the format of the rules was not fixed either.

The format of the rules which then defined the UK General leadtimes can be seen in Figures 5.1 and 5.2. They consisted of nineteen categories of order for which an earliest hot rolling week and standard product delivery week was given. Added to this, twenty two rules applicable only to certain categories may amend the leadtime to provide the actual earliest delivery week.

Two complications existed. Firstly `reserved tonnes' (production reserved for a particular customer) allowed orders to have shorter leadtimes. The second complication was `wide weeks' and `campaigns'; that is orders of a particular sort which would not be produced every week. The rules did not apply to these orders and the rules included clauses stating that leadtimes for these orders should be obtained by referral to the allocators. Six such clauses can be seen in Figure 5.1. Generally these `referred' orders were seen as also encodable using expert systems.

The task was viewed by the author as straightforward; the basic forty rules should be easily developed into a consultative expert system to provide leadtimes.

Overview of the Prolog Development

Before embarking on the trial development, the decision was taken to use LPA Prolog version 1.2 initially to develop the system. This was because LPA Prolog was immediately available; other software would be used later anyway to allow comparison of software tools.

The rules being already written down, the first step taken was to determine an effective scheme for encoding them. This scheme consisted of a set of rules, effectively questions, with further rules to prevent the first set from firing until they were known to be relevant and not after they were irrelevant or had been already answered. When no rules remained to be fired the system accessed a separate set of rules to determine the leadtime.

With the rules thus encoded, a simple menu function was written to allow a user to answer proffered questions. To provide for input independent of this function, the system was enhanced to allow for unsolicited inputs and equivalent terms to be entered. Such an ability could have resulted in impossible combinations being entered. To overcome this possibility, the knowledge was re-coded, allowing contradictory inputs to be identified and to then allow the user to choose which was wanted. The system was also provided with the ability to identify any previous inputs which would have been made inadmissible by such contradictory inputs and to delete them. The system thus became able to answer `what if' questions; a purely fortuitous outcome.

The system was deficient in a number of ways. In particular, the system would not cope with leadtimes extending past a year end or over holidays. Consideration of this functionality highlighted the requirements of system maintenance and the focus of the development turned to consider this.

The author did not view the editing of Prolog clauses as a viable method of maintaining the system. Both under Prolog and within a word processing package, typographical errors could not be coped with without understanding Prolog's functionality. A maintenance interface was thus considered essential. To this end, a more sophisticated menu system was built to allow access to such functions. However, the next step required a greater control over keyboard inputs than was easily obtainable under LPA Prolog.

For illustration, consider editing a question which the system asks a user. There is a need for the system to be able to edit all aspects of such questions at once, requiring the system to input individual key strokes at any point on a screen displaying all relevant aspects of that question.

Initial work to develop such a function was carried out but the complexities involved in controlling Prolog's backtracking proved very frustrating. The task was seen as larger than the development justified. Given this situation, it was considered advisable to suspend further development of this Prolog system. Consideration was given to other expert system software tools as alternative development vehicles for the entire system.

Overview of the Crystal Development

A number of expert system software tools were considered for the second development of the leadtime system. At the time of this work, in 1987, the availability of such software was rapidly increasing. But rather than consider in depth all such software, a reasonably hasty decision was made to use the expert system shell Crystal version 2.10. Again the grounds for such a decision were that other software could be used later if required. Crystal was, after all, then one of the best selling micro-computer based expert system software packages.

The author's most vivid memory of building the Crystal system was the quantity of typing required. Indeed, so tedious was the task that the system eventually only covered Hot Rolled and Cold Reduced products fully. Despite having less functionality than the Prolog system, the Crystal version took over twice as much disc space. The main reason was the inability of Crystal to allow rules and menus to be declared as variable names rather than constants; analogous to a third generation language without arrays.

The maintenance interface was in some respects acceptable. Crystal provides `input forms' which were used to allow leadtime lengths and the timing of holidays and year ends to be maintained. Also Crystal provides a menu function which was used to combine the consultation and the maintenance within the same Crystal system. This was however possibly a corruption of the menu function's intended use.

Beyond this, maintenance required editing the Crystal rulebase itself. Editing the rulebase was plausible in that, unlike Prolog, editing could be carried out without understanding the functionality of the language. However, there were a number of drawbacks with editing the Crystal rulebase. Firstly the volume of input would be gargantuan and would require very detailed instructions to guide the work. Secondly, the maintainer, not the system, would determine the relevance of new leadtime rules. For instance, rather than providing a new rule covering certain products, the maintainer would have to explicitly identify where the rule is relevant. It is worth noting that this

Crystal system would not therefore gain the distinction of an expert system as characterized above in section 4.1.3.

Work was carried out rewriting the system to make the rules more understandable by providing all maintenance functions within rulebase editing. The author adjudged the rewrite as not successful. The loss of `input forms' did not bring commensurate improvement to other parts of the maintenance task.

5.2.2. IMPLICATIONS OF THE LEADTIME SYSTEM DEVELOPMENT

Development of the leadtime system highlighted the issue of the maintenance interface as well as allowing consideration of the adaptability of expert systems and the aspects of complexity which would affect the task of specifying an expert system's function.

The Maintenance Interface

The issue of maintenance was strongly highlighted by the leadtime system development. The issue revolved around the choice of software tools.

The use of Prolog, which can be described as an Artificial Intelligence language, required the development of a user interface written in Prolog or the use of another system to act as a source of maintenance data. Without it, the syntax of Prolog made amendment of the Prolog rulebase too difficult for someone unable to programme in Prolog. For instance a misplaced bracket during the editing of a relationship could delete the entire relationship.

The use of Crystal, which can be described as a simple expert system shell, did provide usable interfaces to account for some maintenance tasks. Beyond this however, the maintenance of some aspects of the system, including amendments to the leadtime rules, required editing of the Crystal rulebase itself. The use of another system to act as a source of maintenance data would not aid this situation in the same way as Prolog. It appeared to the author that the understandability of the rulebase was expected to ensure that effective maintenance could be achieved through editing the rulebase. This did not seem practical due to two problems.

Firstly, the inclusion of implementation devices within the rulebase reduced its understandability, something mentioned by Chandrasekaran as a problem within expert systems generally (267). Within the XCON development, this has already been associated with the maintenance issue (347). Even the most powerful tools currently available have been found to limit the available problem representation structures (236).. Costello (350) reflecting on his development of a knowledge base for a scheduling system using KEE states that such limitations resulted in a "rather opaque" representation structure. These powerful tools then do not appear to be free from the need to include implementation devices within rulebases. Within KEE, recourse can be made to Lisp functions. However, this does not overcome problems with representation structures or with user interfaces (247). Rather it draws into question the usefulness of the functions provided by KEE.

The second problem the author identified was that an understandable rulebase may not simplify maintenance. The author would argue that simple maintenance requires all relevant data to be gathered together. Thus, in the case of the leadtime system, a question, the possible answers, the uses of the question and things made relevant by the answers really ought to be displayed together, and not have to be located by paging through a rulebase.

It was the original intention to build more versions of the leadtime system using different software packages. However both the

Prolog and the Crystal developments had highlighted the maintenance interface issue which seriously coloured consideration of other software packages.

One package which was seriously considered as a vehicle for a third leadtime system was Xi. Yet deeper investigation of the literature (351,352) showed that Xi would fare no better than Crystal in the maintenance issue. Indeed the author found only one literature reference to the needs of maintenance made in the context of software tools (353), while Buchanan (354) discusses the same issue in 1979 without mention of the tools employed. Currently the author can add only one other relevant reference (355) to this list. That there is so little mentioned within the literature concerning the maintenance interface suggested that it is not seen as an issue generally. It was felt at the time by the author that future software tools would tend not to concentrate on catering for such requirements.

Since this decision, tools still do not appear to consider the difference between development and maintenance, although there are exceptions to this. For instance, task specific packages which require only the problem situation to be entered by the expert thus divide developer from maintainer. An example of this within PP&C is GENESIS developed to carry out industrial planning and scheduling (356).

There is one adjunct to this maintenance issue within PP&C. The need for sophisticated interfaces for maintenance purposes means that, in the absence of appropriate software tools, even a trivial system like the leadtime system requires considerable effort to implement.

Implied Adaptability Inherent in Expert System Design

Consideration of the adaptability of the systems developed reinforced the view expressed in section 5.1.2 above. That is that adaptability revolves around the amount of redesign required to incorporate new functionality, either with or without redesign of the tools containing the knowledge.

Generally, the development of both systems mirrored the development process of Dipmeter described by Smith (298); that is development of the systems functionality was interspersed with development of what Smith called the "tools". Smith states that development of these "tools" requires "redesign".

The author would be more precise than Smith. The system's functionality can be improved by adding to the rule-base and sometimes by appending extra functionality to the "tools". But more often than not, any development of the "tools" required the "redesign" Smith spoke of, usually a profound and difficult task.

This difficulty is probably best illustrated by the number of activities in figure 5.3 which access the same data for different purposes. The range of these purposes may be construed by some as evidence of adaptability. However this does not reflect the adaptability of the system as a whole, which is the real issue.

A significant aspect of the difficulty in "tool redesign" was understanding the coding of the expert system, a problem mentioned by others with respect to Prolog (133,268,274,276,357) and other expert system software (220,229,255,259,358). The "redesign" was often extensive and great difficulty was found trying to maintain modularity of the design within this development. Indeed, "redesign" was sometimes carried out to reintroduce modularity.

Two aspects of Prolog in particular made understanding the coding difficult. Firstly, the author found difficulty in determining the relevant functionality of the current design as the focus of the design work moved around the system. This appeared to be due to the non-

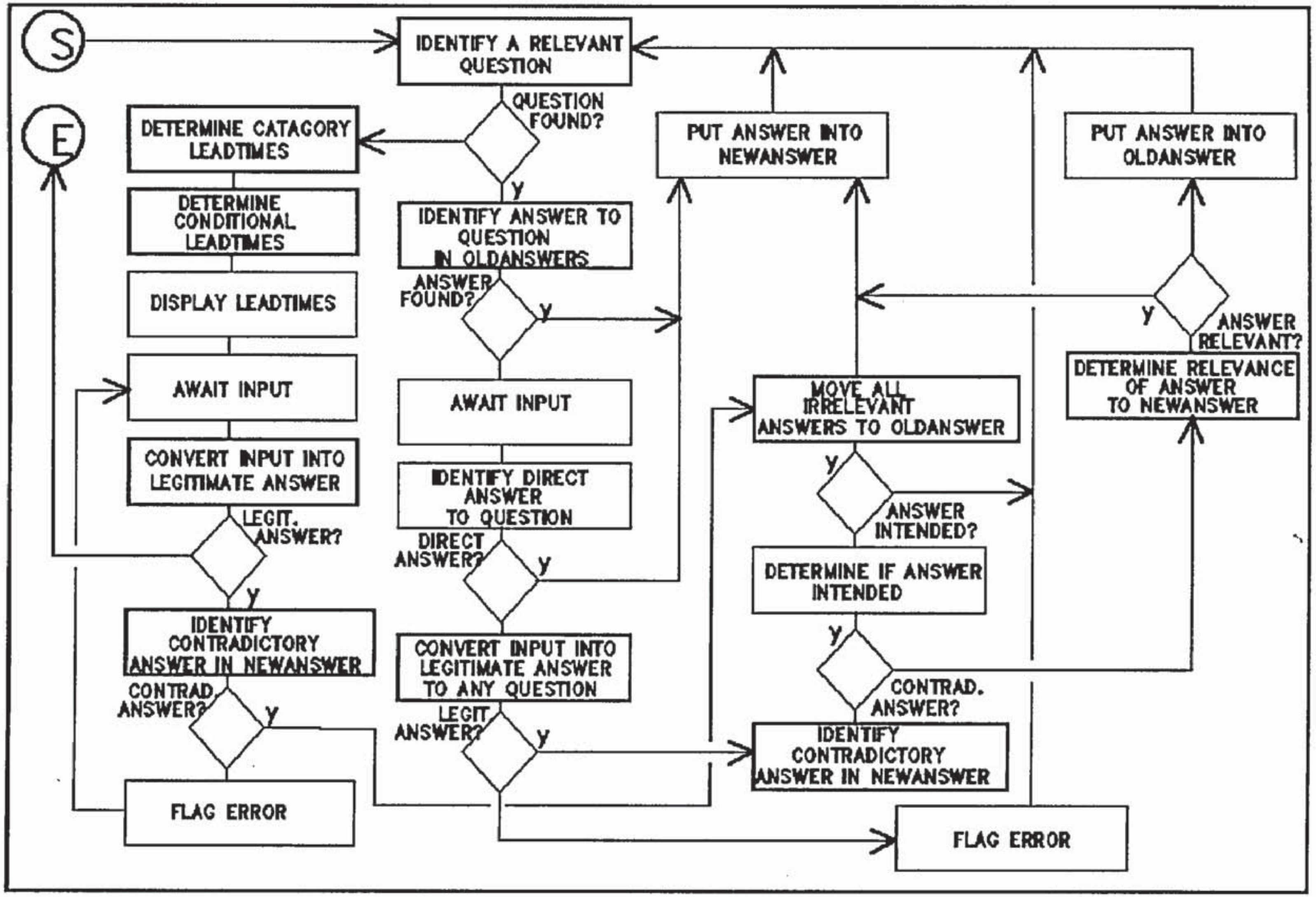


FIGURE 5.3 SPECIFICATION OF THE PROLOG LEADTIME SYSTEM
Activities in bold boxes are knowledge-based and generally access the same data.

procedural nature of Prolog. Understanding the functioning of a particular relationship required understanding of all separate clauses of the relationship as well as the functioning of called relationships. The interdependence between function and called functions appeared far greater than normally encountered in procedural languages.

The second aspect was the complex data structures implicit within the functioning of Prolog. That is the author found understanding variables placed arbitrarily within lists more difficult than understanding named variables and arrays within a procedural language.

The leadtime system development indicated that expert systems were not particularly adaptable and would benefit from a specification to guide the design process.

While such conclusions were dependent on the software used, the author could see reason to speculate that any software which may prove easier to redesign would also have restricted functionality. The Crystal system was a profound example of this. The adaptability of Crystal revolved around the author's ability to use, indeed, abuse the restricted functionality available.

Also the author could see reason for hypothesizing at this stage that the conclusion was relevant generally within the domain of PP&C.

Implied Complexity Inherent in Expert System Design

The antithesis of the proposition that expert systems are too complex to be specified would be a specification developed from an expert system. Figure 5.3 stands as a specification of the functionality of the Prolog system. It was developed from the code after development of the system had been completed and gave a greater understanding of the system than the coding ever had done.

The system of course was a trivial one. The argument may therefore still be valid that non-trivial expert systems would defy specification. However, for this to be correct, the connectivity between activities would need to be greatly increased. The author would expect such complex connectivity only within the operation of a large knowledge-base; the term knowledge-base being used as defined in section 4.1.3. The leadtime development did not provide any evidence in this respect. But what had been provided was a case of an expert PP&C system with significant functionality lying outside the operation of the knowledge-base.

With regard to an expert system providing only a percentage of correct answers and the difficulties for specification thus created; the leadtime system actually relied on identifying an order configuration. Without one, it was unable to provide an answer. Consideration was never given to covering this eventuality. Here is a clear message. PP&C relies on such rules and regulations as the leadtime rules. If they proved inadequate they would be amended accordingly. Indeed, all areas of expertise or rules considered in section 5.2.1 contain either such definitive rules and regulations, defined formally or informally, or factors of custom and practice which could be legitimately represented as definitive rules.

Conclusions from Leadtime System Development

The issues of maintenance and adaptability suggested that even apparently trivial expert PP&C systems will require significant effort to be developed into usable systems. Such a conclusion is of course dependent on available software tools. However the work reported here throws doubt on the ability of the expert system tools available in the immediate future to reduce this required effort.

The adaptability of expert systems has been shown not to be an argument for using an exploratory approach to expert system development. The design work would definitely have benefited from a prior specification, something alien to the exploratory approach.

With regard to the complexity of expert systems, excessive complexity may still exist within large knowledge-bases preventing specification. However it has been shown that an expert system within PP&C can have significant amounts of functionality beyond the operation of knowledge-bases, again suggesting that conventional software specifications can be written for expert PP&C systems. Again it is important to stress that the term knowledge-base is used here as defined in section 4.1.3.

With a specification shown to be both possible and useful, it is concluded that the exploratory approach is neither required nor desirable within PP&C. The size of task in implementing even a trivial system within PP&C and the problems of adapting an expert system to different tasks further justifies the requirement for a problem-driven approach within the domain of PP&C.

5.3 DEVELOPMENT OF A PROBLEM DRIVEN APPROACH

The experience of developing the leadtime system suggests strongly that an exploratory approach is inappropriate to the domain of PP&C and has further strengthened the case for using a problem-driven approach. The development of such a problem-driven approach is considered in this section.

The approach as developed had a number of different inputs. Most important was the leadtime development and further conceptual design work carried out to test the constructs within the proposed approach. Also acting as an input was expert system methodology within the literature, as were the constructs within soft system methodology

(SSM). Other inputs came from the discipline of software engineering.

Development of the problem-driven approach consisted of tackling five requirements. The general reasoning behind these requirements and the constructs harnessed within them are presented first, followed by the relevance of a problem-driven approach within PP&C and similar domains. Next the reasoning behind the fashioning of the individual constructs is presented. Finally the conceptual design work used to test the constructs is described and the use of some of the constructs illustrated.

5.3.1 REQUIREMENTS OF A PROBLEM-DRIVEN APPROACH

A problem-driven approach implicitly has two attributes. Firstly the approach should tackle the predominant problem facing the organization or attempt to identify and tackle such a problem. Secondly the means of tackling (or solving) the problem should not be pre-ordained.

The problem-driven approach to expert system development proposed here is based on five requirements developed from these two attributes. These are:-

(1) An appropriate specification and design format, separate from any enquiry process.

(2) An understanding of the process by which the specification is both developed and used, given that a prototype is no longer the central vehicle of development.

(3) An ability to cope with the softer aspects of the development of an expert system within the domain of PP&C.

(4) An ability to identify possible contributions that an expert system could make within any designed solution.

(5) A means of preventing the choice of expert system software tools from restricting the solution.

Specification and Design

The most evident requirement is that of being able to specify the system's role in tackling the predominant problem. To achieve this, the author proposes the use of Soft Systems Methodology (SSM) to provide activity diagrams of the implementation. SSM has been used to develop information system requirements through the use of `primary task' models and the `maltese cross' (26). The author proposes a more flexible use of SSM. It is evident that expert systems within the domain of PP&C may require extensive `finding out' exercises. This further clouds the boundary between specification and design which is already far from clear (337,359). The specification medium of activity models is thus extended into the area of design through increasing the level of resolution of the model and identifying flows between the activities.

The Use of Prototypes and Non-Functioning Models

The use of SSM as a specification and design medium also drew attention to the uses of prototyping. Initially considered was the enquiry process and whether prototyping was the only vehicle for such finding out. The use of alternative non-functioning models, including non-software models, appeared to have advantages.

The idea that prototyping and non-functioning models could be seen as alternatives grew to include finding out as a supporting function for design as well as for specification. The construct cataloguing the uses of such models was not seen as an aid to deciding what type of model to use and to what end. Rather, the construct would enable the use of such models within a problem-driven approach to be recorded.

Implementation Issues

Another requirement was the consideration of implementation issues within the domain of PP&C. In the main, SSM is well suited to coping with such issues. One area did appear to be deficient resulting in the development of a construct to aid the identification of actors within an expert system development. Such a construct is certainly not alien to SSM. Really it is just a refinement of the more general roles of client, problem-owner, etc, already defined within SSM (1).

Aids to the Conceptual Design of Expert Systems

SSM was then to be used to provide a specification and design medium. The process of design can be described as consideration of alternative possible methods of solution. Design was found to be complicated by consideration of expert systems as possible sub-systems of the total design. While general areas of the design to which expert systems may be applicable could easily be identified, the possible roles such expert systems could play were not. Such roles represented design options; if not identified, possible solutions could be missed. This led to the development of two constructs to assist in identifying the roles which expert systems could play. These constructs will be described in section 5.3.3.

Coping with Expert System Techniques

The need to be unconstrained by software tools was identified within the leadtime system development. This was compounded by the need to develop user and maintenance interfaces. A further constraint was the perceived need to link expert systems to other sub-systems of the implementation. In this a solution was found in using Prolog with a database package, the latter allowing easy development of user interfaces as well as providing a medium for developing conventional software sub-systems.

5.3.2 RELEVANCE OF THE APPROACH TO PP&C AND OTHER DOMAINS

Having now outlined the scope of the proposed problem-driven approach, it may be observed that almost no mention was made of the domain of application, PP&C, except as concerning implementation issues. In this sub-section the rationale behind using a problem-driven approach within PP&C is presented. In doing this, the possible relevance of the approach in other domains will also become evident.

There are two reasons behind the need for a problem-driven approach within PP&C. Firstly the problematical nature of PP&C (described in chapter 3) makes identifying appropriate solutions far from straightforward. This includes not just the complexity, uncertainty and conflicting objectives, but also the need to cope with informal systems and other implementation problems. Given that solutions are not immediately apparent, an exploratory approach would not appear appropriate within PP&C unless expert systems could be shown to be adaptable. The leadtime system development described in section 5.2 suggests strongly that the converse is more correct; expert systems are not currently very adaptable at all.

The second reason revolves around the nature of the knowledge-bases relevant within PP&C; that they undergo change and also that they do not prevent useful specification of the system within which they would work, the latter conflicting with the view currently advocated within mainstream expert system literature.

The personal view of the track record of expert systems in section 4.3 identified more sophisticated tools being used for PP&C systems and process control systems than other types of system. This was despite the major interfacing of the expert system within PP&C being to databases rather than to instrumentation as is the case with process control systems. The leadtime system development identified the maintenance issue. That is the knowledge-bases within PP&C change both their content and their structure, requiring significant maintenance effort or sophisticated tools to contain the possible changes in structure. Both options increase the cost of expert system solutions. As such their application will require more careful consideration than afforded by an exploratory approach.

The leadtime system also highlighted the use of PP&C knowledge-bases as being small enough within the total system to allow specification of that system. Indeed, it was argued in section 5.2.2 that PP&C knowledge-bases are effectively automating explicit rules which otherwise would be used by humans. This appears to allow the specification of the system within which they work. This finding from the leadtime system development is not conclusive. The author is however confident that the problems of PP&C will tend not to be solved by an application consisting in the main of a large knowledge-base whose operation would defy specification.

5.3.3 SATISFACTION OF THE REQUIREMENTS TO ACHIEVE A PROBLEM-DRIVEN APPROACH

The reasons behind considering the five requirements for a problem-driven approach were presented above in section 5.3.1. In this section the design of the constructs and methods used to satisfy the requirements are discussed.

Specification and Design

In section 4.5.1, that part of the expert system literature which could be as considered problem-driven was discussed. Of this literature, most suggest the specification should increase in detail as the development progesses (227,323,324). While use of Soft Systems Methodology (SSM) has been suggested within the specifying process (320,321), other methodological advice to ensure the problem is useful to solve is provided by Kraft. He recommends tackling problems which have to be solved (227).

Another source to be considered is software engineering. In this context, specification techniques can be divided into those based on data flows, those based on data structures and those based on objects. The data flow based approach is widely agreed to be more flexible but is also seen as inferior given requirements of modularity and adaptability (337,360). A data flow based approach was chosen. There were three reasons for this choice. Firstly a flexible approach appeared the more prudent choice when faced with a task of unknown difficulty or content. Secondly the idea of defining data structures to cover the content of a knowledge-base appeared problematical to the author. Finally, the accepted use of SSM as a specification techniques leads to the development of activity diagrams and then to the identification of data flows.

Using SSM as an initial specification medium provides conceptual models; effectively activity diagrams. Wilson (26) has described the use of SSM to identify information requirements by developing a `primary task' model. Wood-Harper et al (361) also use SSM within their Multi-View Approach for defining information requirements. While not using a formal `primary task' model, Wood-Harper et al check the model through analysis of the information flows defined. This does suggest strongly the abilities of SSM to provide adequate activity diagrams without developing a formal `primary task' model.

The methods of SSM (1) appear to bear much similarity to the

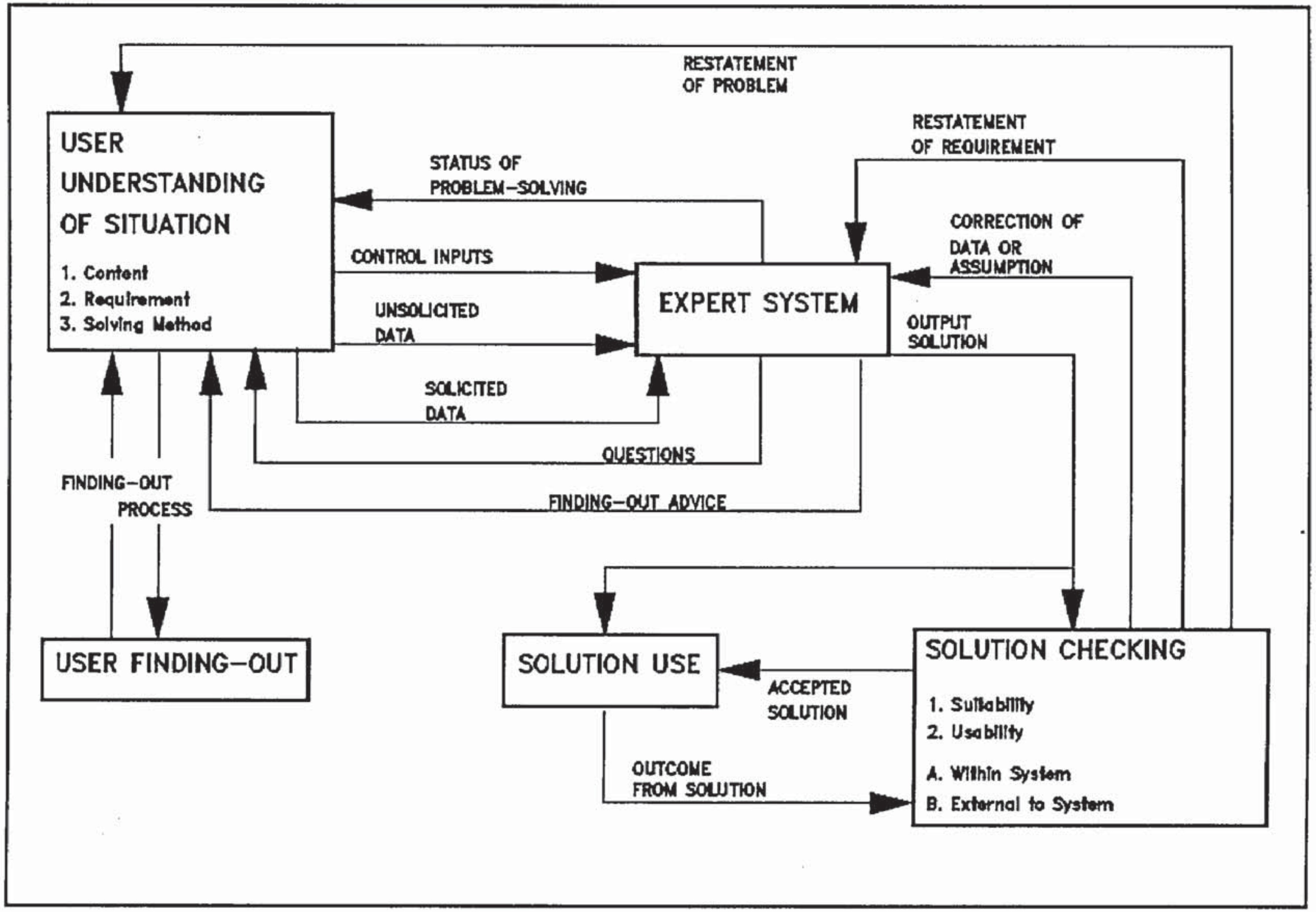


FIGURE 5.4 POSSIBLE EXTERNAL INTERACTIONS OF AN EXPERT SYSTEM WHICH DEFINE THE ROLE OF THE EXPERT SYSTEM WITHIN A 'SOLUTION'.

methods of SADT (362-365). SADT (Structural Analysis and Design Technique) is widely used for software requirements analysis (337), although much of it is not public knowledge (365). SADT and SSM do not appear to have borne comparison previously (366). The similarity adds weight to the choice of SSM as a method of determining software requirements.

The trial conceptual design work (see section 5.3.4 below) carried out to test the constructs of the problem-driven approach confirmed the need for flexibility in the use of SSM. Development of a `primary task' model was found to be secondary to identifying feasible and desirable change to improve the problem situation, the classical use of SSM (1).

Aids to the Conceptual Design of Expert Systems

Identifying possible roles of expert systems within a design was found difficult during the conceptual design work. Aids to identifying expert system applications exist within the literature; however it was argued in section 4.5.2 that they were of little help within a problem-driven approach. Moll used a spectrum of expert system roles within his problem-driven approach based on the level of involvement of the expert system in the decision process (321). The diagram presented in figure 5.4 expands on Moll's work to give a diagramatic representation of possible interactions between an expert system and its `wider system'. The diagram, when mapped onto both the system and its `wider system', allows the analysis of the different roles which could be carried out by the system.

The diagram proved most useful in identifying the possible roles available within expert system design. With respect to the inputs into an expert system, figure 5.4 catalogues ways in which a user, or an external system, can control the operations of an expert system. This

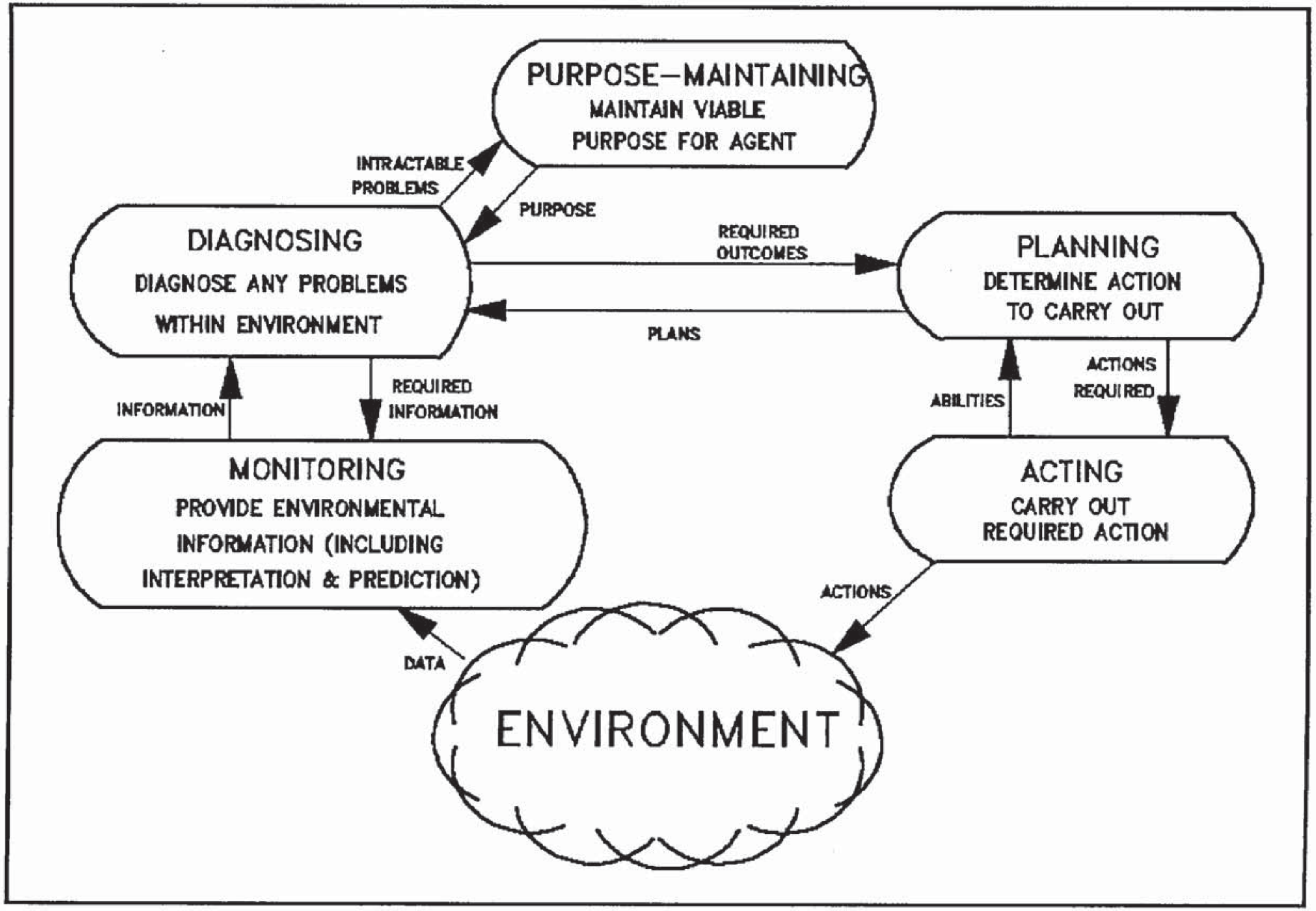


FIGURE 5.5 THE "INTELLIGENT AGENT" WHICH ILLUSTRATES POSSIBLE TASKS FOR EXPERT SYSTEMS AND THEIR INTERACTIONS.

includes the most basic control input, initiating the operation of the expert system. Also the "user understanding of the situation" shows the different areas in which the expert system could increase its role.

With respect to the outputs, the methods by which the expert system can increase its role by checking or using the `solution' are also catalogued.

These variations in expert system design were termed "roles". The diagram appears to the author to be exhaustive except where the benefit from the system is not an output but an `emergent property' of its use. Tutoring systems presented in Basden's list of system categories (251) are of this sort.

A second construct to aid in the identification of design options is presented in figure 5.5. This is an annotated conceptual model and was named by the author the "Intelligent Agent". During the conceptual design work, SSM was harnessed to aid in identifying possible expert system design tasks. One root definition used, for instance, was as follows:-

> A SYSTEM WHICH ACTS WITHIN ITS ABILITIES TO
> MAINTAIN A PURPOSE WITHIN AN ENVIRONMENT.

Very similar root definitions were used on a number of other occasions, suggesting the general usefulness of the model. In addition, the model was seen to effectively model the interactions between the possible tasks of expert systems as presented by Clancey in the form of a taxonomy (335) which were criticized in section 4.5.2 above. Given these findings, the model was annotated and used in subsequent conceptual designs. Such use of conceptual models the author has termed `semi-precious' to indicate that although such models may be reused, the origin of the model should not ignored; its form should not become precious. The possibly contentious nature of `semi-precious' models within SSM must be stressed (367).

The Use of Prototypes and Non-Functioning Models

By turning from the exploratory approach, prototyping is no longer the main development vehicle. Initially, the new role of prototyping was seen as a finding-out vehicle; a "throwaway" prototype. However, initial conceptual designs carried out to test the problem-driven approach suggested that this did not define the role of prototypes totally. It also suggested that prototypes should not be considered alone. Other models of the designed system could carry out the same roles as a prototype. Such models included the use of conceptual models and activity diagrams derived from SSM.

The question then becomes: what roles would such models be used for? Indeed, what role does the specification actually play?

Software engineering offers a number of concepts with regard to prototypes which are widely known; horizontal and vertical prototypes, evolutionary and `throw-away' prototypes. Yet for some time it has been noted that there has been little agreement as to what is meant by the term `prototype' (368). Pressman (337) identifies three roles for prototypes but does not give these roles names. Floyd (369) does provide names:-

(i) Exploratory prototypes to focus the understanding of developer and user during initial requirements analysis.

(ii) Experimental prototypes in which prospective solutions are tested experimentally.

(iii) Evolutionary prototypes in which the prototype is used to build the final system or determine its functionality.

Gudes et al, in the context of an expert system development (322), describe both experimental and exploratory prototypes as well as describing another prototype role, that of a knowledge acquisition tool.

The above covers the full extent of useful software prototyping concepts identified within software engineering. The concepts within the "evolutionary" development of DSS and OR systems, covered in section 5.1.2 above, add little to this. Yet this coverage does not appear to be exhaustive. In particular, evolutionary prototypes appear as a compound function compared to the other defined uses.

The issue was not that guidance was required to decide on the correct use of prototypes and other models. More the problem was to be able to comment on the prototyping which was carried out. It proved possible to identify six uses for models as being worthwhile in this regard.

The first use was termed EXPLORATORY. It covers all uses of models where the output is information rather than a specific deficiency in the model. Thus knowledge acquisition tools and Floyd's EXPLORATORY prototypes are included in this use.

The second use was termed EXPLANATORY. Here the output from the model's use is an informed user or client.

The third was termed EXPERIENTIAL. Here the model of the system or part of it is tested against the problem situation to determine how well it copes. That includes identifying its organizational feasibility as well as its technical feasibility.

The fourth was termed EXPERIMENTAL. Here the model is tested against something other than the problem situation. The crucial difference between EXPERIMENTAL and EXPERIENTIAL models is seen in the source of the `standard' against which the model is tested. With EXPERIMENTAL models, this source is essentially contrived.

Two other uses were defined. An EXEMPLARY model provides learning through examining its internal workings. An EXPOSITORY model provides a storage medium for a designed system.

These six uses of models can be compared with the three types of prototype described by Floyd. EXPLORATORY models remain as he described. Floyd's experimental prototypes are divided into two types, EXPERIMENTAL and EXPERIENTIAL. Evolutionary prototype describes the continuity of model gradually developed into the working system not the use of such a model for any particular role within this process. The six uses of models describe possible uses of such a prototype during its development. The model types EXPOSITORY and EXEMPLARY may describe aspects of the continuity of Floyd's evolutionary prototypes. Finally, EXPLANATORY models are not mentioned by Floyd.

Understanding these different uses of models was considered an important part of understanding the problem-driven approach.

Coping with Expert System Techniques

The continual improvement of tools available for expert system development means that any decision made with respect to this issue could quickly become outdated. This made the choice of appropriate tools less intense; the idea of comparing candidate tools in such developments as the leadtime system development was replaced by simply considering the alternatives available.

The leadtime system had pointed to a need for an unrestricted use of expert system techniques and for non-trivial maintenance interfaces. The first of these criteria appeared to rule out the use of anything other than the most powerful expert system development tools or the use of a so called Artificial Intelligence (AI) language. The second criterion at first appeared to mean that the use of AI languages would entail the investment of a lot of time developing the tools to support the maintenance interface. This would have to be balanced against the cost of purchasing a powerful development tool and the possibility that such a tool once purchased could then prove to be restricting. While the most powerful tools allow access to Lisp as a functional language and so could in this severe manner avoid limitations in the representation structures, the user interface functions may not allow the easy building of a maintenance interface (247), the very reason for using such tools.

A third alternative considered was that of using a proprietary data base management system (DBMS) to provide the maintenance interface for an AI language. Previously, expert systems had been proposed for supporting the operation of databases (370). Also mentioned has been the need of expert systems to interface with a database if the task is "data intense" (371), an instance of this being scheduling tasks (33). The literature revealed no reference to using a database as a means of providing a `user' interface. A further attraction of this third alternative is the ability of the DBMS to act as an `application generator'. Thus the use of conventional software techniques within the DBMS would allow alternative techniques to be used instead of a singular expert system solution.

A fourth alternative considered was the use of Poplog, a system comprising of Prolog, Lisp and POP-11, the latter a procedural language. Within Poplog, commands from POP-11 can be interspersed with Prolog commands (372). Being procedural, POP-11 was considered as a means of simplifying the development of a maintenance interface compared to using Prolog alone. The advantage over using a DBMS was that a far tighter coupling could be achieved between the Prolog and procedural functions. In terms of a maintenance interface, such tighter coupling was not considered important.

A final fifth alternative was also considered; the use of conventional languages, in particular Pascal which supports user-defined data structures. In comparison to using Prolog, Pascal's procedural nature may reduce the task of building a user interface while increasing the task of providing expert system techniques.

The third alternative appeared the most appropriate. The AI language chosen was Prolog (LPA Prolog version 1.2); the alternative Lisp was beyond the experience of the project team and so was not considered a candidate.

The particular DBMS chosen was called DataEase (version 2.5). It was not as powerful as other available DBMSs as an `application generator', the other packages considered being dBASE 3 plus and Paradox. But DataEase appeared to be a more straightforward package. Relations within the database, the menu system and user passwords were all explicitly part of the framework of DataEase. It therefore appeared that dBase 3 plus and Paradox would be less easy to develop as a maintenance interface.

With hindsight, the decision to use Prolog does bear further explanation. In particular, one of the issues of the adaptability of expert systems mentioned in section 5.2.2 was the interdependence between functions being greater within Prolog than within a normal procedural language.

While the design medium of a system existed solely as a prototype, indeed the sole expository model, this difficulty remained significant to the adaptability of the system. With the design carried out in a more favourable medium external to Prolog, the difficulties of using

Prolog as identified at that time were seen to be dispelled.

The problems with Prolog which were yet to be encountered during the project were not then identified, although with hindsight there was indication of their existence. (See section 6.3.)

Implementation Issues

The use of SSM to cope with implementation issues of a computer system has not been specifically addressed in the literature. Neither Wilson (26) nor Wood-Harper et al (361) consider implementation issues directly. Lyytinen (373) discusses the use of SSM to cope with implementation problems of existing information systems but not during the design of such systems.

The task of coping with the implementation issues of a PP&C computer system is not inappropriate for SSM however. In section 3.4.7 these impementation issues were listed as follows:-

(1) Coping with the informal system and the decision-making process.

(2) Gaining the support of top management.

(3) Identifying user requirements.

(4) Identifying information requirements.

The first two of these, coping with the informal system and the decision-making process, and gaining the support of top management can be seen as the fare of classical Mode 1 SSM. As anticipated, within the conceptual design work these implementation issues were quickly identified as were some aspects of the user requirements.

As the detail of the conceptual models increased, the emphasis in the process changed. Higher less detailed models of the system became effectively primary task models and systemic desirability was replaced by technical feasibility as the output from the more detailed lower level diagrams. This prompted recourse to Mode 2 use of SSM. That is, expanding the detail of the specification required less and less use of SSM. Rather SSM was used to provide models to compare against parts of the developed specification. This is a similar role to that carried out by the two constructs, the models of the `roles' and the `tasks' of expert systems described above (figure 5.4 and 5.5) which the author has described as `semi-precious' conceptual models (367). In this manner, part of the Mode 2 use of SSM was used to identify the user requirements, third in the list of PP&C implementation issues above. This work could be described as design work of the `total' system. As the implemented software acts only as a sub-system to this `total' system, such design work remains part of the specification task for the software system. Other Mode 2 use of SSM extended into design of the software itself.

The last item in the list of PP&C implementation issues, information requirements, have been identified for conventional software systems using the methods described by Wilson (26) and Wood Harper et al (361). Within the conceptual trials, identifying the information requirements of knowledge-based activities appeared to involve no extra complication in this process. One exception to this was the maintenance requirements within PP&C, identified as important during the leadtime system development. Such requirements are not considered by Wilson or Wood-Harper et al except in mentioning `design flexibility' and in terms of redevelopment of the system as a whole. Identifying maintenance requirements was seen by the author as a possible use for Mode 2 SSM.

The use of SSM thus appeared from the conceptual design work to be appropriate for coping with the implementation issues of an expert PP&C system.

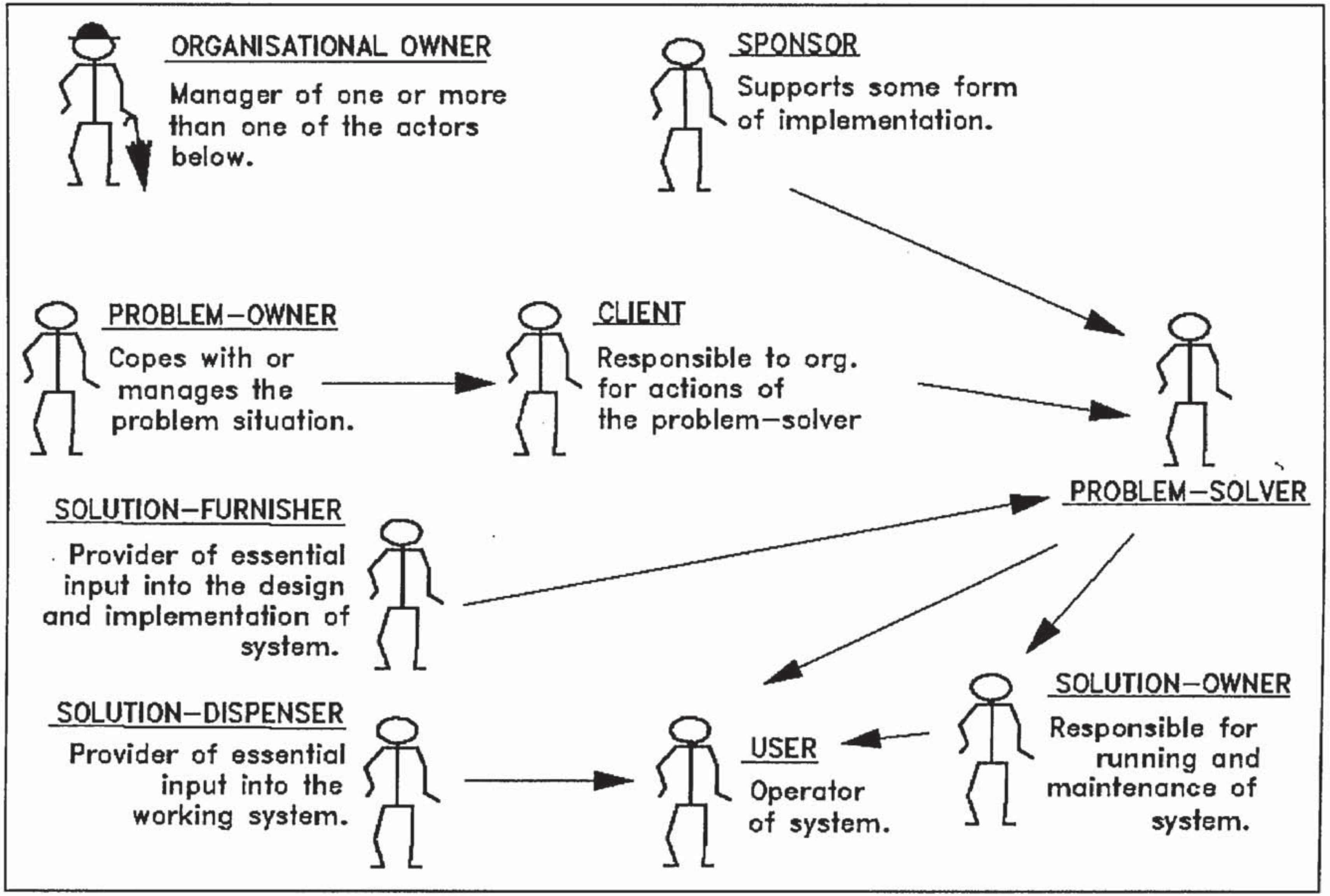


FIGURE 5.6 THE SET OF ACTORS IMPORTANT TO A COMPUTER IMPLEMENTATION.

One area in which SSM did appear to be deficient however was in identifying the actors and their roles with respect to the projected system within the problem situation. Lyytinen (373) also identified such a need with regard to Information Systems. Similarly, the author found the roles of stakeholders explicitly named within SSM to be insufficiently rich. Therefore an extended set of actors was developed (figure 5.6). By identifying such actors within an implementation, problem-solvers can decide whom to involve in the design process, whom to consult and whom to simply keep informed.

The use of SSM in both Mode 1 and Mode 2 and covering the user requirements as well as organizational requirements extends far beyond that described by Moll (321) who developed only detailed primary tasks to identify the composition and boundary of the expert system's role.

5.3.4 TRIALS WITH PROBLEM-DRIVEN CONSTRUCTS

The conceptual design work used to develop and test the constructs for use in a problem-driven approach revolved around five designs. The problems tackled during these developments were not of great concern within BSSP. However to test out the constructs, the problems were considered as though they were.

As the trials were thus effectively working on an erroneous assumption, interactions with stakeholders were not pursued. Information was either taken from the notes of the author's initial finding-out exercise from the beginning of the project or through informal contact with available personnel.

Of the five designs, the first was the leadtime system already developed using the exploratory approach. The problem was taken to be the need to increase the complexity of the leadtime rules and the impact of this added complexity on the organization.

Two other designs were based on other general areas of expertise and rules within allocation, previously mentioned in section 5.3.1. The problems considered within these designs were taken as `the task of allocation' and as `the maintenance of loading rules'.

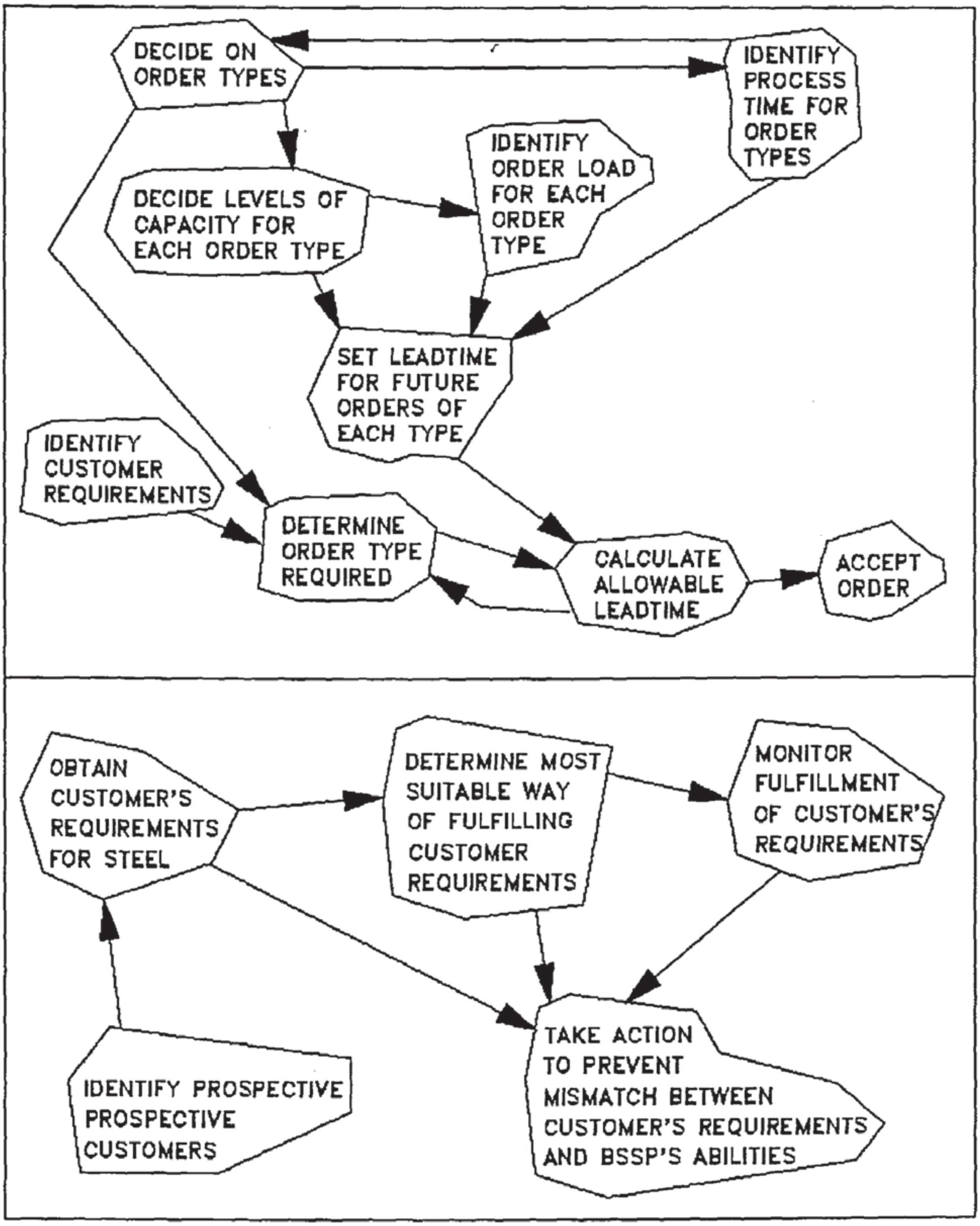
Two further conceptual designs were carried out. The first of these designs was based on an actual problem which was at that time being solved using an engineering solution. That was the coil storage and transport problem at Shotton works for which storage racks were then being installed to replace overcrowded storage bays. With this conceptual design, a computer system to utilize the current facilities more efficiently was taken as an alternative solution to the storage racks.

Finally, the last of the five designs was actually already the subject of an expert system development called Sesame (11), an experimental fault diagnosis system at Ravenscraig works. Little was known by the author about this real world development. It was chosen mainly due to the saying `the exception proves the rule'. Fault diagnosis is not a PP&C task. It was seen as an application less likely to benefit from a problem-driven approach and thus as a useful test for the constructs of the approach.

A detailed account of the conceptual design work encompassed in these trials is beyond the scope of this thesis. While all the five designs showed the usefulness of the developed constructs, this usefulness is best demonstrated by later uses of the problem-driven approach.

There is however one exception; an area which was best illustrated in one of these trials. That area is the tension between user and

FIGURE 5.7 THE TWO CONCEPTUAL MODELS SHOWING THE MISMATCH BETWEEN THE LEADTIME RULES AND THE ROLE OF SALESMEN.



expert and how the concept of a `primary task' system can exist within such a tension; or more correctly, how an expert system design can relieve such a tension. This is described in the following sub-section.

Specifying the Leadtime System

The leadtime system described in section 5.2 was developed using an exploratory approach without any initial specification of its required function.

The problem-driven approach firstly identifies the problems facing an organization or part of an organization, and only then considers how to tackle them. Determining minimum order leadtimes for UK General customers is by no means one of the problems facing Order Allocation. Thus the initial part of the problem-driven approach was by-passed. The starting point for this conceptual trial, then, became a statement of the problem which was taken as being the problem. Using SSM, this statement can then be used to structure the approach to the problem.

The statement of the problem was the starting point for the development of a conceptual model, the model being shown in Figure 5.7. Comparison of this model with reality identified two general functional areas, the formulation of the leadtime and the use of the rules by the user. (The significance of the maintenance interface had been identified from the start.)

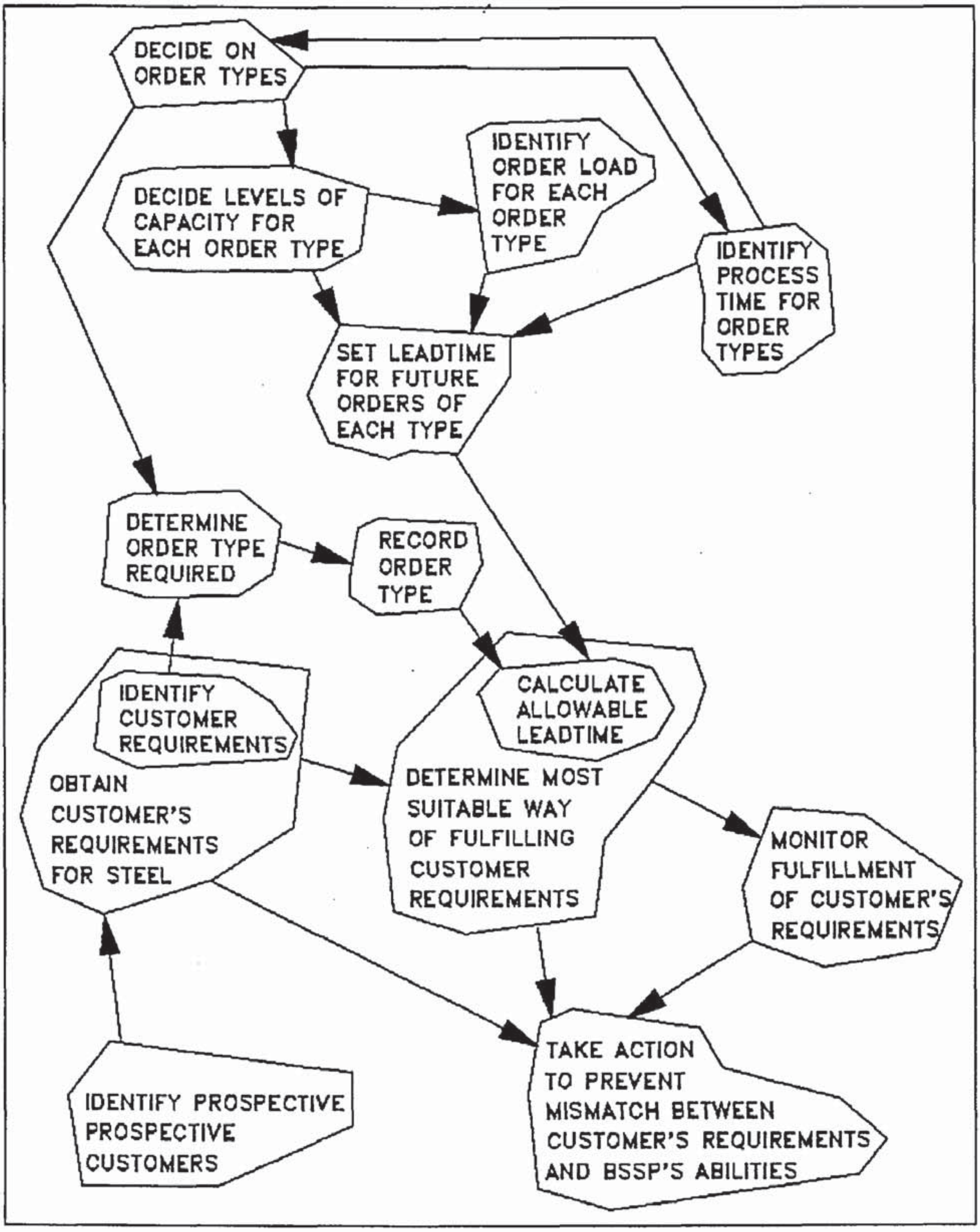
Continued work with SSM gave reason for concern over the second general functional area. It became very apparent that the use of the rules by the user, while a plausible `primary task' activity for both Operations and centrally-based Commercial personnel was certainly not so given the roles of the actual users, the regional sales office staff. Both modelling the role of the regional sales offices and expanding the detail of the original model showed this. Sales office personnel do not normally wish to know the earliest possible delivery for an order. Their initial need is to know whether or not the delivery required by a customer, now or in the future, is going to cause a leadtime problem.

The main difficulty with this situation was that it was not possible to turn the use of the rules around to answer the salesman's question in a sensible fashion. As they stood, the categories and additional rules represented almost nine thousand possible `products'. Most of these `products' were rarely, if ever, encountered. Thus an enquiry checking delivery dates would entail almost as much questioning by the expert system as would providing the earliest delivery for a particular `product'. Added to this, a salesman would often be concerned with multiple items; multiple consultations.

O'Keefe et al (262) stated that "users do not like simply answering a series of questions." Their advice was that expecting a user to answer fifteen questions without some form of system output was expecting too much. Further the allowable number of questions would decrease if the system was not seen as offering beneficial output. The leadtime rules suffered from this problem of too many questions. Indeed the majority of questions would appear as irrelevant during a single consultation.

The situation was such that, in the author's opinion, unless the rules greatly increased in complexity (and in reality this did not happen,) a salesman would be better served using the information as currently displayed on his computer terminal (and shown in figures 5.1 and 5.2) than consulting an expert system. In other words the format of the leadtime rules, while adequately defining an order's leadtime, did not suit the role carried out by the salesmen. A `tension' existed between the role of the users and what would otherwise be a

FIGURE 5.8 CONCEPTUAL MODEL OF LEADTIME SYSTEM DESIGN WITH 'ORDER TYPE' RECORDED ON COMPUTER BYPASSING NEED FOR EXCESSIVE CONSULTATIVE OPERATION.



straightforward primary task.

This opinion was however not the final word. The author analysed the activities of the two models in figure 5.7. This led to a design which reconciled the `tension' between these two models by introducing an extra activity RECORD ORDER TYPE as shown in figure 5.8. In this way salesmen need not be concerned with the detail of the leadtime rules except when a totally new rule is added or a new product is asked for by a customer.

This restatement of the task undertaken by the expert system moved the ownership of the expert system's role from Order Allocation into the Commercial Department. It is a further illustration of PP&C problems not being self-contained.

Defining the functionality of this new role of the expert system then became a commercial decision. Further conceptual work was carried out to prepare for a debate to discuss these findings, in the first instance with Operations staff and then later with Commercial Department staff. This further work concluded the conceptual work carried out on the leadtime system.

5.4 USING THE PROBLEM-DRIVEN APPROACH WITHIN THE PROJECT TASK

This final section of the chapter draws together the various threads of the first five chapters of the thesis relevant to the development of the system which is the subject of the next chapter.

In chapter 2 structuring the problem situation led to a measure of performance for the project; unallocated arrears. As the major causes of unallocated arrears were not known, the initial project task became identifying these major causes through the building of a system, UAAMS (UnAllocated Arrears Monitoring System). The function of UAAMS was to monitor the effects of these various causes on the level of unallocated arrears. In this manner the project could identify and address the most significant causes of unallocated arrears.

Chapter 4 reviewed the capabilities of expert systems. The conclusions were described as "tenuous", but the evidence did suggest a useful role for expert systems within PP&C. However mainstream expert system methodology was criticized for being exploratory in nature and the belief articulated that such an exploratory approach would be inappropriate to the domain of PP&C. The inappropriateness of this approach was demonstrated in chapter 5 with the trial development of the leadtime system. A more appropriate problem-driven approach was developed and tested by carrying out five conceptual designs. Following this work, the approach appeared mature enough to be tried out in a real situation; in the development of UAAMS.

Chapter 3 reviewed the literature covering the different approaches to tackling the PP&C task and the techniques described as useful within PP&C. However each of the various approaches and many of the techniques were argued as inappropriate to the problem situation. The only positive contribution from this literature into the project was in identifying the possible usefulness of Mode 2 SSM, simulation and mathematical programming.

As the approach developed in chapter 5 is problem-driven, it must not be seen as solely an approach to building expert systems. Within the development process the usefulness of other techniques, for instance simulation or mathematical programming, may be harnessed in preference to expert systems should they appear more appropriate. The approach also makes much use of SSM both in Mode 1 and Mode 2 form.

CHAPTER 6
THE DEVELOPMENT OF UAAMS

The focus of this chapter is the development of UAAMS, the system which it was hoped would allow the major causes of unallocated arrears within BSSP to be identified. From this, it was hoped that a means of reducing the levels of unallocated arrears could be determined and that such a means would be culturally feasible in part due to the findings of UAAMS.

The development of UAAMS, being a problem-driven approach did not presuppose the use of expert system techniques. The usefulness of such techniques as simulation and mathematical programming was also considered. However, the original rationale of the project (see section 1.1.2) meant that the use of expert systems would not be dismissed lightly.

6.1 STRUCTURING THE SYSTEM

This section covers the work carried out to structure the UAAMS system. The reasons behind the development of UAAMS are briefly restated (see section 2.2 for a full account) followed by an account of the design work which led up to the consideration of expert systems as possible solution methods.

The Reasons for Developing UAAMS

The first consideration with regard to the development of UAAMS was the use to which the system would be put, and by whom. UAAMS was intended to identify the levels of unallocated arrears resulting from a number of causes and thus to enable action to be taken to reduce these levels of arrears. The user of the system, in the first instance, was to be the author.

That a system was needed at all for this task was argued in section 2.2. The analysis needed to be run over a period of time due to its reliance on current data. Also the analysis needed to be quite detailed. Further, action in reducing the level of arrears would involve stakeholders beyond the project team. Such stakeholders would need to be convinced that the correct causes of unallocated arrears had been identified. In addition, it was not realistic to expect that any action resulting from the project work would eliminate difficulties due to arrears once and for all. The system therefore might well be reused. Finally, the central function within UAAMS, roughcut allocation, was seen as possibly useful within systems for directly preventing arrears; systems which would possibly be commissioned once the causes of arrears had been identified.

Thus the dominant rationale behind the development of UAAMS was the need to reduce the work involved in identifying the causes of unallocated arrears. Further to this were other reasons for `systemizing' the work, reasons which might in due course involve users other than the author.

UAAMS was intended to identify the significant causes of unallocated arrears. Thus an initial input into the UAAMS design was the list of possible causes. The use of the list had not been criticized by any members of the project team. Further, the use of such a list appeared acceptable to the wider set of stakeholders. However, when the results from UAAMS identified the relative significance of each item on the list, both the list and the results from UAAMS could have been subject to criticism because they might be seen as apportioning the responsibility, or indeed the blame, for the arrears.

The use of the list of causes to structure the task set for UAAMS was seen as unavoidable. However, as the list was arguably correct, work to obtain a broad acceptance of it was to be carried out once UAAMS was further developed but prior to any results being obtained.

With regard to criticism of the results from UAAMS, the raw data and the intermediate results would reside in a database built using DataEase. It would thus be easy to analyse this database to answer any unjustified criticism.

With the basic output from UAAMS being a simple list of causes and their contribution to unallocated arrears, the main concern was the amount of maintenance UAAMS would require both for continual use and for reintroducing the system after a period of time. Identifying such maintenance, Pressman's "domains of change" (337), required an understanding of the functionality of UAAMS. The focus of development thus turned from the requirements imposed on UAAMS to analysing the task being undertaken by UAAMS.

The Task Undertaken by UAAMS

In section 2.2 the the causes of unallocated arrears were grouped into seven categories. Using all seven categories however would have encompassed considerable effort especially in terms of data collection. Thus it was decided to reduce the seven categories initially into three groups:-

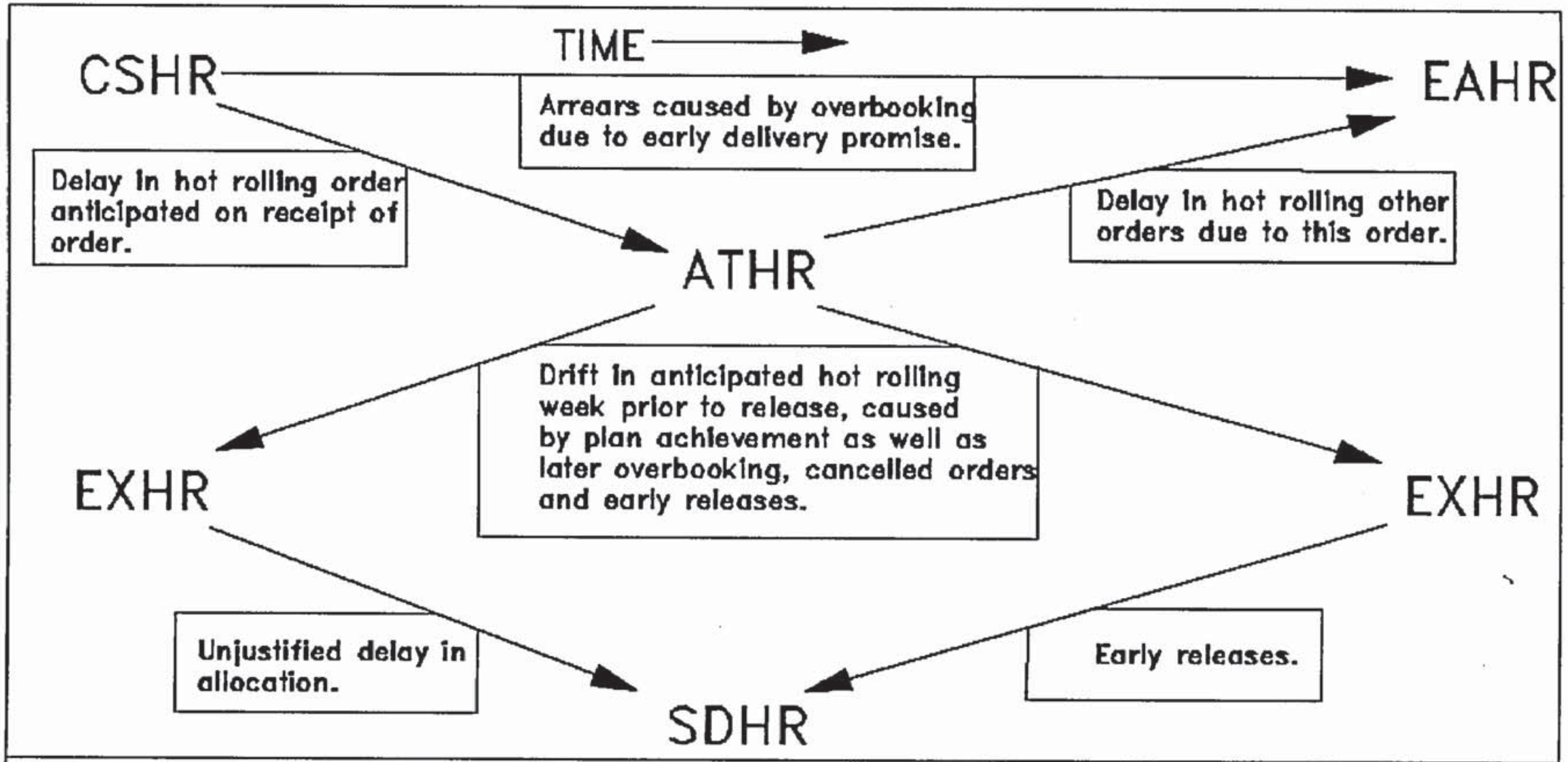
(1) Arrears caused by orders booked above planned capacity.

(2) Arrears caused by poor planning.

(3) Arrears caused by poor allocation.

Once the levels of arrears in the three groups were quantified, the option of further analysing the causes within the groups remained open.

Unallocated arrears occur when an order is released to the works for delivery in a week later than promised to the customer. With the



CSHR Hot rolling week promised to the customer by the sales office on receipt of order.

EAHR Hot rolling week calculated by UAAMS for a newly received order based on receipt date.

ATHR Hot rolling week calculated by UAAMS for a newly received order based on CSHR.

EXHR Hot rolling week calculated by UAAMS for an order during the week UAAMS expects release. Based on CSHR.

SDHR Hot rolling week given to an order by allocation when released to the works.

FIGURE 6.1 METHOD FOR ANALYSING THE CAUSES OF UNALLOCATED ARREARS.

production leadtime correct, this also applies to the hot rolling week as well as the delivery week. These hot rolling weeks are known by the names of the `fields' within the central computer order files. The week promised to the customer is known as CSHR and the allocated (or scheduled) week is SDHR. Thus, when SDHR is later than CSHR, unallocated arrears occur.

The author identified three other calculated hot rolling weeks which would allow arrears to be divided between the three groups defined above. These were termed EAHR, ATHR and EXHR.

EAHR represented the earliest hot rolling week determined at the time the order was received which would prevent overbooking. Only part of the difference between CSHR and EAHR would directly affect the order thus booked. This is because EAHR represents a `first come first served' priority method. Thus a second hot rolling week, ATHR, represented an anticipated hot rolling week with the priority based on CSHR.

The anticipated hot rolling week could change while an order remained unallocated. This could be due to changes in planned capacity and actual plan achievement. However, the anticipated hot rolling week can also change due to later overbooking, cancelled orders and orders being released early. The third hot rolling week, EXHR, represented the week which, given priority based on CSHR, the order should have been scheduled and released to the works. Differences between EXHR and SDHR then represent early releases or unjustified delay in allocation.

The inter-relationships between the three calculated and two original hot rolling weeks are illustrated in figure 6.1.

Using the three new hot rolling weeks, the levels of arrears resulting from the different causes can be calculated. The first two causes are easily calculated. Arrears resulting from overbooking occur when CSHR is earlier than EAHR. Similarly arrears resulting from poor allocation occur when SDHR is later than EXHR.

The third cause is more involved. During the time that an order is awaiting release, a difference between ATHR (calculated at the time of order receipt) and EXHR (calculated at the time when it should be released) can be caused in four ways. The first way is due to overbooking; if the ATHR of a new order is earlier than the `affected' order and the EAHR is later. The second way would be caused by the cancellation of an order ahead of the `affected' order which would reduce overbooking. The third way results from an order being release with SDHR earlier than the `affected' order but with EXHR later, or visa versa. The remaining fourth way of causing a difference between ATHR and EXHR is due to plan fulfilment.

The one complication in using the difference between ATHR and EXHR was the means of identifying an `affected' order. The solution was seen in recording the delays due to the first three causes through time for each route (`route' being defined in section 6.2). Any further variation between ATHR and EXHR would thus be caused by plan fulfilment.

With the values of EAHR, ATHR and EXHR known, the task of calculating, validating and justifying the causes of arrears was seen as an appropriate task for database processing. This then left the task of identifying the values of EAHR, ATHR and EXHR; the roughcut allocation task.

It was evident that the next stage in the development of UAAMS would be to structure the functionality of UAAMS; an initial stage in its design. It was also evident that the major part of the functionality of UAAMS was the roughcut allocation task.

Also considered at this stage was the scope of UAAMS. Would it cover all products of BSSP or only a small number? The client was eager that UAAMS would cover cold reduced products as this was at that time the `problem' product. (It did not remain the `problem' product for long as it turned out.) Thus the decision was to cover hot rolled and cold reduced products with other products being considered only as products also using hot and cold mill facilities.

The Initial Structuring of UAAMS

The initial structure of UAAMS is shown in figure 6.2. This first attempt at developing a structure illustrated that the task of producing a roughcut allocation was not well enough understood. The minimum necessary set of activities could be developed with reasonable confidence; however, as with the conceptual model of PP&C developed in section 3.3.3 above, the logical dependencies remained indefinable without more specific knowledge of the situation. For instance, when routing an order, should the order load be considered or is the order load so variable as to be best considered only within re-scheduling?

The initial structure also involved a design assumption. This concerned the method of loading orders onto capacity to identify overloads as a first step in deriving a feasible schedule. The alternative, scheduling the orderload using some priority scheme up to the available capacity was assumed as less appropriate to what was being built; a rough-cut allocation system. However the alternative was not dismissed; rather, it was to be considered again during the design of the re-scheduling procedure.

As shown in figure 6.2, the various procedures within UAAMS were

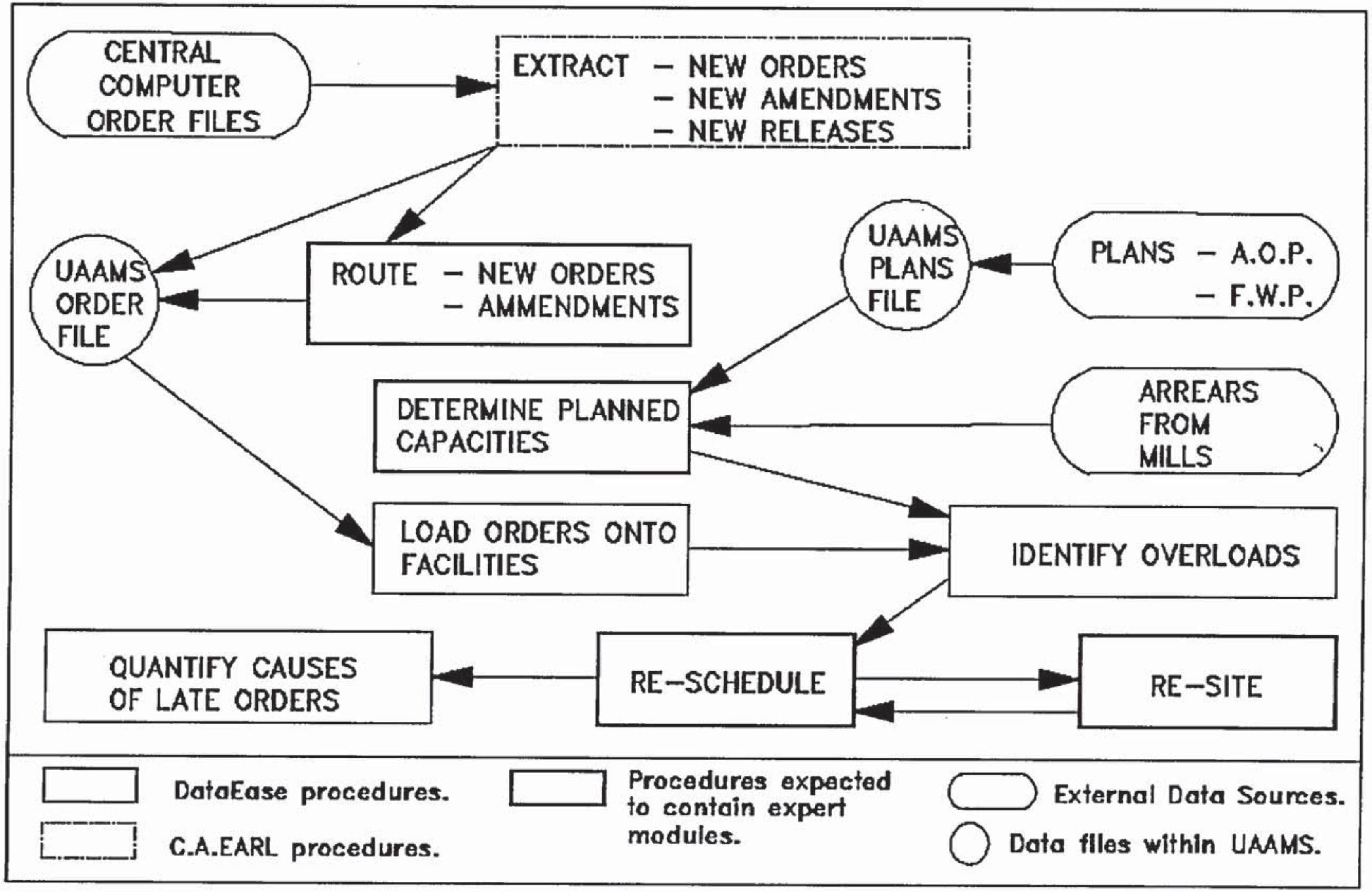


FIGURE 6.2 THE INITIAL STRUCTURING OF UAAMS.

seen to require different types of solution.

Three external data sources were identified at that time. One of these data sources, the central computer order files, could be accessed easily to identify new orders, amendments, cancelled orders and newly released orders. A straightforward database function would then allow data to be down-loaded into a DataEase database run on a micro-computer. Data from the other two data sources were considered as best entered manually into UAAMS. The main constraint with the availability of data was the inability to identify interworks orders resulting from the specific release of an order to the works. This was not considered critical however.

There remained then seven procedures within UAAMS. The last, "quantifying causes of late orders" was discussed above and considered as a straightforward database application. As shown in figure 6.2, two other procedures were also considered at this stage as database applications.

The remaining four procedures were considered as possibly containing expert system modules to some extent or other. The next step was then to expand these procedures to define the scope of any expert system modules contained within the procedures.

6.2 FRAMING THE EXPERT SYSTEM MODULES

The design work next turned to the expansion of the initial structure of UAAMS and in so doing to define more precisely the uses of expert systems. The first three sections successively describe the expansion of the routing procedure, the capacity planning procedure and the "load capacity" procedure. The following two sections describe the expansion of the scheduling procedure which comprised three of the activities within the initial structure. A final section presents the lessons deriving from this work.

6.2.1 THE ROUTING PROCEDURE

Analysis of the routing task identified a straightforward role and task for two expert system components. The knowledge required by such a component and methods currently used within BSSP were then identified. However the major consideration within the routing procedure was not concerned with expert systems. It was the development of a logical model through which to record the various routes.

Analysis of the Routing Task

Three models of the routing task were built to assist in the analysis, the first two based on the `Intelligent Agent' and the third analysing the possible `roles' which could be played by a routing system, both constructs described in section 5.3.3 above.

The first model, shown in figure 6.3, identified the relationship between design and routing. Effectively, routing relies on design to define the required production processes. Indeed it is design itself which is concerned with what is achievable through the different production processes. Without expanding the routing task to carry out some design functions, the task of routing remains quite straightforward.

The outputs from the routing task also required defining. The output was not to be a decision of where an order was to be produced; this was seem as a scheduling task. However, to decouple it from the scheduling task, the reasons for any routing constraints needed to be recorded as such constraints could be overruled by scheduling decisions.

The second use of the `Intelligent Agent' expanded the PLANNING

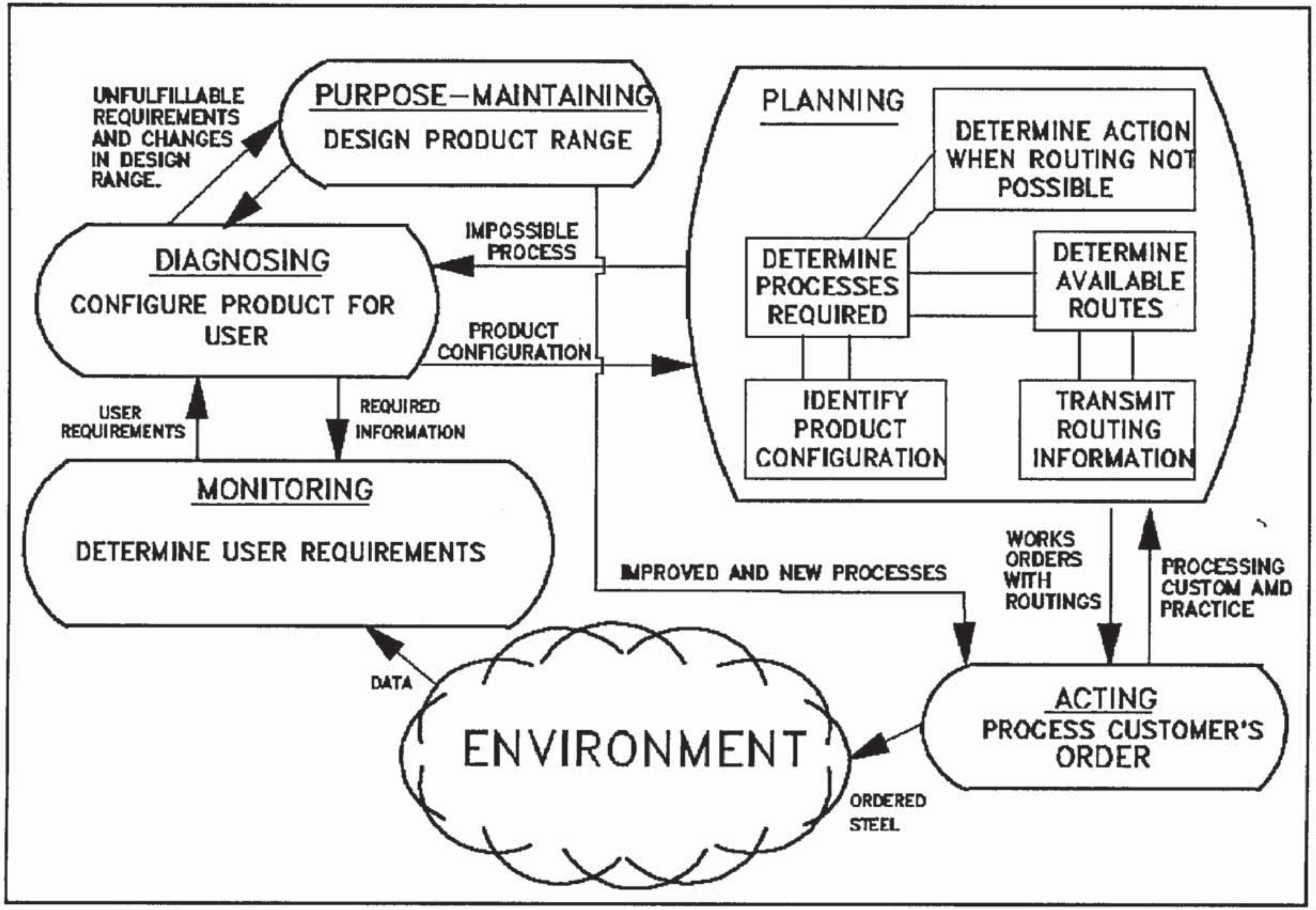


FIGURE 6.3 THE "INTELLIGENT AGENT" MAPPED ONTO THE ROUTING TASK SHOWING ITS RELIANCE ON THE DESIGN FUNCTION.

activity of figure 6.3. Both this and the model based on the `roles' of expert systems identified little more than the first model, reinforcing the belief in the straightforward nature of the routing task. The one new consideration was the identification of allowable products and how to handle errors within the input data.

Consideration of Currently Used Routing Methods

Expansion of the routing procedure had identified an uncomplicated role for any expert system component. This role could be considered in two parts; identifying the processing required by an order and identifying which works were capable of carrying out this processing. Such tasks were of course carried out within BSSP already. Thus a major consideration was to determine to what extent these traditional methods could be harnessed.

The first part, identifying the processing required by an order, is normally carried out within BSSP using C.A.Earl subroutines known as the "product decodes". These are maintained by the allocators and their use by UAAMS would require no extra maintenance work. However such decodes only identify part of the processing required by an order. The rest of the task is carried out by various other C.A.Earl programmes. These programmes contained rules required by UAAMS but not in a usable form. Also these rules were found to access much of the same data as the product decodes.

A number of design options were considered at this stage. The decision made was to incorporate all rules within a knowledge-base within UAAMS.

The second part, identifying which works were capable of producing the order, was not normally carried out by computer. The central computer order file did contain fields showing `feasible' works for each order as well as identifying customer `preferences' and `insistences'. However, this information did not extend to the use of a works through interworks orders, a vital requirement for UAAMS. Coping with interworks orders, indeed feasibility generally could be achieved through the building of a knowledge-base covering the product manual which lays out the product range of each works. This would result in a large knowledge-base.

An alternative method was to use the `feasibilities' from the order files and add to this rough information from works personnel as to which types of interworks orders could be routed to the different supplying works. As the initial version of UAAMS was to cover only hot rolled and cold reduced products, such information would be required only from the three cold mills and so this option was chosen.

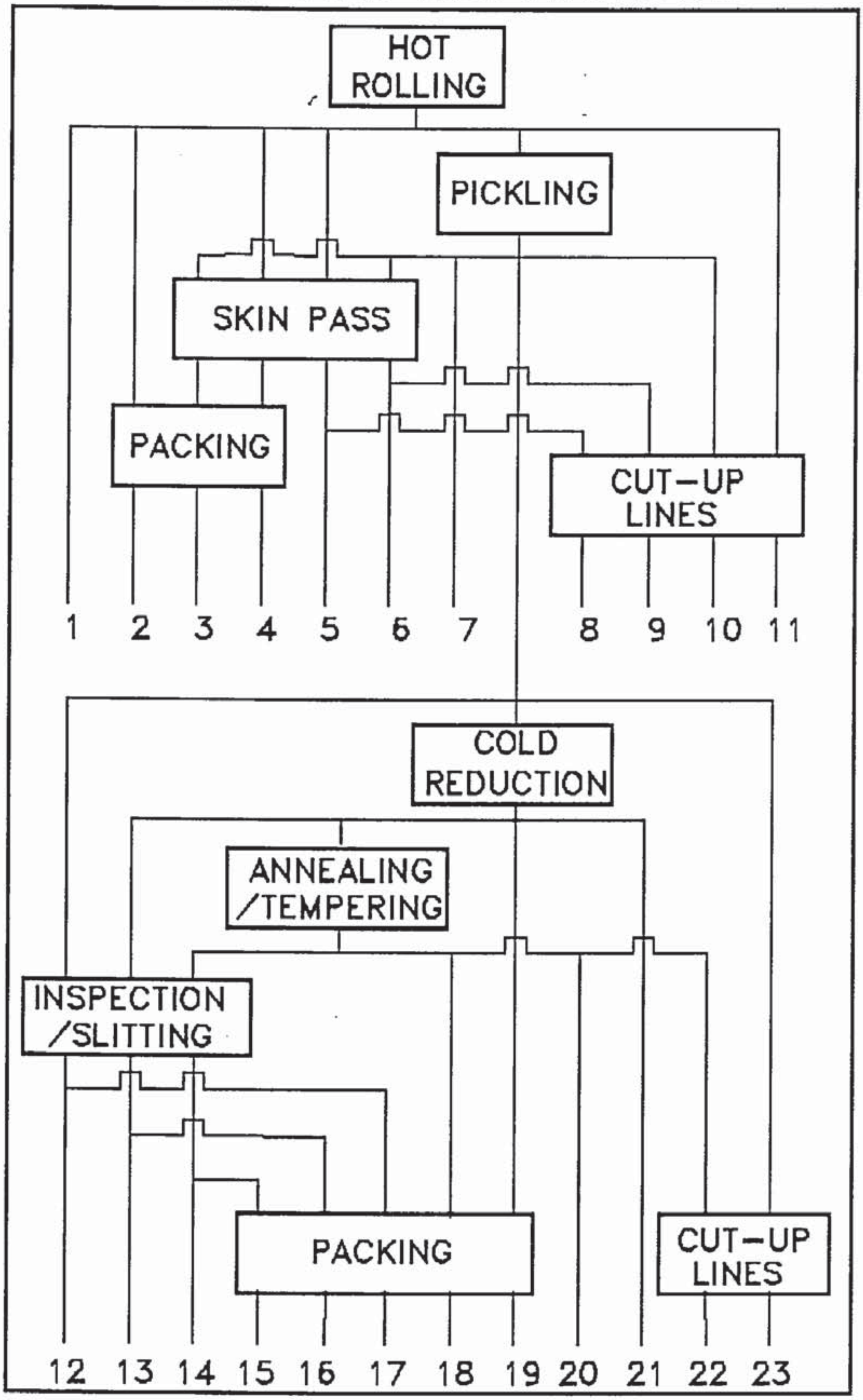
The Knowledge Involved in Routing

Hart (374) introduced the terms `deep' and `shallow' (Hart actually used the terms `deep' and `surface') to describe knowledge-bases with and without underlying representations of fundamental concepts.

The knowledge used in this first part of the routing task can be seen to be essentially shallow in nature. The processing required by an order depends on customer requirements and the design of the product. Thus any underlying representation would have to encompass product design, an impracticality in the case of UAAMS.

The second part of the routing task can also be seen to be shallow, although less so. The underlying representations of the capabilities of different machines are naturally translated into heuristics. In the case of BSSP these heuristics appear as the product

FIGURE 6.4 THE TWENTY-THREE TYPES OF HOT AND COLD MILL PRODUCTS.



manual. No underlying model of the process is required to determine, say, whether the coil diameters are achievable at a particular works. An underlying model of the product also appeared of little use. Equating physical dimensions to the coil weight being the limit of such a model within strip steel production.

The `shallow' nature of routing systems is also suggested by Feigenbaum et al's description of Northrop's routing system, ESP (197).

A Logical Routing Model

As stated above, the main consideration within the development of the routing procedure was little concerned with expert systems. This was the development of a logical routing model for UAAMS, especially how the availability of alternate routes should be recorded.

The design of this routing model defined much of the format for the plans to be used by UAAMS. As such the design of the routing model could not be made in isolation from the design of the plan format. Thus design of the routing model also had to take account of how the plans would cope with such things as plans for the different markets not covering all facilities, different measures of capacity for different facilities, as well as complications such as wide orders which could be processed by only a limited proportion of the capacity as defined in the routing model.

The logical routing model developed was not seen as the solution but only as a solution. The routing task was broken into the two separate parts. Firstly an order would be categorized as one of twenty-three types of product (figure 6.4). The different sites which could carry out the different stages in a route could then be identified.

The decision was made that the output from the routing decisions would be, for each works at each stage in a route, a label such as feasible, infeasible, preferred, insist, etc, different attributes defining different constraints. For instance, an insist referred to a customer insist. Such terminology generally conformed to that used by allocators. To allow the scheduling of aggregate loads rather than individual orders, these different attributes would later be converted into feasible and infeasible. These then could be used to identify an order `unique' to one site at one stage and an order `free-to-allocate' to more than one site. Identifying `unique' and `free-to-allocate' orders was not seen as a routing task. It was eventually to become part of the `load capacity' procedure.

The routing data required within the scheduling task is dependent on the routing complexity. Within BSSP, no type of facility existed on more than four sites. Also significant was that when four sites were involved, the preponderance of siting options involved more than two sites. Given these two circumstances, the scheduling task required the following data for each route; the load of `unique' orders on each site for each stage, the load of `free-to-allocate' orders applicable to each site and the total load of `free-to-allocate' orders.

6.2.2 THE CAPACITY PLANNING PROCEDURE

The "capacity planning" activity was required within UAAMS because there was no definitive weekly plan available beyond the horizon of the Four Week Plan (FWP). While the FWP was to be entered into UAAMS by hand, as was the Annual Operation Plan (AOP), extending the weekly plans beyond the FWP was considered difficult enough to require computerized assistance. That is, for instance, converting a quarterly plan into weekly plans would involve a lot of calculation. Further, any such planning would have to cope flexibly with the split of capacity

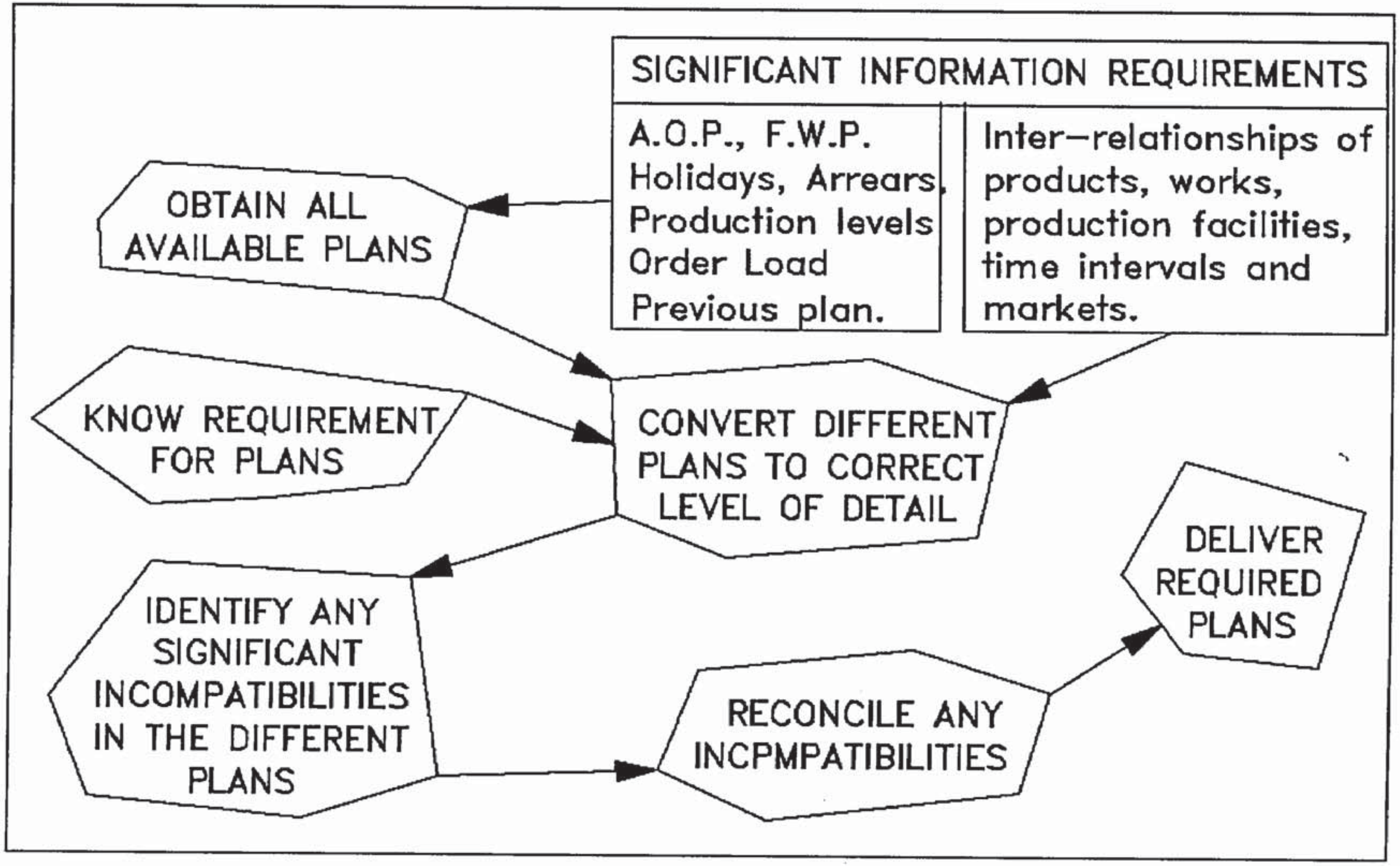


FIGURE 6.5 THE INITIAL EXPANSION OF THE CAPACITY PLANNING PROCEDURE

use on each site between different products and different market sectors. Such consideration intimated the need to use expert systems for this task.

The initial expansion of the capacity planning task is shown in figure 6.5. This diagram was analysed by first identifying the major information requirements and then considering the effects of such inputs on the activities in the model.

This analysis of the "capacity planning" procedure showed that much of the understanding of the task had been misplaced. Certainly the scale of the task was not as great as initially thought. The AOP is seldom revised. Thus it was not considered necessary to automate its expansion from a quarterly to a weekly plan. Also the existence of previously derived weekly plans used during earlier running of UAAMS had not been taken into consideration. Amending these previously used plans was not considered a difficult task.

One issue which was highlighted concerned the reasons for differences between the expanded weekly AOP and the extended weekly plans used by UAAMS. Mill performance could cause divergence from plan and would be compounded by the interactive process of strip steel production. More importantly, variations in demand coupled with differing interpretations of such quarterly plans within the sales functions could also be a major cause of divergence. Indeed, instances of this have occurred. An example was the under-use of capacity by a sales organization not allowing them commensurate over-use later in a quarter as expected by them.

This issue was seen as important to the project and the operation of UAAMS only if planning and plan fulfilment were found to be a major cause of unallocated arrears. As a result the decision was made to use initially within UAAMS only the FWP and a weekly version of the AOP derived manually from the quarterly AOP.

The final consideration with regard to the "capacity planning" procedure was the output from the procedure. "Delivering required plans" was not simply providing a set of planned output levels. Also required would be information to manipulate them. Analysing the situation using the "Intelligent Agent", the difficulty with manipulating plans was found to revolve around how to update or amend inadequate plan configurations; for instance, how to cope with a new production facility or a new type of `campaign'. If the format of the plans were known to be adequate and not subject to change, implicit data structures would suffice. Here, implicit data structures were evidently inadequate.

An expert system solution was not the only possible solution however. The relational database techniques available within DataEase provide explicit relationships although the meaning of the relationships remains implicit. Such a use of a relational database is shown below in section 6.2.3. Analysis showed that relational database techniques would suffice for the purposes of UAAMS.

6.2.3 THE LOADING CAPACITY PROCEDURE

The procedure "load capacity" proved not to need the assistance of expert system components. Indeed, the design proved to be well within the abilities of conventional database methods. This design is shown in figure 6.6 and was actually implemented in DataEase, mainly to gain experience in the use of the package.

The procedure incorporated the conversion of the `routing' attributes (described in section 6.2.1) into the more definite

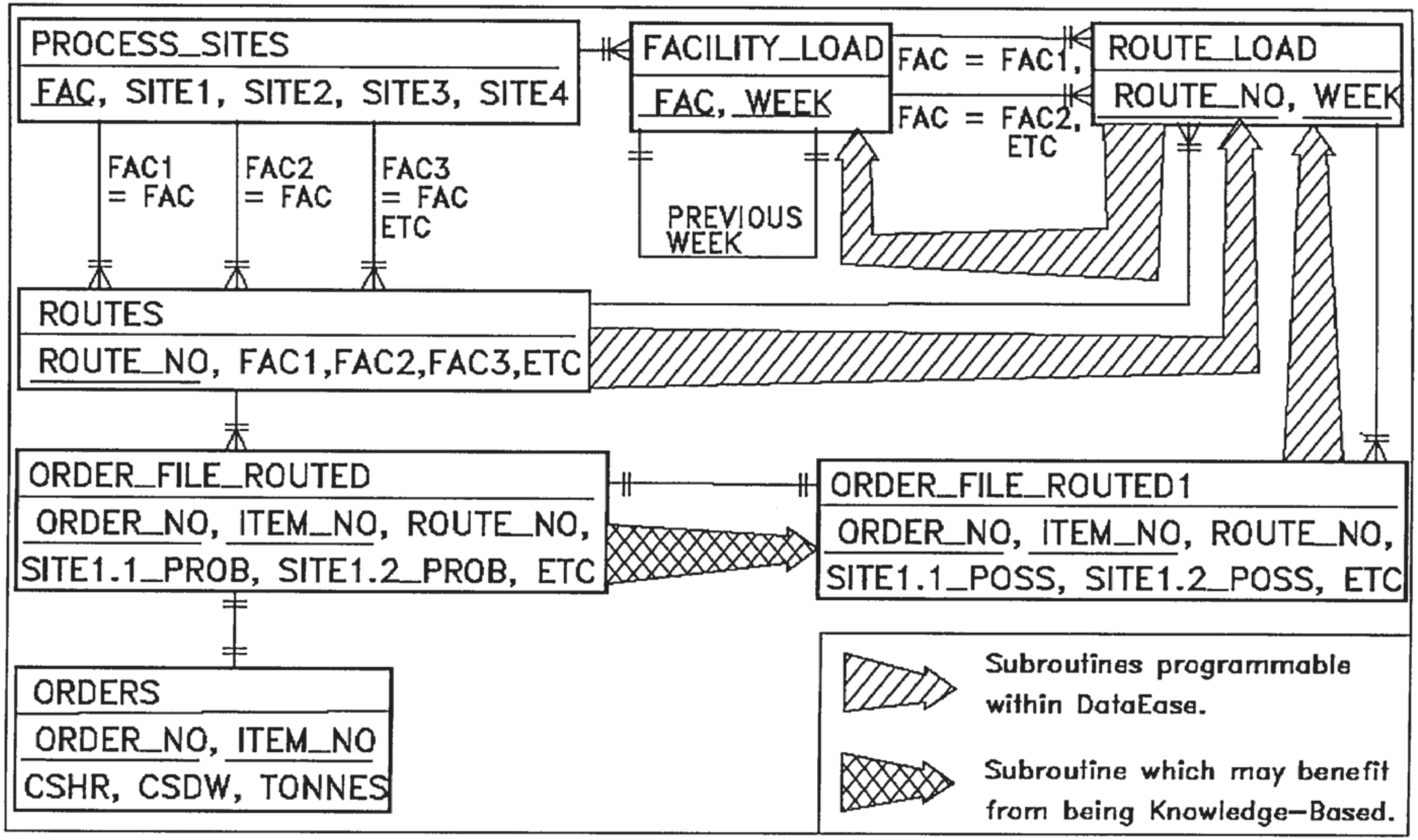


FIGURE 6.6 DATA STRUCTURE AND PROCESSING WITHIN THE "LOAD CAPACITY" PROCEDURE.

`feasible' or `infeasible', a task considered as possibly requiring expertise.

The expertise was required to decide whether a particular `routing' attribute would be converted in the standard way or not. Illustrating this, a bottleneck on a particular hot mill cut-up line would override a `preference' or an `insistence' to prevent them becoming `unique' to that mill. There could however be exceptions. Perhaps skin pass orders would remain `unique' due to bottlenecks at other mills.

The design as implemented had no facility to override a `routing' attribute let alone cope with exceptions. However design work was carried out identifying alternate methods of incorporating this expertise.

One significant point with this design was the restriction imposed by the available data. The `routing' attributes recorded in the central computer order files were acknowledged to be deficient. For instance, while some customers could `insist' on a particular works, other customers were known to reject the possibility of particular works supplying them. Identifying such a rejection was not possible from the data provided by the central computer order files. Incorporating such features within UAAMS then would require data acquisition and maintenance.

6.2.4 SCHEDULING METHODS

Scheduling lay at the heart of the rough cut allocation task and thus at the heart of the operations of UAAMS. How the scheduling was to be achieved could thus be argued as the most important aspect of the UAAMS design.

In section 6.1, it was pointed out that the initial structure of

UAAMS included an assumption as to the method by which UAAMS would carry out the scheduling task. Here, this and other approaches to the scheduling task are considered.

The first three sub-sections consider the different methods of scheduling. The fourth sub-section covers the use of SSM as a first step in providing a structure to the scheduling task. A final sub-section reflects on this design work.

Consideration of Alternate Solutions

It can be seen from the literature that a practical theory of scheduling has yet to be developed. Existing research, as described in section 3.3.4, has not yet progressed beyond the preliminary stages. Given this situation, different sources were considered in an attempt to identify relevant scheduling methods.

The most important source was the methods already used in BSSP. Also considered here are the techniques identified in chapter 3 as possibly useful to the project; simulation and mathematical programming.

The Relevance of Scheduling Methods Used in BSSP

Three activities within BSSP are relevant to the scheduling task. These are the allocation task itself, development of the FWP and the development of the AOP.

The allocation task follows closely the FWP. However allocators also take account of more of the complexity found at mill level, as described in chapter 2. This complexity is articulated by the allocators in terms of constraints imposed on the release of orders. Such constraints vary in importance depending on how much they affect initially the allocation task and ultimately the FWP. Such constraints, then, feed back into the development of the FWP. These constraints however could be identified even though they did vary in importance and change over time.

The FWP is heavily dependent on the AOP. Indeed the AOP itself defines very closely the planned capacities. Variations from the AOP are generally caused by mill performance and order overloads. Reconciling problems caused by such variations can be seen as an available pool of expertise. However the major inputs into such reconciliation are quite low level priorities. That is the FWP may adapt the AOP given available capacity by, say, maximizing tinplate outputs. Such a `policy' however is an immediate `policy'; it would not apply over the length of the orderbook, only over the length of the FWP. Such `policies' could not be used without some understanding of the timescale over which they applied. This was not seen as practical.

This then left the methods used to develop the AOP as a possible relevant source of scheduling methods. The AOP is developed by dividing the output of BSSP into distinct products; hot rolled coil, hot rolled pickled coil, etc. Each product is then equated to an `aggregate' route through the production facilities of BSSP incorporating the concept of `yield'. The demand for these various products can then be converted into a demand for each product on each facility. Generally the priority products are the down stream products, although trade-offs will occur in extreme situations. Trade-offs between different markets also occur.

The idea of loading different products onto different routes and then coping with overloads by identifying priority routes appeared one solution to the scheduling task. However the expertise used to carry out the trade-offs within the AOP and FWP was not seen as appropriate. Firstly, except in the case of the AOP, trade-offs applied only to a short period of time, a period less than the length of the orderbook.

Secondly, such trade-offs were often the output of complex negotiations. Modelling such negotiations was not seen as possible.

The relevant tasks within BSSP then were not immediately usable for the task in hand. Neither the development of the AOP nor of the FWP aided the task of reconciling overloaded routes.

The constraints imposed on the allocation task did however appear to be useful enough to be considered as a possible input into the design of a roughcut allocation system.

The Usefulness of Simulation and Mathematical Programming

Simulation as a method to provide a schedule is not appropriate to the problem as stated. Simply, large numbers of complex interactions are not present once the orders are routed onto fixed capacities using defined leadtimes. The difficulty is in deciding which orders to load onto which capacity rather than determining the consequences of such actions.

The use of mathematical programming to provide a schedule was less easy to dismiss. The scheduling problem could be seen as balancing the achievement of the planned output from each route given the capacity constraints. Three major reasons lead to the dismissal of mathematical programming as a solution method. Firstly the different routes were not all as significant as each other. Major capacities (eg, the hot rolling process) would not be affected greatly by minor capacities (eg, the skin pass). Thus the problem of balancing the achievement of twenty-three different routes was capable of being broken into sub problems.

A second reason for not using mathematical programming was the complexity which would also be involved in the problem. Balancing the achievement of planned output for each route involved was seen as a means of obtaining a schedule rather than an objective. Also the balance could be seen as achievable over different time periods. Thus the mathematical programming formalism, the solution of a set of simultaneous equations such that some other function is optimized does not immediately appear to fit the stated problem. The author has suggested that consideration of the design process of mathematical programming may well show need for conceptual support (293).

A final reason should not be overlooked. That is a designer's abilities with the available method (319). The author had little experience with mathematical programming and was thus less inclined to accept it as a solution method.

The Chosen Scheduling Method

In the absence of a practical theory of scheduling methods, many scheduling methods were compared using Soft System Methodology (SSM) to identify alternative approaches to the scheduling task. The methods considered included those discussed in the preceding two subsections as well as those reviewed in chapter 3. Such use of SSM to clarify concepts has been described elsewhere (1). However, within the context of a design process, such use of SSM the author described in chapter 3 as Mode 2 SSM.

The use of SSM identified two important inputs into the choice of scheduling method from the problem situation itself. It appeared impractical to ignore the use of routes (more correctly, product types) and the leadtimes used within BSSP. This was necessary to allow use of the FWP and AOP. Once this had been identified as a major design decision, the choice of scheduling method became quite straightforward. The assumption made in section 6.1 could then be defended.

Figure 6.7 shows the scheduling method chosen in the form of a

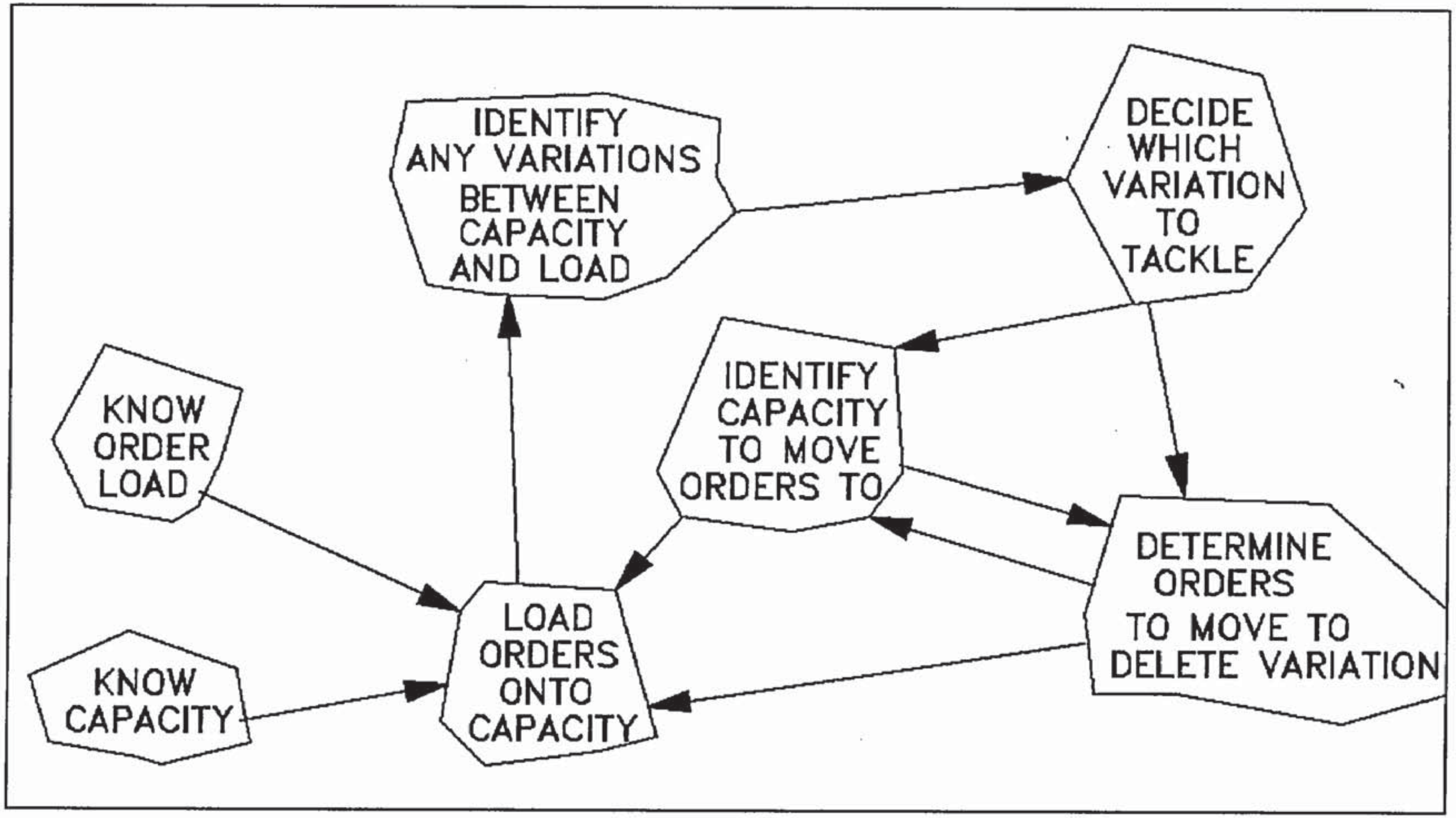


FIGURE 6.7 CONCEPTUAL MODEL USED WITHIN THE DESIGN OF THE SCHEDULING PROCEDURE.

conceptual model. The method is analogous to a scheme for closed-loop Material Requirements Planning (MRP). Such a scheme is possible here because the plans allow separation between the different products and markets. Also there are none of the complications involved with batch sizes and stock levels.

The design of the scheduling method had developed markedly since the initial structuring of the UAAMS system (figure 6.2). The design represented three of the procedures within the initial structuring of UAAMS; rescheduling, re-siting and overload identification. Nor was the design of the scheduling method complete. Indeed, Figure 6.7 was one of the conceptual models generated to identify the scheduling design options mentioned above. Figure 6.7 explicitly represented a number of design decisions. To be satisfied with these decisions, further design decisions had been made or the need for further decisions identified. In this manner the implications of the previous design decisions were examined. The result of this work was to confirm the scheduling mechanism represented by figure 6.7 as a usable scheme for carrying out the scheduling task through the harnessing of expert system methods to direct its use. However, many design options remained to be identified and the much of the system's design at this point of the design process remained speculative. This design is presented in figure 6.8 and will be described in section 6.2.5.

Beyond parts of the design expected to use the mechanism represented in figure 6.7 were two other procedures, assumed to be expert procedures. These two `appended' procedures were not seen as an integral part of the design. Rather they were seen as additional refinements which might not be needed in the final system. The first would check and if required amend the schedule at an individual order level. The second would provide input into the "Loading Capacity" procedure, allowing preferences for overloaded works to be ignored.

A final input into this stage of the design was consideration of the allocation constraints considered previously to be of possible use within the design. Many of these constraints, for instance constraints due to minor facility capacities, would be catered for within the scheduling procedure as it stood. The remaining allocation constraints, for instance the batching of orders into `campaigns', mainly affected only a small part of the orderbook for hot rolled and cold reduced products and could thus be dealt with by the two `appended' procedures.

Reflecting on the Design Process

The design process used to develop the scheduling method started with the use of SSM to clarify possible design options. As the design became better defined, the subjects of the conceptual models built became more specific to the situation. Eventually the designs being considered became too specific for SSM to be used to model them directly. From this point however the problem was well enough understood to develop specific designs and test then against criteria developed during the preceding analysis.

The design process had three major inputs. These were the problem situation, the different scheduling methods and the problem being tackled itself.

Although not complete, the design then passed from considering scheduling methods generally to considering expert system solutions. The transition was initiated by the identification of expert system `solutions' amongst the design options and the acceptance that the mechanism represented in figure 6.7 was acceptable.

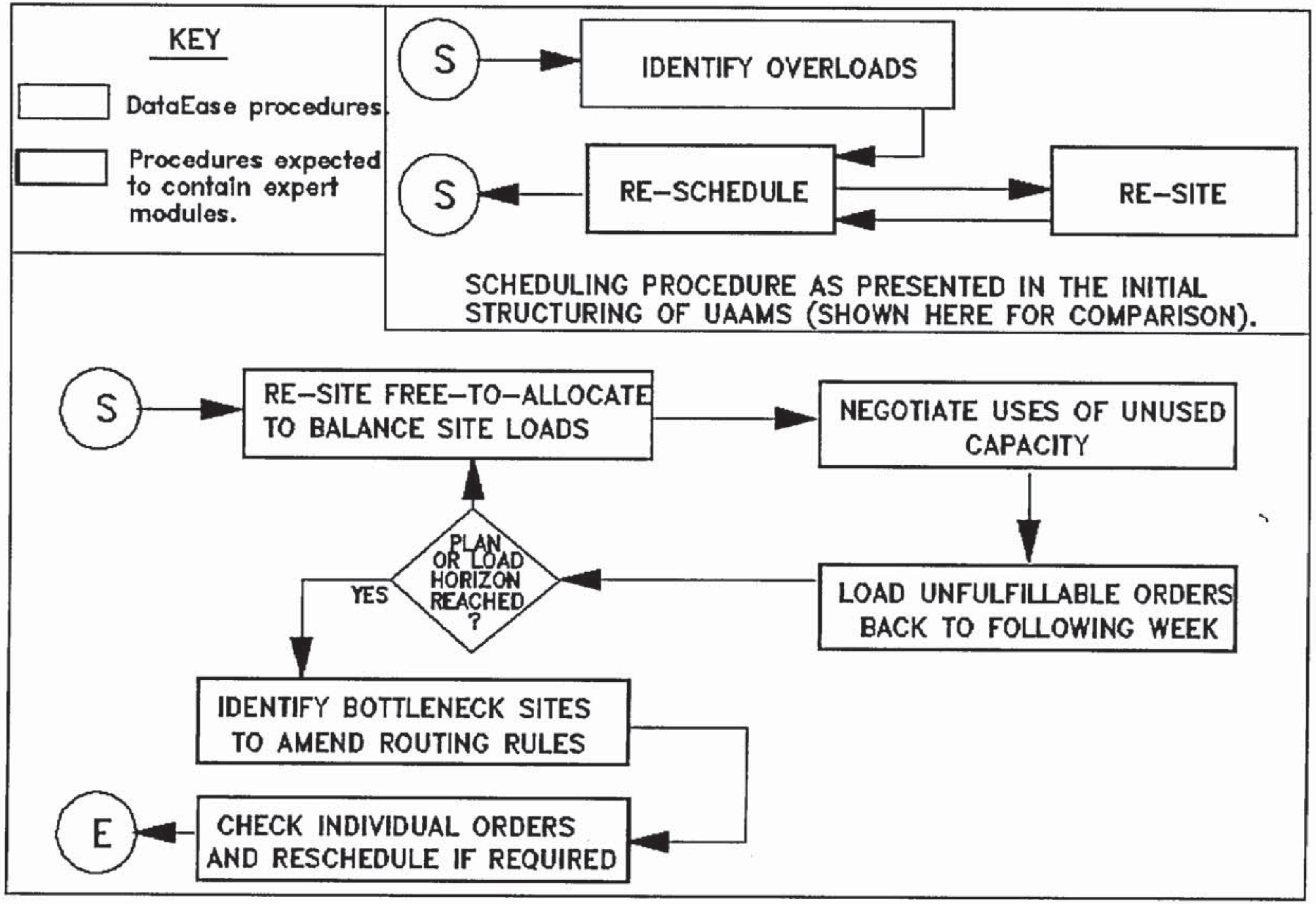


FIGURE 6.8 MODEL SHOWING THE REDEFINED SCHEDULING PROCEDURE DESIGN PRIOR TO THE FRAMING OF THE EXPERT SYSTEM MODULES.

6.2.5 FRAMING THE EXPERT COMPONENTS WITHIN THE SCHEDULING PROCEDURE

This section describes the continuing design of the scheduling procedure in terms of expert system solutions. The first sub-section describes the state of the design at the start of this work.

A Description of the Design at the Start of the Framing Process

The start point for the framing of the expert components within the scheduling procedure was the design, much of it speculative, presented in figure 6.8. Within this design were two appended expert system modules. The two appended modules were not seen as requiring further design work. The need for such modules within the scheduling procedure was not definite and they were apparently independent from the operation of the rest of the scheduling procedure.

The remainder of the scheduling procedure design consisted at this time of an iterative loop. Within this, the `Identify Overloads' activity from the initial structuring of UAAMS had been replaced by the mechanism of figure 6.7 used within different activities within the loop.

The "re-site" activity remained as a separate activity. However its function had not remained constant during the design process. The "re-site" activity had originally been considered as a mechanism for utilizing the capacity on different sites to avert overloads. This had changed to a simple mechanism for coping with `free-to-allocate' orders by preventing such orders from being loaded onto sites which would thus contribute to an overload situation.

The iteration within the scheduling procedure initially provided an overall mechanism for the procedure's operation. The underlying logic was that any order not allocatable in a certain week would be loaded back onto later weeks. The gradual understanding of the mechanism of figure 6.7 however showed this iteration to be less important than the different ways of re-scheduling the order load and that these ways of re-scheduling would be better described as `negotiations'. The effect of this was to identify the requirement for an overall method to decide which was the correct type of `negotiation' to carry out at any particular time during the scheduling procedure.

This then describes the state of the scheduling procedure design at the point in the design process when consideration turned to expert system solutions.

Checking the Effect of Not Using an Iterative Method

Consideration of an expert component to replace the iterative loop within the scheduling procedure led to the use of SSM to examine the function of the loop. This analysis confirmed that the loop simply provided a systematic approach to the scheduling process by scheduling successive weeks loading back the excess orders to the next week. Such an approach was not necessary.

The sequence of activities within the loop could be replaced by the use of the individual activities as and when necessary. Firstly, `loading back' the order load could be replaced by a record of the level of order overload for each successive week. Re-scheduling functions then need not be restricted to systematically re-scheduling each week in succession. Also the "re-site" function could be restricted to use only when a particular site was a bottleneck causing other sites to be underloaded or itself became underloaded due to lack of orders within an overload situation.

Defining Useful Re-Scheduling Mechanisms

The analysis carried out to identify the implications of replacing the loop also provided one major point of learning. When the design began, the scheduling task had been seen in terms of identifying ways of overcoming overloaded capacity. The task was now seen as how to use any underloaded capacity that existed. Such a stance resulted from the use of routes and markets within the plans. With no underloaded capacity, all routes and markets would produce to the level of the plan. Loss of production or changes in a markets share of capacity for what even reason represented variations from the plan, effectively changes in the plan.

The ability to concentrate on underloaded capacity as an initiator of any re-scheduling mechanism led to the search for such mechanisms. Four such mechanisms were identified which also had been observed occurring within the real-world situation. The first of these was based around the "re-site" mechanism. In the real-world, the most exaggerated examples of this are the `doglegs' described in chapter 2.

Two further mechanisms were termed `trend trading' and `market-led grabbing'. `Trend trading' occurs when one market trades its own under-use of capacity with another market for later capacity when it is overbooked with orders. `Market-led grabbing' occurs when a market suffers from customers changing from one product to another. The shortfall in capacity, if any, often extends to only one extra facility. Real-world examples of such instances resulted in the market concerned gaining preferential treatment.

The final mechanism was the back-stop mechanism. This would identify uses for unused capacity and portion it out to the various routes and markets.

The importance of these different mechanisms and the different reasons for using one rather than another needed to be determined.

Further, such methods could be easily related to the real-world situation. The use of these mechanisms to amend the FWP so as to make use of underloaded capacity was certainly not alien in concept to the actors in the situation and due to this may draw criticism from them. The real-world yielded no explicit scheme for the use of such mechanisms. Obtaining a consensus from these actors as to the correct use of the mechanisms would not be easy. The actors included sales personnel and site personnel as well as most of the staff of the Central Operations department. The author was reluctant to involve such a diverse set of people in a debate concerning the use of the mechanisms so early in the development.

The design of UAAMS thus involved significant soft problems as had the design involved in the conceptual trials as described in chapter 5. However, in this case, the problems so far found did not affect any design commitment and thus could be tackled later in the development. It was envisaged that a provisional scheme involving the four mechanisms already identified would be presented in the first instance to Order Allocation staff. Work preparing for this presentation was expected to involve SSM.

Analysis of the Re-Scheduling Mechanisms

Prior to developing this provisional scheme, the four mechanisms were subject to analysis individually, involving the use of the Intelligent Agent and the `roles' construct. A development of the conceptual model illustrated in figure 6.7 could also have been used but the author was concerned to test the more formal constructs of the problem-driven approach.

The process of analysis was similar for all four mechanisms. The first construct used was the Intelligent Agent which led directly to an

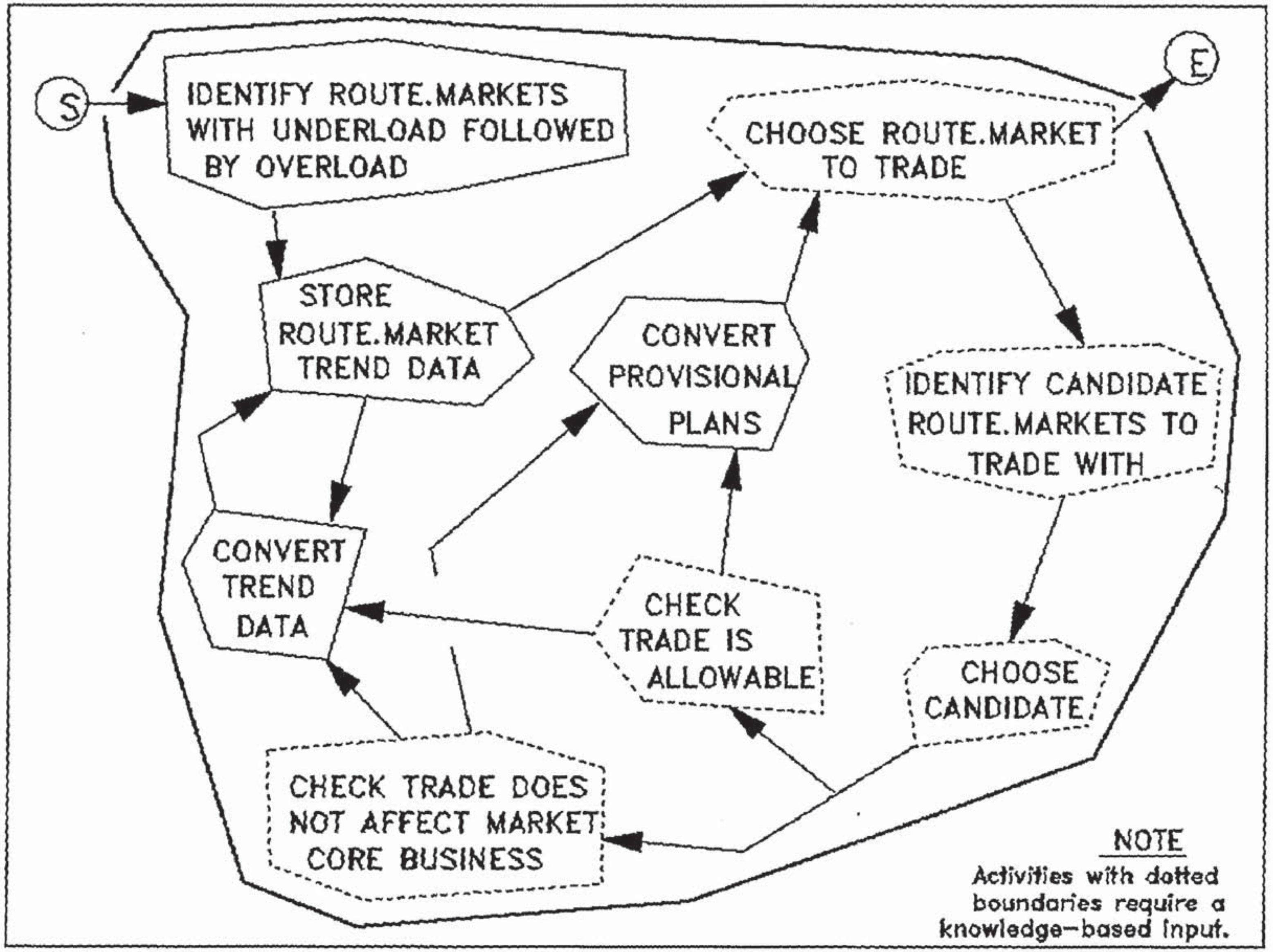


FIGURE 6.9 ACTIVITY DIAGRAM OF THE 'TREND TRADING' MECHANISM

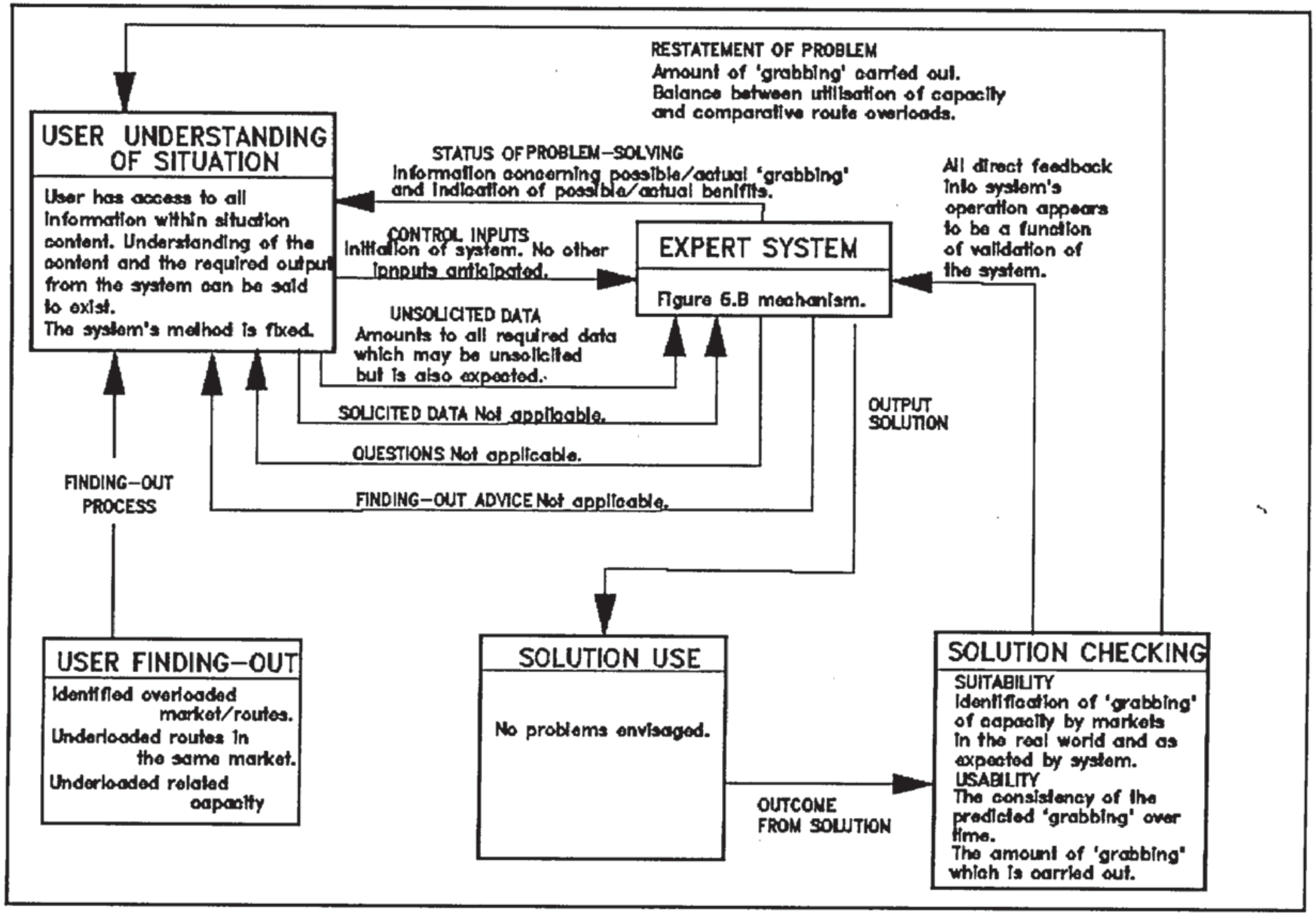


FIGURE 6.10 THE USE OF THE 'ROLES' CONSTRUCT TO ANALYSE THE FUNCTION COVERING THE MARKET-LED 'GRABBING' OF CAPACITY.

initial activity diagram of the mechanism. Figure 6.9 shows the diagram for the `trend trading' mechanism. Within this diagram are many activities involving significant amounts of search and heuristic choice which indicated the use of expert system techniques. The conclusion was draw that these mechanisms were best implemented in Prolog.

The next stage in analysing the mechanisms used the `roles' construct. These developed the design of the different mechanisms very little as the boundary of the designs had been well defined already. However they provided a valuable wide view of the mechanisms. Figure 6.10 provides an example of the use of the roles construct.

The Design of the Scheduling Procedure

The design of the four mechanisms had assumed that the scheduling procedure would use each mechanism in turn, exhausting its use prior to continuing with the next mechanism. This assumption was taken as the provisional scheme for the scheduling procedure. The design of UAAMS had now been developed to a point where feedback from stakeholders in the situation was required. The structure of UAAMS at this stage in the design process is shown in figure 6.11.

The next stage in the development of UAAMS would have been preparation for the presentation of the design to the stakeholders to identify whom to involve in the presentation and what design issues to present to them. This stage was not to be carried out as, prior to this, it was decided to implement part of the scheduling procedure before carrying out this work; namely the "re-site" module. The reasons for this decision are given in section 6.3. Difficulties in implementing this "re-site" module were to terminate the UAAMS development.

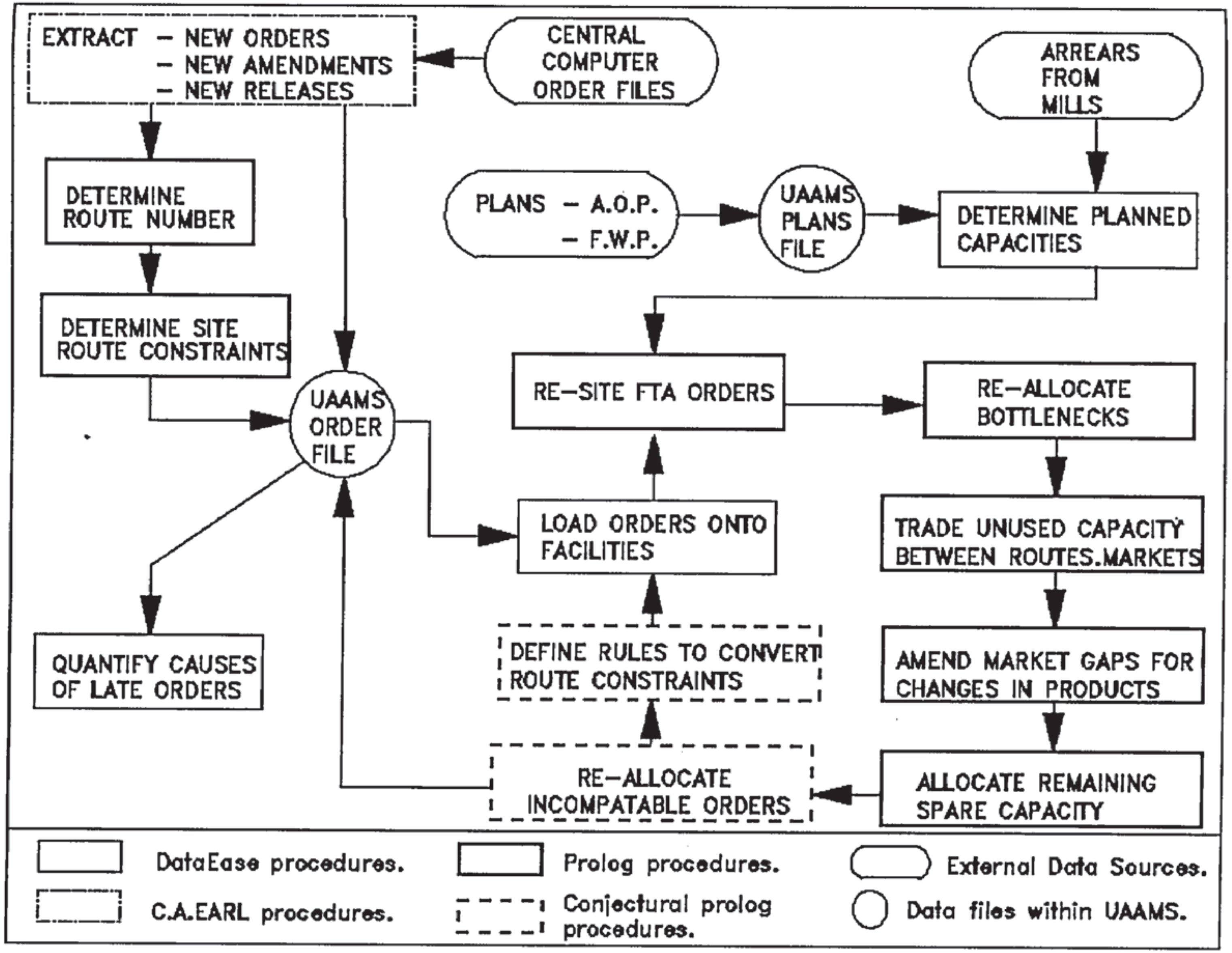


FIGURE 6.11 FINAL STRUCTURE OF THE UAAMS DESIGN.

6.2.6 LESSONS FROM THE USE OF THE PROBLEM-DRIVEN CONSTRUCTS

The constructs of the problem-driven approach had been used extensively within the UAAMS development in the framing of expert system component. A number of lessons were becoming clear following this work.

The first lesson concerned the transition from analysis of a computer solution to analysis of expert system solutions. The inputs into this decision remained poorly defined. The transition itself however marked more a mental decision by the developer rather than any great change in the design process; SSM was being used to analyse the situation both before and after the transition for very similar tasks.

A second lesson from the UAAMS development was that the different constructs were used in the same sequence throughout the development of UAAMS. Firstly SSM was used to analyse the situation followed firstly by the Intelligent Agent and then by the `roles' construct. Often the `roles' construct was not greatly useful. This was more due to the nature of UAAMS than any deficiency in the `roles' construct. One reason for this was that UAAMS was intended to be a rough cut system in which the sophistication often identified using the `roles' construct was not required.

A third lesson concerned the manner in which the UAAMS design, despite relying on `hard' constructs, was able to provide a domain for debate in the uses of the four mechanisms independently from any committed design decisions. This ability to involve stakeholders in the fundamental design of a computer system was seen as being due to the perceived abilities of expert system techniques to cope with any conclusion resulting from such a debate.

The final lesson confirmed an impression formed during the conceptual trial developments mentioned in chapter 5. The development of UAAMS was certainly problem-driven. Yet the extent to which the problem had driven the design was not at all apparent. Much of the design work and many of the design decisions could just have easily been part of an exploratory development. To identify the impact of the problem-driven approach, it was necessary to analyse the uses made of prototypes and models within the design. The findings of this work will be described in section 6.4.

6.3 DEVELOPING THE "RE-SITING" MODULE

This section presents the reasons for developing the "re-site" module as well as describing the function of the "re-site" module, the development of the module and the causes of the difficulties encountered during this development process. Finally, the impact of these difficulties on the development of UAAMS is explained, as are the reasons for this leading to the termination of the UAAMS development.

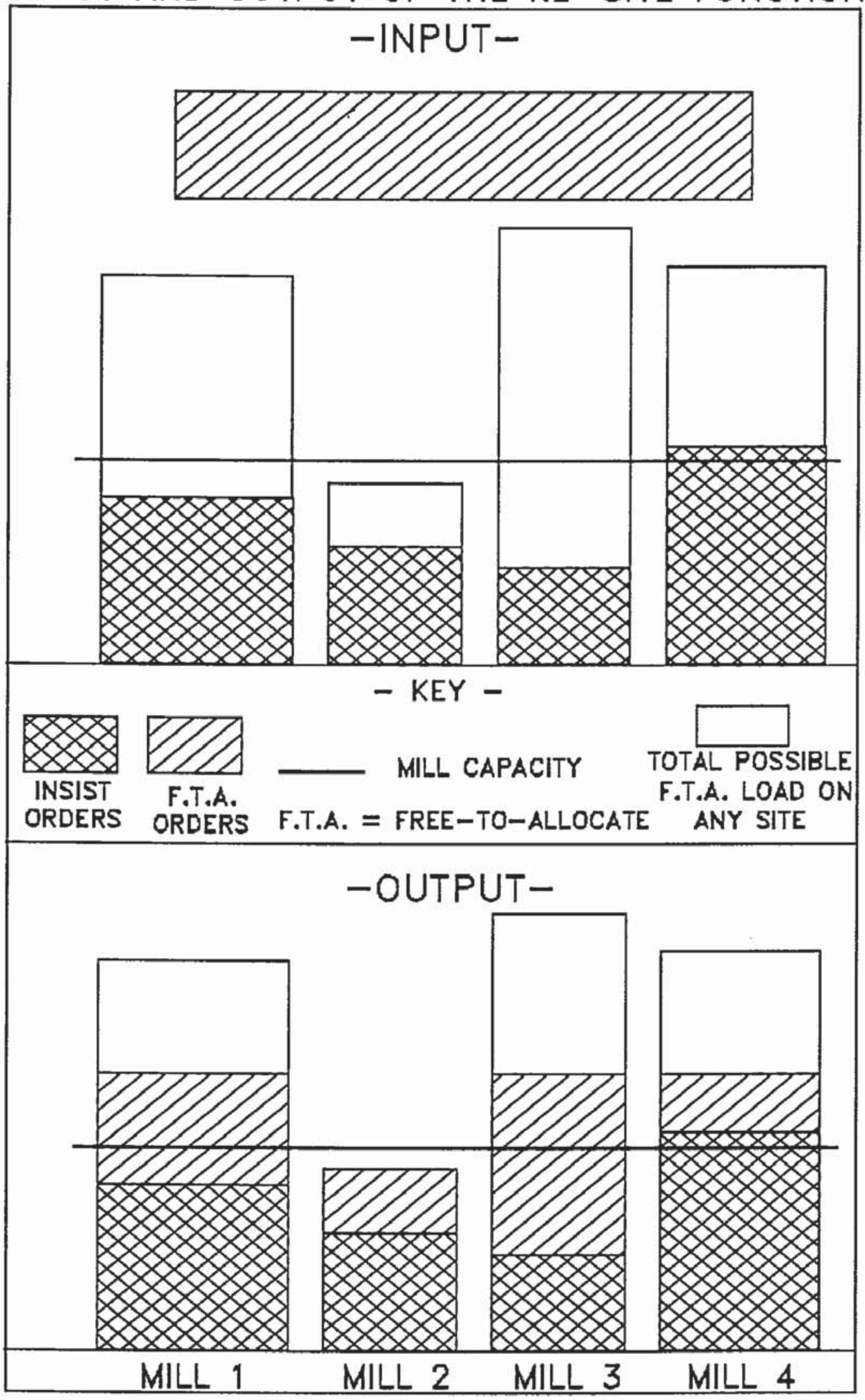
The Decision to Develop the "Re-Site" Module

The decision to develop the "re-site" module at this stage in the design process was prompted by two considerations.

Firstly, the `load capacity' procedure had already been implemented in DataEase and it was thought that this procedure might play a significant role in the presentation of the design to the stakeholders. The `load capacity' procedure did not load `free-to-allocate' (FTA) orders onto individual sites. An implemented "re-site" module would provide a realistic capacity-free FTA load.

A second reason was to enable the estimation of the time required to develop Prolog subroutines. It was considered that time constraints might possibly restrict the development of UAAMS, something which would have to be made known to stakeholders were it found to be the case. The

FIGURE 6.12 DIAGRAM SHOWING THE BASIC INPUT AND OUTPUT OF THE RE-SITE FUNCTION



"re-site" module was not expected to take a significant amount of project time.

The "Re-Site" Function

The "re-site" function is most easily described with an illustration of its input and output (figure 6.12). The load on a particular facility will consist of `insist' and `free-to-allocate' (FTA) orders. As described in section 6.2.1, these could be recorded as the total `insist' and the total possible FTA load on each site together with the total FTA load for all sites. The "re-site" function shares out the FTA orders, loading each site to the same percentage of capacity unless insufficient orders are available.

The initial use of the re-siting function was for loading FTA orders for a particular route onto the planned capacity for each site. Due to this, any interworks transfers would be explicitly defined within either the plans or the routing decisions; impractical interworks transfers would thus be avoided.

Later uses of the module would redistribute FTA orders for a market or a route once unused capacity had been allocated to them. As this was no longer part of the planned use of capacity, impractical interworks transfers could occur. For instance there would be nothing preventing interworks transfers occurring at every stage in a route. Coping with such problems was, in the first instance, taken as a function external to the "re-site" function being developed. However it was expected that internal coordination of the uses of the "re-site" function would be a more efficient method and would probably entail the use of conditional limits for a site's total possible FTA load.

The Development of the Routing Module

The first step taken was the building of an activity diagram of

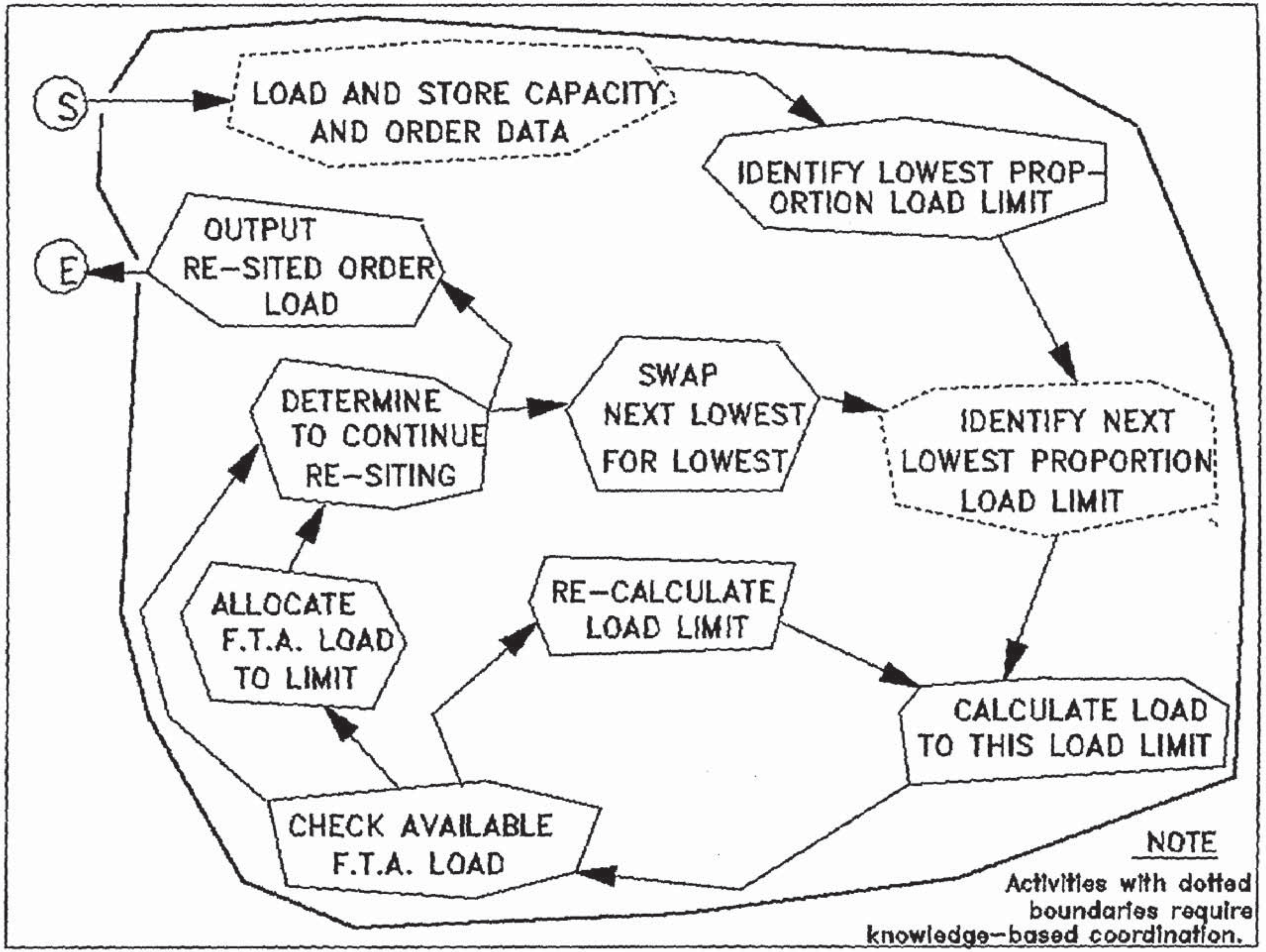
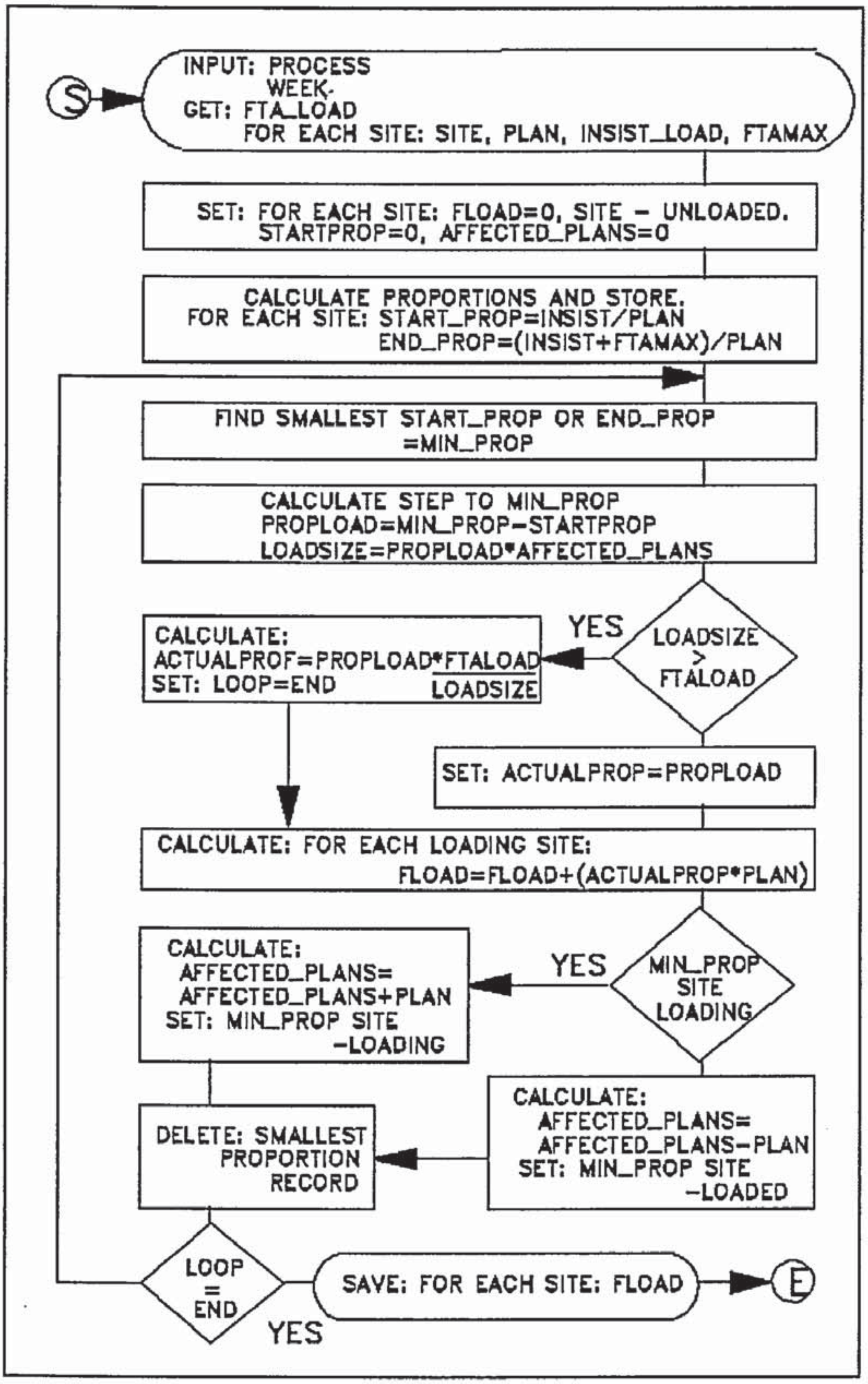


FIGURE 6.13 AN ACTIVITY DIAGRAM OF THE RE-SITE MODULE

FIGURE 6.14 A FLOW DIAGRAM OF THE RE-SITE MODULE'S FUNCTIONALITY.



the module (figure 6.13) which was seen to be well enough developed to allow Prolog coding to commence. Implementing this design however did not proceed smoothly, the reasons for which are discussed below, and a considerable length of time was taken up in the task. To assist in the task, a more exact specification of the module was developed (figure 6.14). While clarifying the task in hand, the development of the module still proceeded slowly.

In due course the module became functional. The code however was considered to be too difficult to understand given that it may become part of a usefully working system. Thus further time was expended attempting to reduce the complexity of the coding with partial success.

Eventually some eight weeks were expended developing the "re-site" module. (The coding of the final version is presented in Appendix B.)

The Reasons of the Difficulty in Developing the "Re-Site" Module

There were three reasons for the difficulty in developing the "re-site" module.

Firstly, the "re-site" module consisted very much of procedural activities. Prolog could not cope with these procedural aspects of the module in a straightforward manner.

A simple linear procedure can be easily developed within Prolog through the use of a relationship listing each procedural step successively as conditional statements. However, the insertion of branching and iterative loops within the procedure requires a more complex and considered use of Prolog functionality. For instance, iteration within Prolog can be achieved by using a FORALL statement, a recursive relationship or by forcing a failure within a clause. The implications of choosing any one of these methods can be profound for a design but such implications defied identification prior to attempting their use.

A second reason concerned the passing of data between Prolog relationships. Prolog has two mechanisms for doing this. The first, instantiation, requires the naming of the data being passed to a relationship when it is called by another as well as the naming of the data being returned back from that relationship. The second mechanism requires the explicit saving of clauses as well as recalling them and deleting them after use. Both these mechanisms can become quite involved as the volume of data being transferred between relationships increases, adding to the complexity of a Prolog implementation. The only method of avoiding their use is to use larger and thus fewer relationships. While larger relationships may reduce the complexity of the Prolog, they also significantly reduce its understandability which was itself another reason for the difficulties in developing the re-site module.

Understanding the functioning of Prolog code was a problem encountered during the development of the Leadtime system described in chapter 5. The Leadtime system was an exploratory development and it was expected that the difficulties encountered would not be repeated when the system being developed was adequately specified. A precise specification was available during the development of the "re-site" module yet understandability was still found to be a major difficulty.

The Termination of the UAAMS Development

The difficulty in developing the "re-site" module drew into question the wisdom of continuing the UAAMS development. Some four months remained of the project and the time available to the project was becoming a major concern. A great deal of time had been spent developing the "re-site" module which itself drew into question the anticipated time required to develop other Prolog modules. Examination of the other designs within the scheduling procedure identified similar procedural activities and uses of data to those of the "re-site" module, two of the reasons behind the difficulty in developing the "re-site" module. While the experience of developing the "re-site" module would allow similar modules to be developed more quickly, the task still appeared too great given the time available.

Prior to terminating the UAAMS development, alternative languages to replace or supplement Prolog were considered as a means of speeding up the development process. However, no immediate solution could be seen which did not involve testing the usefulness of the candidate language and so expending more project time without a guarantee of a viable solution. The languages considered included the procedural language, Basic, as well as the use of Lisp and Poplog.

Given the state of the UAAMS development, two options appeared open. The first was to continue with the development of UAAMS using Prolog. The second was to develop a different system in a more promising field, namely a routing system.

The first option was expected to result in a system with restricted functionality which might even then remain unusable.

The second option appeared more promising and was chosen. The routing procedure within UAAMS appeared a straightforward development and an appropriate routing problem was available in the maintenance of X10, a conventional routing programme. The work that resulted from this decision, the development of MICRO-X10, is described in chapter 7.

One factor which bore heavily on this decision should be stressed. Once completing UAAMS in a usable form became unlikely, further work identifying the existence (or not) of a language more appropriate than Prolog for the development of UAAMS would not have been in accordance with the project remit. Such work would certainly not have led to the completion of UAAMS, either during or following the project. The remit asked not for the advantages of expert systems to be identified. Rather it asked for such advantageous systems to be built.

6.4 FINDINGS FROM THE UAAMS DEVELOPMENT

The UAAMS development identified a number of points of learning. These concern the usefulness of Prolog, the usefulness of expert systems within the domain of PP&C and the use of a problem-driven approach. Other points remained uninvestigated due to the termination of the development.

The Usefulness of Prolog

Lack of understandability of Prolog, identified as a problem during the leadtime development described in chapter 5, was found to be a significant impediment to the development of Prolog functions within a scheduling system, even when the function was adequately specified. This difficulty, coupled with the methods of defining procedural devices and the method of passing data between relationships, appears to make Prolog inappropriate for the development of such functions as the "re-site" function which consist very much of procedural activities.

It was argued in section 4.2.4 that assumption within expert system literature make conclusions regarding the usefulness of expert systems highly tenuous. While a number of authors have criticized

Prolog for its poor understandability (133,268,274,276,357), they do not see this as making Prolog unusable and do not express any resulting limitations for the use of Prolog. Certainly the criticism is not based on experiences developing practical working systems (276). The findings from the UAAMS development thus add significant context-related learning to the literature.

It was not possible to investigate the adequacy of Prolog in comparison with other languages. Literature concerning the practical use of different languages for expert system development remains sparse and poorly defined. As such it is not possible to relate these findings from the UAAMS development to those of other authors. For instance, Kraft states that Prolog is superior to Lisp for certain uses. He states:-

> ...Prolog will be needed for a certain class of problems which are more mathematically oriented (227).

Whether Kraft would consider the "re-site" module to be mathematically or arithmetically oriented is unclear.

The Usefulness of Expert Systems within the Domain of PP&C

UAAMS has drawn into doubt the usefulness of Prolog as a language for building a scheduling system. It is significant that although Prolog has been criticized by some authors for lack of understandability, it has not been explicitly named as an AI language with particularly restrictive uses. This then leaves open the question as to whether any AI language is suitable for the building of expert scheduling systems or whether expert scheduling systems should be built using conventional languages.

The routing knowledge within UAAMS was found to be shallow in nature and apparently well suited to the development of expert systems.

The analysis carried out within the UAAMS development suggests this may be a finding general to routing systems. The development of MICRO-X10, described in chapter 7, allows further investigation of this point.

The remaining area of PP&C covered by UAAMS, plan maintenance, was not a direct concern during the development of UAAMS as UAAMS was required to test this function as a possible cause of unallocated arrears. However, the analysis did identify the task as being less complex than the scheduling task with regard to arithmetical manipulation but less well defined and more subject to `soft' problems encompassing conflicting objectives than the other tasks within UAAMS. This remains however a situation-specific observation rather than a conclusion generally for PP&C systems.

The Use of a Problem-Driven Approach

The development of UAAMS provides an example of the use of the problem-driven approach. The development of the initial design of UAAMS was expanded and redefined as expected. While this process, broadly a linear development, was much as expected, at a detailed level, the design could be described as a `messy' (71) learning process assisted by the problem-driven constructs. Given that the subject of the design, the rough-cut allocation task, was poorly understood, this is understandable.

One aspect of the problem-driven approach concerned the building of the initial structure of UAAMS. This was a speculative structure and incorporated explicit assumptions. As such this initial structure was an EXPLORATORY model of the system. Further it did not model the wider system within which UAAMS was expected to operate. Such a use of models was in keeping with the type of system being developed, namely poorly defined and, as a monitoring system, relatively remote from its wider system.

The design work in expanding the EXPLORATORY model included the use of EXPERIMENTAL models (for instance, the routing model in figure 6.4) and further EXPLORATORY models (the Intelligent Agent comparison in figure 6.3) and in so doing amendments were made to the initial EXPLORATORY model allowing it to be recast as an EXPOSITORY model. From this point, the process continued to develop EXPERIMENTAL and EXPLORATORY models. Indeed, the "re-site" module was such a model in that it was developed to identify the speed with which Prolog modules could be developed. Other types of model were not used due to the termination of the development.

This use of models differs in two ways from the main stream, exploratory approach. Firstly the initial EXPLORATORY model was not a functional, vertical prototype but a broad-level description of the system. Secondly, the EXPERIMENTAL and EXPLORATORY models next developed were intended to test the assumptions, both explicit and implicit, present within the `main' EXPLORATORY model to allow it to be recast as an EXPOSITORY model. Within the main stream approach, EXPERIMENTAL or EXPERIENTIAL models are developed to gradually identify the required functionality of the finished system. However, such models tackle the easier aspects of a system's functionality first (276) and, as they are integral with the `main' EXPLORATORY model, an EXPOSITORY model is never developed.

<u>Areas of Research Left Unexplored by the UAAMS Development</u>

The termination of the UAAMS development had left a number of areas of the problem-driven approach unexplored.

Firstly, although data from DataEase had been used within the re-site module, the use of DataEase as a means of building a user interface and database for an expert system had yet to be demonstrated.

A second area concerned the effects of a problem-driven approach on the implementation of expert systems. This was not covered by the UAAMS development at all.

A third area concerned the maintenance issue and the effect of the requirements imposed on an expert system design by the need for maintenance within the domain of PP&C.

A fourth area concerned the abilities of the problem-driven approach to support the detailed design of an expert system. While two parts of UAAMS had been implemented, the "re-site" module had represented the only expert system and was not considered as representative of possible expert systems throughout the domain of PP&C.

The UAAMS development had identified the routing knowledge required by it to be `shallow' in nature. The development of a separate routing system allowed the nature of routing knowledge to be investigated further. It also provided a second chance for the areas remaining unexplored by the UAAMS development to be investigated.

CHAPTER 7
THE MICRO-X10 DEVELOPMENT

This chapter describes the development of a second expert system called MICRO-X10. The reasons for developing this system are presented first, followed by a description of X10, the conventional programme which MICRO-X10 was intended to replace. The process by which MICRO-X10 was developed is then described followed by the tests carried out on the system. A final section presents the conclusions drawn from this work.

7.1 THE DECISION TO DEVELOP MICRO-X10.

With the development of UAAMS terminated, a number of areas of the research remained unexplored. These were as follows:-

(i) The use of DataEase as a means of building a front-end for a knowledge-base written in Prolog.

(ii) The use of the constructs of the problem-driven approach to assist in implementation and to cope with the associated soft problems.

(iii) The development of a detailed design using the constructs of the problem-driven approach to aid the design itself.

(iv) The scope of the maintenance issue.

(v) The usefulness of expert routing systems.

The decision to terminate the development of UAAMS marked the end of the truly problem-driven approach. While some three months remained of the project, the possibility remained for the development of some other expert system to further test the usefulness of expert systems within the domain of PP&C and the usefulness of the problem-driven approach within such a development.

Obtaining agreement quickly from the project team for a new development was not possible due to holidays. In considering likely developments, one stood out as a prime candidate. This was the development of a replacement for the programme X10.

The development of such a system appeared attractive. The Order Allocation staff were enthusiastic about the possibility of replacing X10 with a more understandable system. X10, after all, played a significant role within Order Allocation. Further, the author was conversant with much of the content of X10 and it was evident that much of the knowledge contained within X10 was also applicable to other tasks.

Agreement was obtained from available members of the project team to develop an expert system version of X10. It was to be known as MICRO-X10 as it was to be developed to run on a micro-computer.

7.2 THE NATURE OF X10

This section presents a description of X10 and its role within the Order Allocation task.

The Role Carried Out by X10

X10 is one of a number of programmes which the allocators run every morning to check the orders on the central order files after the overnight file updates. Its function is to identify orders with production leadtimes incompatible to the leadtime rules. That is, X10 checks that an order's promised delivery week is compatible with its promised hot rolling week. This is defined by the same leadtime rules which were the subject of the trial expert system development described in chapter 5. X10 does not however check that the hot rolling week is allowable given the timing of order acceptance.

The output from X10 then is a listing of the unallocated orders which have an incompatible combination of hot rolling and delivery week. X10 provides an allowable hot rolling week for these orders based on the promised delivery week, as well as showing the reasons behind the required leadtime.

An order may appear in the X10 listing simply because the leadtime is incorrect. However it may also be that the order is knowingly being allowed to break the rules due to some operational reason. Thus the orders listed by X10 are checked both by allocators and by the personnel responsible for entering the orders into the central computer files.

X10 plays an important role within Order Allocation. The promised hot rolling weeks are used in Order Allocation as the basis for timing the release of orders. X10 is the only check that the production leadtimes and thus the promised hot rolling weeks are correct.

The Functionality of X10

X10 has encoded within it the same leadtime rules as were the subject of the trial expert system development described in chapter 5. It is concerned only with the production leadtime. That is X10 checks that the promised hot rolling week is compatible with the promised delivery week. It is not concerned with the timing of order acceptance and any allowable promised delivery week resulting from that.

While X10 covers fewer rules than the trial leadtime development described in chapter 5, it expands many of the rules to a far greater level of detail.

X10 is a programme written in C.A.EARL, a database enquiry language. It consists of some one thousand lines of code. The task carried out by X10 can be divided up into sub-tasks as follows, the number of lines of code involved in each sub-task giving a rough guide to their comparative complexity:-

Defining data fields in central computer order file (80 lines).

Ignoring non-applicable orders (10 lines).

Identifying product type (30 lines).

Identifying `free-to-allocate' and `insist' orders (50 lines).

Calculating the effect of works holidays (70 lines).

Identifying inspection requirements (210 lines).

Identifying attributes which affect leadtime (150 lines).

Calculating suggested hot rolling week (100 lines).

Identifying wide orders (80 lines).

Identifying orders knowingly breaking the rules (150 lines).

Printout definition (70 lines).

Of these sub-tasks, the identification of the product type and the identification of inspection requirements are carried out by subroutines which are also accessed by other C.A.EARL programmes. Beyond this there is no structure imposed on X10.

One significant point to make is that X10 is not documented. The only documentation is the listing of the programme.

Maintenance of X10

The most frequent maintenance consists of maintaining the list of orders known to be breaking the rules. Other routine maintenance consists of updating `wide weeks' and works holidays. All allocators are capable of updating X10 in this manner.

Beyond this routine maintenance are more profound maintenance tasks which include adding exceptions into the rules, or amending the clauses of the rules themselves. Such maintenance is carried out by only two of the allocators.

The rules themselves sometimes undergo major amendments in structure. During the time between the trial expert system development and the decision to implement MICRO-X10 the leadtime rules had been modified quite markedly. X10 had been modified such that the rules extending the leadtime could do so in increments of half weeks rather than just whole numbers of weeks. A new set of rules then decided whether a leadtime should be rounded up or down when the calculated leadtime was not a whole number of weeks. Roughly, judging from past experience, the rules, and as a result X10, undergo such major amendment every two years or so. There is just one allocator who carries out such work.

An Example of the Complexity of X10

The majority of the functionality of X10 can be understood by considering an example. There are currently fifteen attributes which affect the leadtime rules in a standard way. The following code is used to identify one of these attributes.

```
507  REF65: GOTO E51 QYDC3 "G" "H"
508  GOTO REF6
509  E51: SET C=5 SET C9=3
510  REF6:
```

This code determines whether the order is for Vitreous Enamel Finish strip. If this is so, the variable C9 is set to identify an increase in leadtime of three weeks which is required by such orders.

One point of note concerns the variable C on line 509. This variable is actually not used by X10. Despite this, the variable remains within the coding as the personnel maintaining X10 have not the time to check properly whether or not the variable is required. This variable remains within the code of X10 some two years after the programme was altered making it unnecessary.

The example above is the most simple of examples from X10. The following example is more typical:-

```
REF6: GOTO F6 PRODUCT "C.R.COIL" AND QYDC3 "A" "B" AND
QYDC7 "T" AND CUST NOT "F0522" "F0523" "F0524"
GOTO F6 PRODUCT "GALV PLN" "ZINTEC" AND QYDC7 "T"
GOTO F6 PART "AFP 0830" "AFP 0906" "13269/0370-1"
GOTO F6 PRODUCT "C.R.COIL" AND CUST "S2683" "S2674"
AND GAUGE <0.41
GOTO F6 PRODUCT "C.R.COIL" AND PRPS "01" AND WIDTH
<1000 AND CUST NOT "V0223"
GOTO F6 CUST "R0249"
GOTO F6 PRODUCT "C.R.COIL" AND CUST "D0282"
GOTO REF7
F6: SET F=5  SET F9=0.5
REF7:
```

This coding identifies orders for `side trimmed last' (QYDC7 = "T") cold reduced coil of General Purpose Finish. However identifying orders for which this rule applies is complicated by exceptions. These exceptions include such things as the customer's part number and the `end use' of the steel. (PRPS "01" identifies the `end use' as drum manufacture). The complexity of this example is quite typical for X10 generally.

Other Considerations Concerning the Functionality of X10

The examples given above of the complexity of X10 can be seen to be a set of discrete rules which determine whether the leadtime should be extended by some fixed amount. They include no procedural devices. A number of other points concerning the functionality of X10 are not demonstrable through such examples.

One point not emphasized above is that many of the rules which increase an order's leadtime are interdependent. Further, such interdependence is not always made evident in the same manner.

The second example above concerned `side trim last' General Purpose Finish cold reduced coil. The rules do not mention the effect on leadtime of `side trim last' for Full Finish or Vitreous Enamel

Finish coil. This is because such finishes already cause an increase in the leadtime which is capable of absorbing the increase due to the `side trim last' operation. The two examples given above are thus interdependent.

The Vitreous Enamel example above is also more explicitly interdependent with the effect of inspection requirements. The extra week normally required for inspection is not required for Vitreous Enamel Finish cold reduced strip and indeed X10 cancels this extra week for inspection. This is because Vitreous Enamel Finish strip always requires inspection. Thus the three extra weeks in the Vitreous Enamel leadtime include a week for inspection.

This interdependence between rules can be considered to be due to the lack of differentiation made within X10 between different types of clause within the same rule. X10 does establish the product type separately from other considerations. However other attributes of an order, for instance quality, grade or finish, are established separately within each rule identifying reasons for an increase in leadtime. That the distinction between order attributes and increased leadtime is not made explicit is due to a number of reasons. One reason is that there is no standardized structure for such attributes. Any structure imposed within such a programme as X10 may be hastily derived or ill-thought-through. Another reason is the nature of the numerous exceptions. While some of these exceptions are identifying such things as finish or grade, others simply reflect custom and practice at the works.

In the main X10 has avoided the use of procedural devices. That is the order in which the different clauses occur within a rule do not affect the output from that rule. This perhaps suggests that the rules within X10 are easily encoded. Certainly much of X10 would be described in expert system terms as consisting of shallow knowledge.

Some sub-tasks of X10, however, cannot be described as shallow and do involve procedural devices. All the overtly temporal considerations made by X10 are of such a form. These include coping with holiday weeks, wide weeks and year ends. Also of this form is the manner in which X10 copes with order feasibility at different sites. The functionality of X10 in these instances could be described as limited. As an example, within X10 order feasibility is considered as being either unique to a particular site or feasible on all sites.

The structure of X10 then is not easily understandable. Perhaps the main failing, and the reason for the reputation that X10 has within Order Allocation as a complex programme, is that the reasons for the various clauses within X10 are not apparent from the listing which is the only documentation.

7.3 THE DEVELOPMENT OF MICRO-X10

The description of MICRO-X10 development is divided into two sections covering firstly the initial design work and then describing the detailed design work.

7.3.1 THE INITIAL DESIGN OF MICRO-X10

This section describes the development of the system requirements for the MICRO-X10 system. The purpose of MICRO-X10 is first discussed followed by consideration of alternative solutions which provided justification for the development of MICRO-X10. Following this work, the possible expert system components of MICRO-X10 are analysed and the need for a prototype system identified. Implementation issues are then analysed. Finally the requirements for the prototype system are presented.

The Purpose of the MICRO-X10 Development

While in section 7.1 above MICRO-X10 was described as an expert system version of X10, its development could ignore neither the reasons for wanting to improve on X10 nor alternative methods of improving it. In this manner the development of MICRO-X10 can be seen as problem-driven.

The first consideration then was the reason for wanting to improve on X10. The deficiencies of X10 expressed by allocators had but one root. Simply the deficiencies were due to the complexity of the programme which prevented understanding of the functionality of X10, which in turn made system maintenance a problem.

Consideration of Alternative Solutions

The development of MICRO-X10 started with the use of SSM to identify the reasons for the perceived complexity of X10 and thus the difficulties encountered during its maintenance. Two separate problem areas were identified as well as three mechanisms to counter such problems.

The two problem areas associated with maintenance were found to be understanding the programme's functionality and checking the correctness of any amendments made. The three mechanisms were:-

(1) Rationalization of the programme in terms of the inputs which are required during maintenance.

(2) Structuring the programme to facilitate understanding.

(3) Automating the task of documentation and organization of the programme.

These three mechanisms were considered in terms of three possible ways of improving the X10 `problem situation', namely providing X10 with documentation, rewriting X10 (but still using C.A.EARL) and developing MICRO-X10.

The first possible improvement, providing X10 with documentation was not seen as improving the situation in the long term. Providing X10 with documentation was seen as a means only of extending the number of personnel capable of amending X10 through explaining the functionality of X10 to them. In the long term the documentation, unless maintained, would become less and less useful. Maintenance of the documentation was considered as problematical as maintaining X10 itself.

The second possible improvement, rewriting X10, was also not seen as a long term solution. Rewriting X10 to rationalize the maintenance inputs and to provide a programme structure required assumptions to be made concerning the future amendments to X10. While documentation of a rewritten X10 would be more durable than documentation of the current programme, a rewritten X10 could also be seen as substituting the old programming style with a new style which in the long term might not be appropriate to the task.

The third possible improvement, the development of MICRO-X10, could be seen as overcoming the problem of programme style by providing an improved programming environment. It would also enable the third mechanism to be realized; self-documentation and organization. Beyond X10, the usefulness of MICRO-X10 in carrying out other analyses of the orderbook by extending its functionality could not be ignored. On the negative side, MICRO-X10 would represent a revolution in computer use within Order Allocation. The effects of such a change could not be taken lightly.

A fourth way of improvement was dismissed out of hand. The possibility of supporting the maintenance of X10 by using an expert system to analyse the programme was not seen as practical without micro-computers directly connected to the central computer being available within Order Allocation. At the time of this decision such a situation was not anticipated.

The analysis of the task of maintaining X10 then suggested that the development of MICRO-X10 was the most desirable long term `solution'. As to its feasibility, the cultural feasibility appeared to rely strongly on the technical feasibility of the different aspects of its functionality.

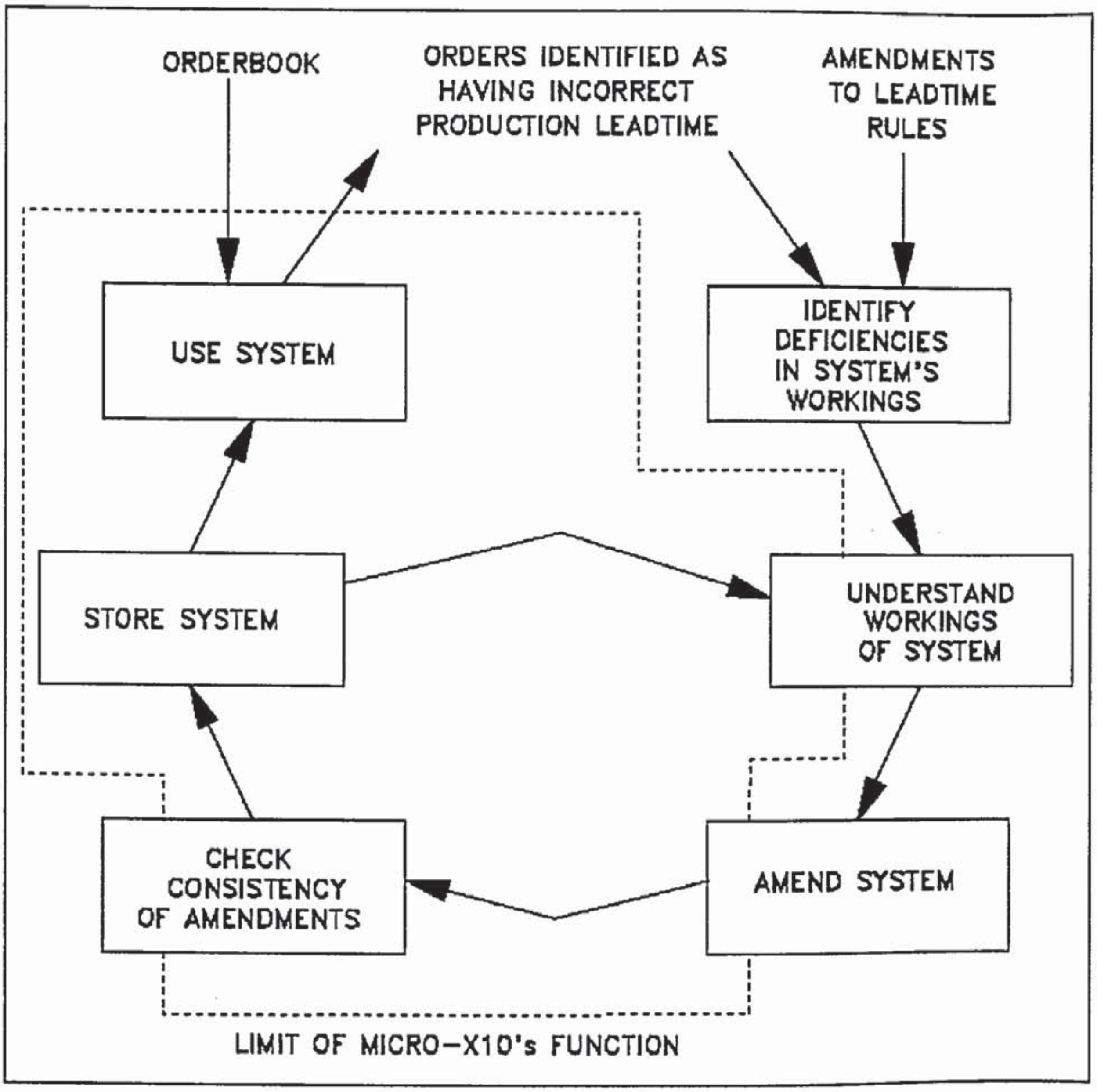
The use of SSM in this instance was non-participative. There was no involvement of actors from the problem situation during the analysis and the identification of feasible change, stage six of the methodology, was effectively by-passed. Both these aspects of non-participation were conscious decisions based on one consideration. At this stage in the development there was little real understanding of the technical feasibility of much of the proposed functionality of MICRO-X10. This use of SSM confirms to the `Mode 2' use of SSM as described in section 3.3.2 above.

Framing the Expert System Components

Consideration of alternative solutions allowed the task of MICRO-X10 to be understood sufficiently for an initial structure, albeit a simple structure, to be developed. This structure is shown in figure 7.1. The task was further analysed using the Intelligent Agent construct (see section 5.3.3) on two areas of this structure, specifically the maintenance and the amendment checking activities.

This analysis yielded two significant points of learning. Firstly the support required by a user to facilitate maintenance was dependent

FIGURE 7.1 THE INITIAL STRUCTURE OF MICRO-X10



on a number of unknowns. A minimum level of support could be seen as the provision of a straightforward rule-based expert system. How useful such a system would be and how much extra functionality would be required appeared to depend on a number of factors; the complexity of the system, the abilities of the user, the familiarity of the user with the system, the complexity of the maintenance being undertaken.

Given this however, it appeared that the design of such extra functionality would be straightforward and flexible. Firstly the extra functionality appeared independent of the minimal system. Secondly the extra functionality could be provided at differing levels of sophistication.

Analysis using the expert system Roles construct (see section 5.3.3) strengthened this view. The mechanisms used within the extra functionality were found to be `raw search'. Generally these mechanisms were independent of each other. The multiplicity of design options greatly complicated the design process. The complexity was such that it was concluded that the user requirements for extra functionality could not then be determined through analysis alone. This first outcome from the analysis suggested the need for a prototype, possibly of the minimal system, to identify the user requirements.

The second point of learning concerned the structure of the rules used within X10. Within a rulebase, the structure is defined simply by the rules used. While an artificial structure superimposed onto the rules would aid understanding of the rulebase, such a structure differed from the logic lying behind many of the rules. An example of such logic is the list of surface finishes appropriate to cold reduced strip; General Purpose, Full Finish, Vitreous Enamelling. Other surface finishes are appropriate to other products.

Developing a logical structure for the rules at first appeared attractive. Work developing such a structure however soon showed that there was no single logical structure. Most evident was that a logical structure depended on the viewpoint taken. As an example, hot rolled strip can be produced either pickled or unpickled. The attribute `pickled' can be taken as a SURFACE FINISH or as a PROCESS carried out on hot rolled coil. It is also considered in some contexts as an individual PRODUCT. Another complication stemmed from the exceptions within X10 as described in section 7.2 above.

Once the difficulties of developing such a structure had been identified it became very evident that a logical structure was not the only means of structuring a rulebase. Other means included explanations for individual rules as well as superimposed artificial structures. Further, providing a structure for the rulebase could then be seen as simply one form of support for a user to facilitate maintenance. Indeed, structuring an unstructured rulebase in such a manner was seen itself to be a maintenance task. In this light, the structuring of an unstructured rulebase was seen as a good test of the maintainability of the rulebase.

In summary, the analysis of the task of MICRO-X10 led to the conclusion that a simple prototype rule-based expert system should be built to allow the user requirements with respect to maintenance to be identified. Also meeting such user requirements was seen to be generally independent of the simple prototype developed. Further, building the X10 rules into the simple prototype would itself provide insight into the maintenance requirements.

At this stage in the design of MICRO-X10 it was possible to define the boundary of the computer system. This is shown in figure 7.1. One of the six activities in this diagram, `Identify Deficiencies in System's Workings', lay totally outside the computer system. Two other activities, `Use System' and `Store System' lay totally within the system. It was these two activities which were to be developed as the simple prototype described above.

The three remaining activities straddled the boundary of the computer system. Identifying the computer requirements of two of these activities, `Understand Workings of System' and `Check Consistency of Amendments', was the reason for developing the prototype. The final activity, `Amend System' implicitly involved the computer system. Computer assistance in automating aspects of the amendment process was identified during the analysis but was considered secondary to the development of other parts of the computer system.

<u>Consideration of Implementation Issues</u>

The decision to develop a prototype to identify user requirements regarding maintenance of the system led to consideration of the use of this system within the problem situation and thus consideration of implementation of both this prototype and of MICRO-X10 generally.

The main difficulty with implementing MICRO-X10 was that it would be replacing an existing, functioning system. Further it would be using a micro-computer to carry out this task rather than the central computer. This would require the downloading of data from this central computer onto the micro-computer, a process until then unused within Order Allocation.

The advantages which were hoped to be realized through the development of MICRO-X10 were twofold. The first was the simplification of the task of maintenance allowing more personnel to carry out the maintenance. Secondly, the analysis of X10 had identified areas in

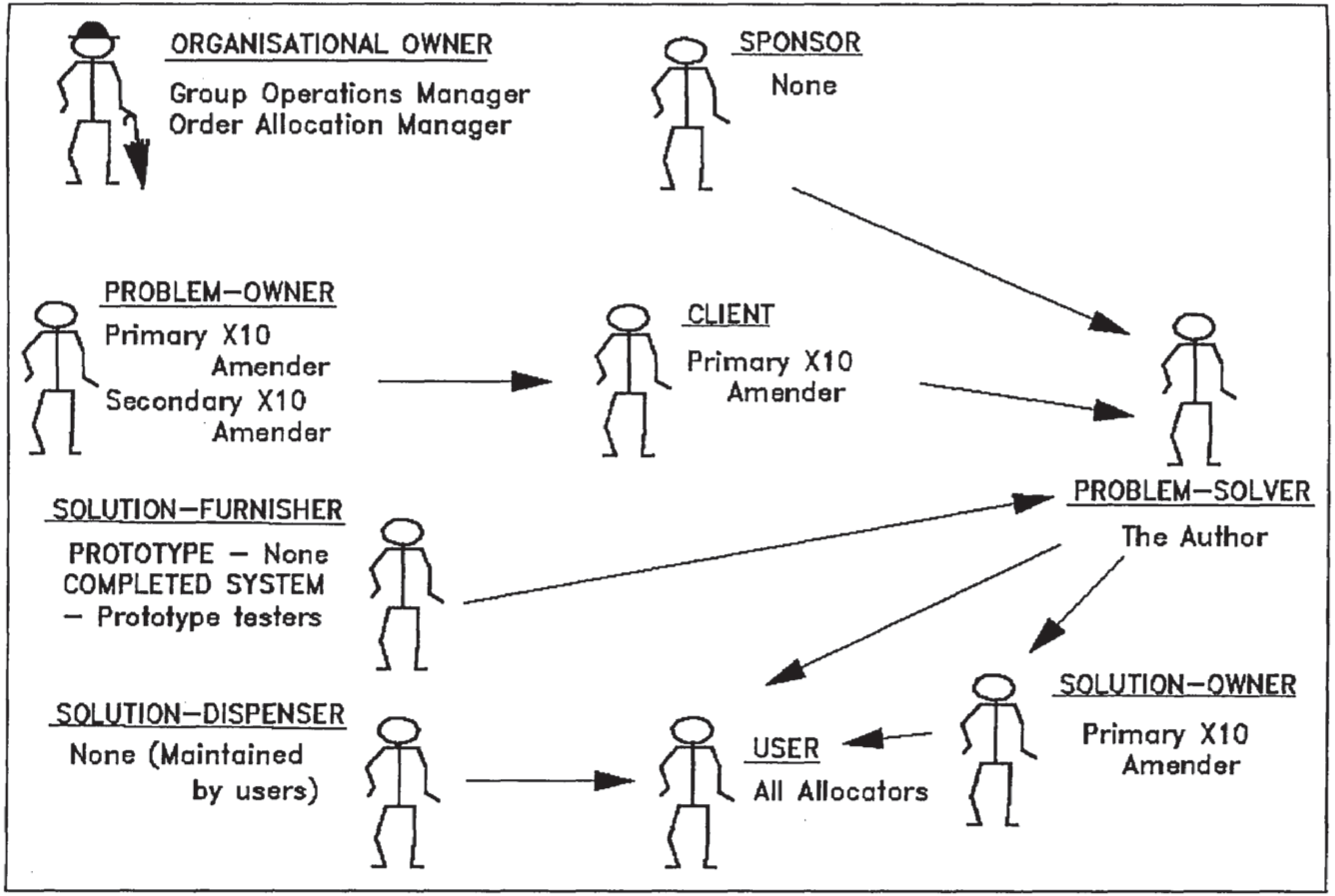


FIGURE 7.2 THE ACTORS WITHIN THE SITUATION AND THEIR ROLES WITHIN THE MICRO-X10 DEVELOPMENT.

which its functionality could be described as limited. Overcoming such limitations in functionality could be seen as another advantage of MICRO-X10. The possibility of extending MICRO-X10 to carry out other tasks and to replace other programmes as well as X10 was seen as being dependent on the success of MICRO-X10. As such it was not seen as a direct advantage of MICRO-X10 at the outset.

The success of MICRO-X10 would then depend on the views taken by the actors involved in its development, use and maintenance. Some actors would be expected to assist in identifying the maintenance requirements. However the immediate question was whether any of them should be involved in the design of this prototype system and if so, to what extent.

The actors involved were mapped onto the set of actors important to a computer implementation. The results of this mapping is shown in figure 7.2. Of the five allocators it will be recalled that one carried out all major amendments to X10 (named the Primary Amender), another was capable of amending and adding clauses to the X10 rules (named the Secondary Amender), while the others carried out no more than routine updates. At this time in the project the position of Central PP&C Manager had disappeared. Thus the only other actors involved were the Order Allocation Manager and the Group Operations Manager.

The most notable point to be seen in figure 7.2 is the lack of Sponsors. Neither of the Organizational Owners were conversant enough with X10 to hold the role as a sponsor of X10's replacement. While they could have taken the role as a sponsor of the development of a new type of computer system, the author was reluctant to offer such a role to them. The author was reluctant to overstate the case for MICRO-X10.

Another point resulting from analysis of the roles of the actors concerned the various roles taken by the Primary X10 Amender. It was he, with the Secondary X10 Amender, who took the role of Problem-Owner. Should X10 become inoperative in any manner it would be he who would overcome the problem. It is unlikely that any other alternative would be considered. Certainly the author knew of no such occurrences. The Primary X10 Amender also took the role of Client. He had agreed that the development was worthwhile. However he was also seen as a strict client. Any difficulties with the operation of MICRO-X10 would have to be amply compensated by improvements. Realistically, the Primary Amender would also probably become the Solution-Owner. Certainly in the short term, he would not allow X10 to lapse until MICRO-X10 had proved itself to his satisfaction.

It was evident that the Primary Amender was not interested in the design of MICRO-X10 unless it had shown itself to be a useful system. He was not enthralled by the mechanisms of Prolog and appeared unprepared to try to understand it unless it proved to be useful. This stance did not extend to the other allocators. However their understanding of the prototype and its purpose was not seen as necessary during the development of the prototype. The possible exception, the Secondary Amender, was rather too busy to involve significantly.

The decision was then made to complete the prototype system as a simple rule-based expert system prior to introducing the system to any of the actors in the situation.

The Requirements of the Prototype MICRO-X10 System

The system requirements for the prototype consisted fundamentally of a sub-system to `store' the rules from X10 and another sub-system to `use' these rules to provide an output conforming to the X10 output.

The `store' sub-system would be developed in DataEase to provide a usable method of updating the rules. The `use' sub-system would be run from DataEase but harness Prolog for the running of the rule-base.

Further system requirements, for instance that the output from Prolog would be stored in DataEase prior to output to the user, had been developed. However these were considered more as design decisions rather than system requirements at this stage.

The system was to be implemented on an IBM Personal System/2 Model 80. This had 80 Megabytes of disc storage space which was divided into three separate discs due to the 30 Megabyte limit for disc size imposed by the operating system. This meant that the DataEase database had to be accommodated on 30 Megabytes of disc. This was believed to be adequate. A maximum limit of one hundred thousand live orders allowed a record size of 300 bytes which was easily adequate given the use of relationships within DataEase to normalize the database.

Another limitation was in the memory available within Prolog. This required that orders be loaded into Prolog in small groups rather than all at once.

Beyond the central system requirements was the need for the rules to be maintainable. In this the system needed to be understandable enough to allow any allocator to amend, add and delete rule sets within the system.

The means by which such maintainability would be achieved would be determined through both the development of the prototype system itself and through trial use of the prototype system by allocators. It was expected that the completed prototype would provide some support for the maintenance task beyond its form as a rule-base. Such extra functionality would thus indicate that the system was capable of expansion if not indicate that it was expected to be expanded.

7.3.2 DETAILED DESIGN OF THE MICRO-X10 PROTOTYPE

This section describes the development of the prototype MICRO-X10 system. The first two sub-sections describe the design of a simple rule-base and the method by which it was harnessed within Prolog. The next two sub-sections describe developments of this initial design both during and after the rules had been written into the rule-base. Two further sections describe the scope of the completed prototype and reflect on the process through which it was developed. A final section describes the reasons why MICRO-X10 failed to become a working system.

The Initial DataEase Rule-Base

The analysis had previously shown that the knowledge within X10 was effectively `shallow' in nature. Further, the subdivision of any rule-base into subtasks had been seen as a function of understanding the rule-base. The conclusion from these points was that the rule-base could be configured as a simple rule-base containing IF/THEN statements which could then be harnessed by the backward chaining mechanism of Prolog. Including explanations within the rule-base within DataEase was seen as providing additional functionality which could be added at a later date if seen to be necessary. In a similar manner, a record of the uses of the rules could also be added later.

The initial design of the DataEase rule-base is shown in figure 7.3. Included within this design was a description field as well as fields intended to indicate the task for which the rule was used although these inclusions were not the result of any serious design work.

The major considerations regarding the design of the rule-base within DataEase were quite mundane. These included deciding such things as field sizes for attribute names and attribute values, the number of

FIGURE 7.3 THE DATAEASE RULE-BASE AS ORIGINALLY DESIGNED SHOWING AN EXAMPLE OF A RULE.

------- RULE RECORD -------

No 0122

USE OF RULE 09 IDENTIFY WIDE/NARROW ORDERS

WIDER TASK 05 DETERMINE PROCESS REQUIREMENTS

DESCRIPTION Identify NARROW orders.

BREADTH eq NARROW

IF

		WIDTH	less	710		
	(or	QYDC2	isin	NARROW_PRODUCTS		
	(	PRODUCT	eq	SHEET		
		PROCESS	not	RESHAPE_RESHEARED	)	
	)	WIDTH	less	SITE_WIDTH		

clauses allowable in each rule, the allowable logic through which such clauses could be combined in each rule, etc,.

The functions which could be used within each clause and the logic combining the various clauses were not text fields but `choice' fields. Illustrated in figure 7.3 are four of the functions which could be chosen; equal to, not equal to, less than, and `isin'.

There were eventually sixteen such functions within MICRO-X10. Of these, nine could be called `comparative' functions; equal to, not equal to, only equal to, more than, less than, more or equal to, less or equal to, `isin', not `isin'. Other functions were arithmetic; plus, minus, multiply, divide. The remaining three functions concerned the number of `solutions' available, the sum of the values of such `solutions' or the sum of the values of unique `solutions'.

Beyond these functions it was also possible to specify the use of Prolog functions from within the rule-base. The completed MICRO-X10 twice made recourse to such functions, specifically INT and STRINGOF.

The values expressed in the rule-base could be in the form of a specific value, the name of the attribute holding that value or the name of a Prolog variable defined elsewhere in the rule. The arithmetic functions obviously require a third operator which was simply squeezed into the value field. In some instances attribute values included what was really an extension to the attribute definition. As an example of this, `EXTRA_WEEK eq NARROW 1' indicated that one extra week was required due to the order being `NARROW'.

Two aspects of the MICRO-X10 functionality could not be contained within the IF/THEN format without either a large increase in the number of clauses in each rule or the inclusion of many more rules in the rule-base. The first of these aspects appeared in the rule-base as an `isin' clause. The `isin' function involved a separate `membership' file. Such clauses would be true if the value of the attribute named was included in the referenced record. The second aspect of functionality did not appear within the rule-base at all. This involved a `decode' file and allowed codes used within the central computer order database to be decoded. In its final form, MICRO-X10 contained twenty-seven membership and decode records. Had the functionality of MICRO-X10 been restricted to the IF/THEN format, these twenty-seven records would have required some 150 rules to encode, significantly increasing the size of the rule-base and the effort required to maintain it.

As well as membership and decode files, files were also developed to contain works holiday details and wide weeks rather than encoding such information within the rule-base.

Harnessing the DataEase Rule-Base

The first step in harnessing the rules under Prolog required the export of the rules from DataEase. It was necessary for the format of the text file thus generated to be compatible with Prolog. This involved the conversion of each function available within the DataEase rule-base into valid Prolog clauses. This was easily achieved using a DataEase `report' to enclose each clause in brackets and move the function name to the front of the clause. Rule 0122 shown in figure 7.3 would thus appear as follows:-

```
((eq BREADTH NARROW) (less WIDTH 710)
                     (or (isin QYDC2 NARROW_PRODUCTS)
                         ((eq PRODUCT SHEET)
                          (not PROCESS RESHAPE_RESHEARED)))
                     (less WIDTH SITE_WIDTH))
```

Such rules could then be harnessed under Prolog through the definition of each function within Prolog, itself a trivial task. In a similar manner, the other DataEase files were exported; the membership file, the decode file, the order file, the works holiday file, the wide week file. Again Prolog functions were easily developed to harness these files.

The final requirement was the Prolog code for loading these files into Prolog and initiating the required Prolog processing.

Developing this transfer of the rules and other data from DataEase to Prolog in the required format involved no significant difficulties. The only problem of any note involved the initial tracking down of errors in this transfer process due to the involvement of so many different inputs into the designed system, inputs which included DataEase data, DataEase `reports', DOS commands and Prolog code.

Re-Design of the DataEase Rule-Base

Once the transfer of data from DataEase to Prolog could be achieved, the rules being entered into the DataEase rule-base could then be tested using appropriate order data entered in the DataEase order file.

As rules were added to the rule-base, it was very soon apparent that the rule-base in its initial form was not suitable. Firstly typographical errors were common and, unless rules were tested in very small numbers, identifying these errors was a time consuming task. A second problem to emerge was that the rule-base was not easy to understand. Certainly once the number of rules exceeded fifty, the number of mistakes in the logic of the rules being added to the rule-base gave reason for concern. An initial solution was found in providing listings of the rule-base using DataEase `reports'. However the listings became less effective as the number of rules grew and the listings became longer. Further they were less than adequate in coping with typographical errors.

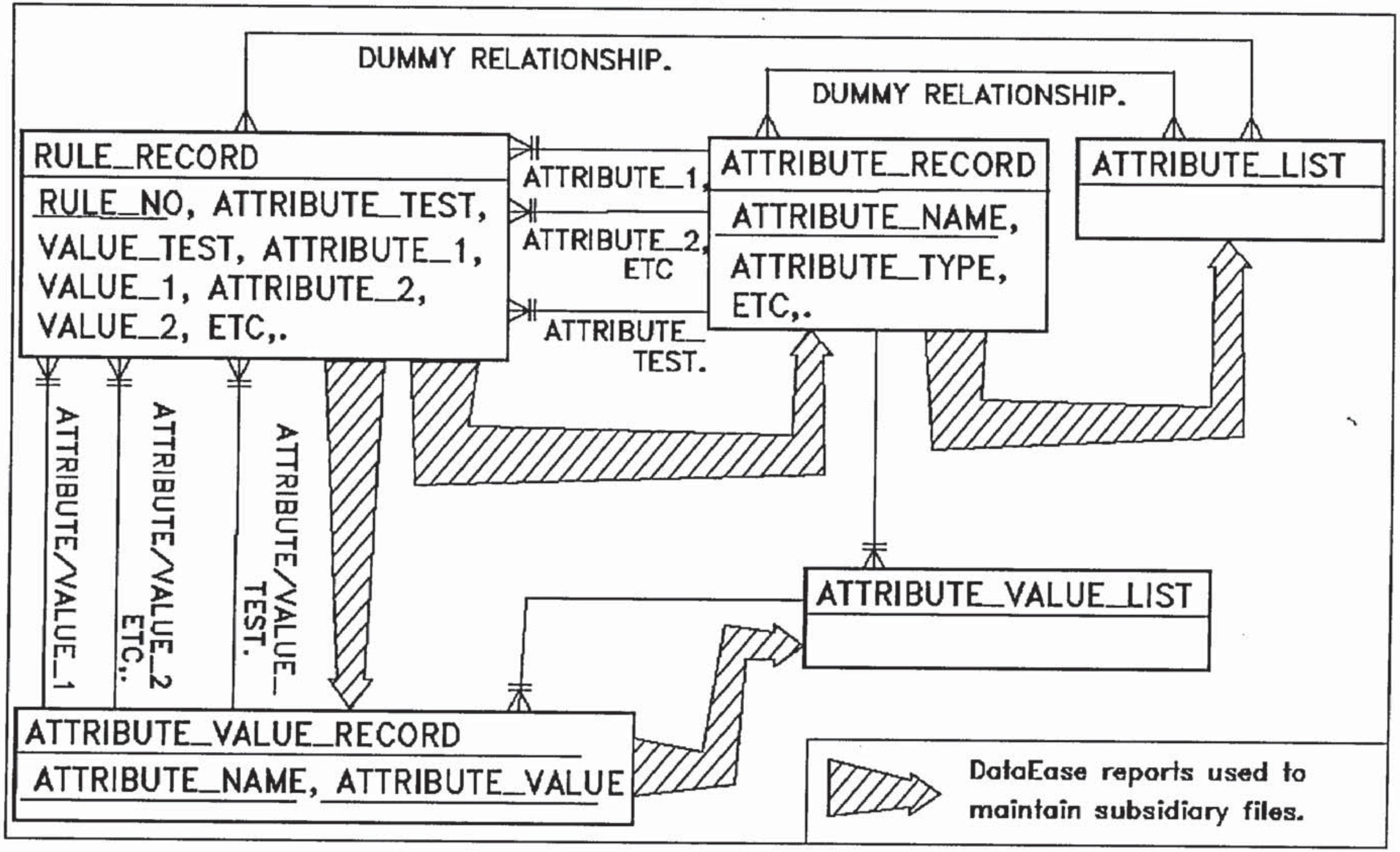


FIGURE 7.4 DATA STRUCTURE DEVELOPED TO SUPPORT RULE AMENDMENT.

To overcome both these problems, new files were created in DataEase to record the different attribute names and the different values ascribed to each attribute within the rule-base. Further files provided lists of these names and values. The files were then linked to the rule-base using relationships. The data structure is illustrated in figure 7.4 and the resultant change in the rule-base is shown in figure 7.5. The data structure allowed access to the new files in two ways. Firstly information from the new files was displayed in the rule-base itself. Secondly the files themselves could be accessed easily from the rule-base.

The data displayed from the new `attribute files' showed the use of the attribute name and the use of the value ascribed to that attribute. If the attribute name or the name-value combination was not present in the `attribute files' no data would be displayed. This would indicate, for instance, that the values used in the third and fourth clauses in rule 0122 in figure 7.5 had not been previously used with those attributes or that they contained a typographical error. The fourth clause indeed contains a typographical error. A `D` is missing from RESHAPED. The third clause has a valid value but for a different attribute (SHEET is a TYPE not a PRODUCT).

The actual values displayed from the `attribute files' show whether or not the attribute or value are present as conditional clauses, as conclusion clauses or as both conditional and conclusion clauses, the code thus displayed being 1, 2 or 3 respectively. A code of 1 indicates an input into the rule-base from the order file. In figure 7.5, the attribute WIDTH is such an input into the rule-base. Exception occurrences of the code 1 do exist and include `isin' values. The presence of a code 2 shows a clause to be unused or one of the

FIGURE 7.5 THE DATAEASE RULEBASE AS IT APPEARED IN THE PROTOTYPE MICRO-X10

------- RULE RECORD -------

No 0122

USE OF RULE: IDENTIFY WIDE/NARROW ORDERS — 09

WIDER TASK: DETERMINE PROCESS REQUIREMENTS — 05

DESCRIPTION: Identify NARROW orders.

ATT	BTH		NAME		VALUE
3	3		BREADTH	eq	NARROW

IF

ATT	BTH			NAME		VALUE		
1	1			WIDTH	less	710		
3	1		(or	OYDC2	isin	NARROW_PRODUCTS		
3			(	PRODUCT	eq	SHEET		
3				PROCESS	not	RESHAPE_RESHEARED	)	
1	1		)	WIDTH	less	SITE_WIDTH		

ATT & BTH – See page 2 below.

Attribute List

ACCESS TO OTHER FILES – See page 3 below.

Page 1 of 3.

commands initiating the Prolog processing for MICRO-X10. A code 3 shows that an attribute is both generated and used within the rule-base.

These values displayed within the rule-base itself reduced the number of logic errors when developing the rule-base and prevented almost all typographical mistakes.

Access to the other files directly from the rule-base was possible by simply positioning the cursor on the appropriate field and pressing the `F10' key. Normally within DataEase, such a procedure would present the user with the list of relationships attached to that file requiring him to choose one. However this list can be avoided by laying out the fields within the file in an appropriate order.

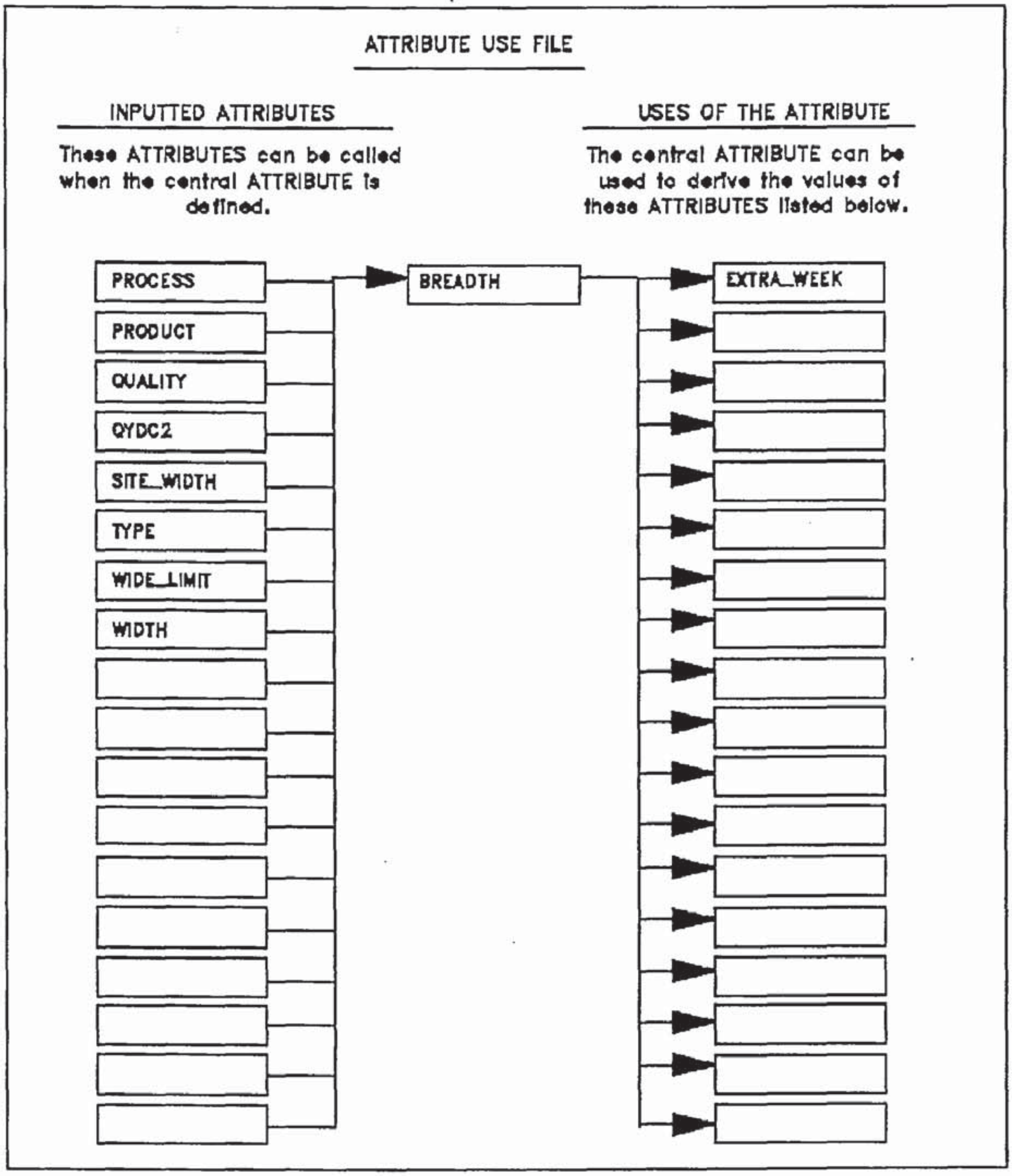
Referring to figure 7.4, the `list' files simply presented the data of the other files in a more useful form. The attribute `list' file for instance listed all the attributes of the completed system in alphabetical order on just four screens.

The main limitation in using this data structure was that it was not interactive. The data within the `attribute files' was updated through the running of a number of DataEase `reports'. These `reports' loaded and sorted all current values from the rule-base into these files.

Another recurring error during the writing of rules into the rule-base was the misplacement of brackets within the rules. Prolog is sensitive to the order of brackets and thus the existence of such errors was easily identified during the testing of sets of rules.

However such occurrences were not seen as appropriate to a working system. Thus a DataEase `report' was written to identify any bracket which was logically misplaced. For instance, a bracket cannot be closed before it is opened and must be closed by the end of the rule. Further,

FIGURE 7.6 A RECORD FROM THE ATTRIBUTE USE FILE.



the first bracket must always be an `or bracket' and nested brackets must alternate in type between `or brackets' and `and brackets'. A DataEase `report' was written to identify any rule breaking these requirements. Incidentally such a utility within Prolog would have been more difficult to develop as Prolog relies on the correct placement of the brackets to define the limits of each rule.

Further Development of Maintenance Tools

While the rule-base had been used to add new rules, it had not been used to alter and amend rules to any great extent. It was evident that there was a requirement to identify the uses of rules and attributes. This led to further functionality being built within MICRO-X10.

The initial consideration was not in identifying the functionality required to assist the alteration and amendment of the rule-base. Rather it concerned whether DataEase could be used to provide any useful functionality. This initial consideration was prompted by the view that any DataEase function could be developed very quickly compared to any Prolog function.

Resulting from this work a new DataEase file was developed to record the use of attributes in a manner similar to the attribute list file developed earlier. A record from the new file is illustrated in figure 7.6. Developing this function to cover the use of rules rather than use of attributes was too complex a task for DataEase `reports'.

Developing further functionality using Prolog was mainly constrained by limitations in the user-computer interface. The network of rule use was rather too complex to display usefully. Figure 7.7 for instance demonstrates the complexity of attribute use within (a large) part of MICRO-X10. Further, using a similar DataEase display to figure

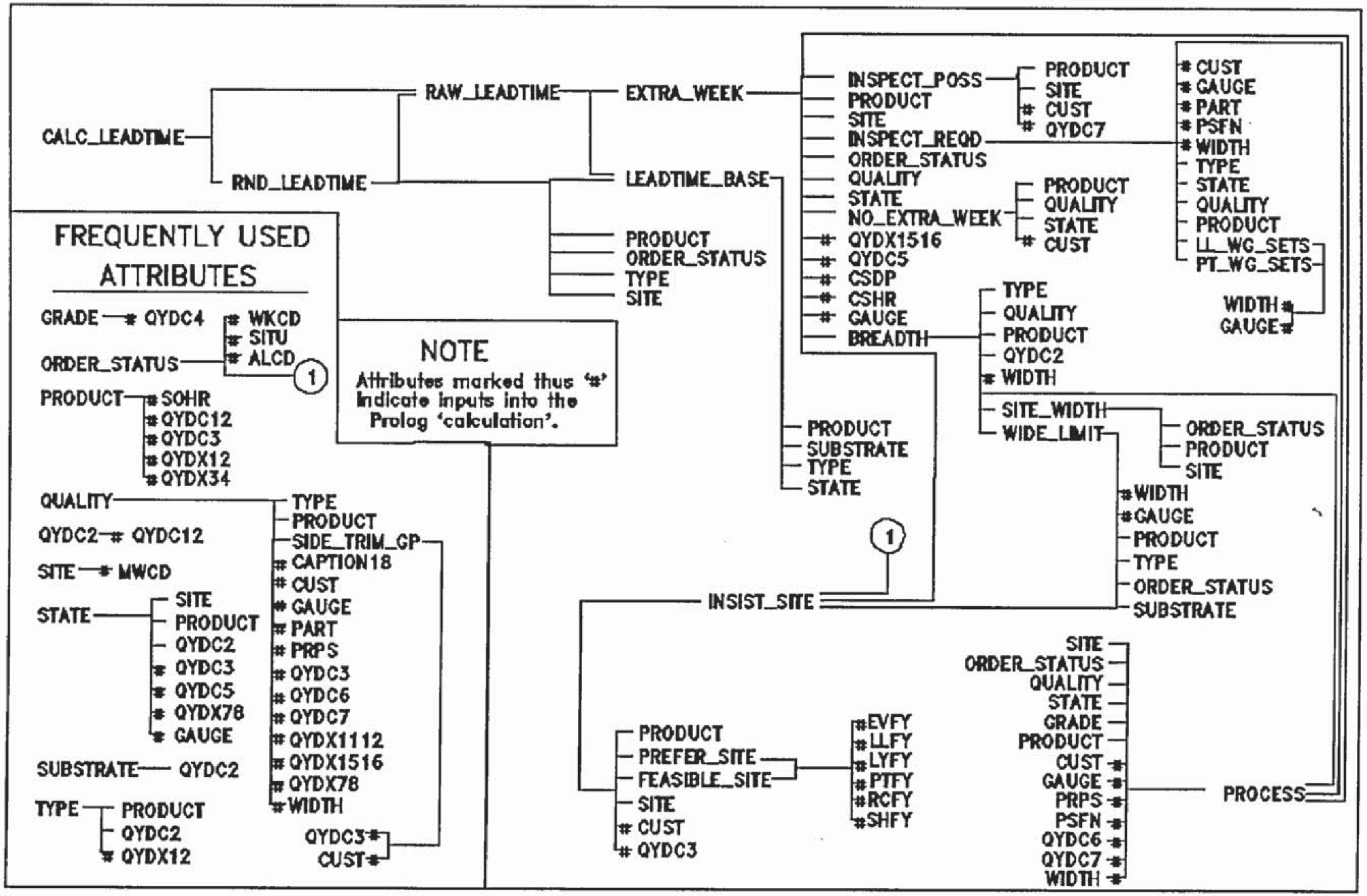


FIGURE 7.7 DIAGRAM SHOWING THE USE OF ATTRIBUTES BY PROLOG TO CALCULATE THE LEADTIME OF AN ORDER FOR MICRO-X10

7.6 but for rules rather than attributes was restricted by the larger amount of information present in a rule. In such a case, no more than two or three rules could be displayed at once.

This stage in the development of MICRO-X10 was completed by developing a Prolog function to identify rule use and record the results in a separate DataEase file. DataEase `reports' to use this file were not developed at this time.

The MICRO-X10 system was by this stage in the development ready to be introduced to the allocators to identify their maintenance requirements.

The First Version of the MICRO-X10 Prototype

When the rules extracted from X10 were written into the DataEase rule-base it contained some 250 rules. If the number of records in the membership, decode, works holiday and wide week files were added to this total along with the Prolog code used to access both these files and the rule-base, the system could be said to contain 350 rules. Beyond this were of course the various DataEase `reports' as well as the Prolog code used to read the order data and to store the results on disc ready for `importing' back into DataEase.

MICRO-X10 never totally duplicated the functionality of X10. The rules defining which member of staff should receive notification of which order were never developed. This function was seen as a straightforward database function as was the printout of the final X10-type report.

A final part of MICRO-X10 which was not operational at the conclusion of the development process was the down-loading of the data from the central mainframe. This was due to the unavailability of the

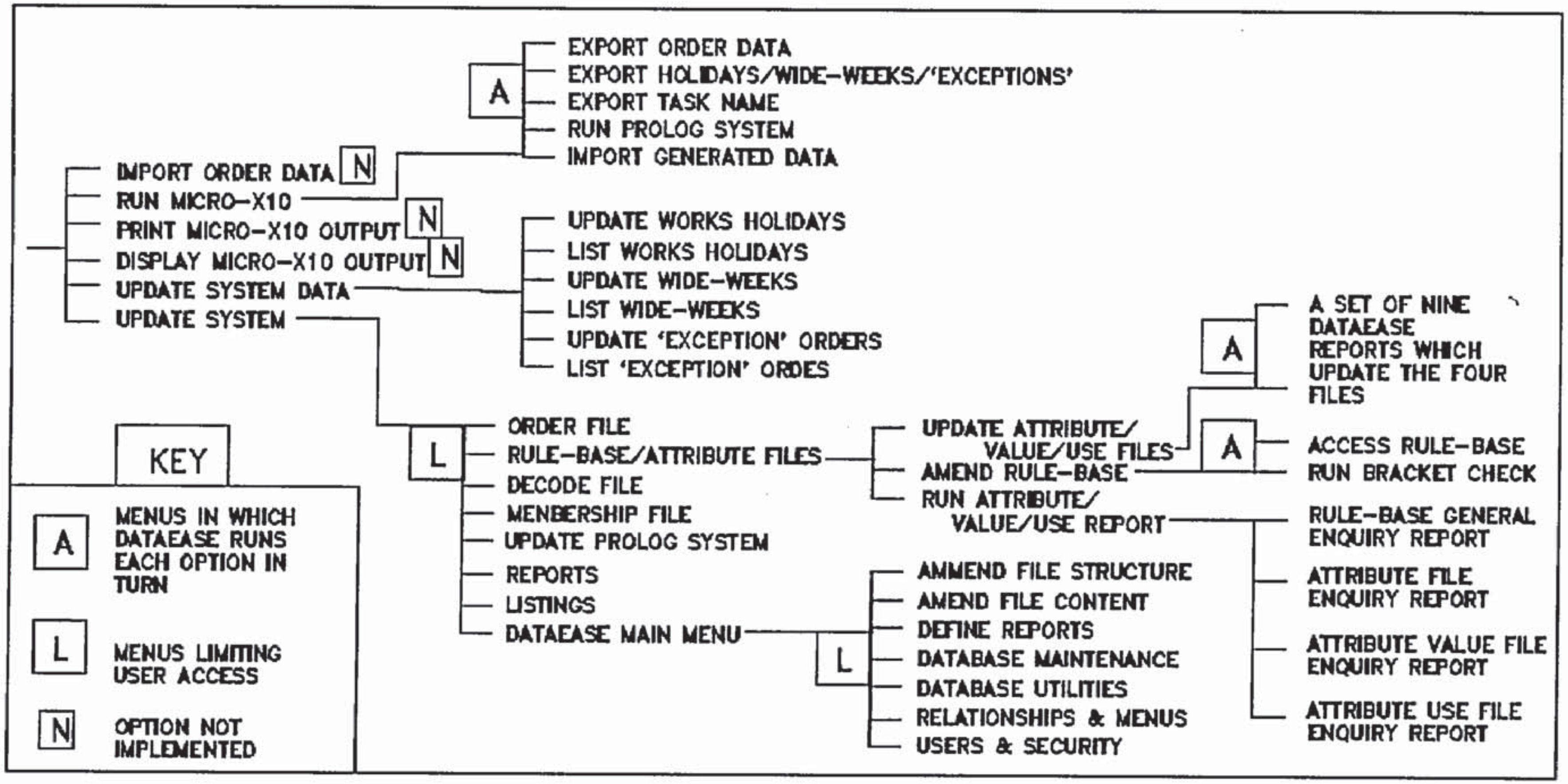


FIGURE 7.8 THE MENU STRUCTURE WITHIN MICRO-X10

micro-computer which was expected to perform this down-loading.

MICRO-X10 therefore neither ran on truly live data nor produced reports identical to the output of X10 to allow the rule-base to be tested properly.

The various functions within MICRO-X10 were accessed by a menu system developed within DataEase. DataEase provided password protection and differing access available to different types of users. Indeed a separate menu structure could have been defined for each individual user. The menu structure developed for MICRO-X10 is shown in figure 7.8.

Reflection on the Development of the Prototype MICRO-X10

With two exceptions, the process of development of the prototype MICRO-X10 had made very little reference to the analysis of system requirements. The author was conscious during the development that `summary' design decisions were being made. Two factors appeared to be responsible for this. Firstly the design decisions were mainly independent of each other and thus the consequences of any particular decision were limited. A second factor was the nature of DataEase. DataEase allowed the rapid development of many design features while at the same time restricting the possible design options.

The `summary' design decisions however were still greatly influenced by the system requirements. All design decisions had been made within the context of developing a prototype capable of carrying out the function of X10 while being maintainable by personnel unable to cope with the maintenance requirements of X10.

The exceptions to the general rule of `summary' design decisions were the initial design of the rule-base and the consideration of design options after the designs available easily through DataEase had been exhausted. In the first instance the design work had really been completed during the requirements analysis. In the second instance the design decisions were dependent on the results of the testing of the prototype.

The design process of MICRO-X10 certainly differed greatly from the design process of UAAMS. In terms of the use of models and prototypes (see section 5.3.3), the initial structure of MICRO-X10 shown in figure 7.1 remained almost unused as an EXPOSITORY model reversing the experience of UAAMS. The prototype was explicitly developed as both an EXPLORATORY model, that is to identify the user requirements, and as an EXPLANATORY model in providing the users and especially the `primary amender' of X10 with information concerning the alternative system to X10.

Within the EXPLORATORY development of the prototype, the development of DataEase `reports' and files would be categorized as EXPERIENTIAL models as their purpose was to identify their deficiencies if any. In developing these EXPERIENTIAL models the prototype system itself took the role of the EXPOSITORY model. It was not until the easy design options had been exhausted that the prototype became deficient in this role.

The development of the rules within the rule-base also can be examined in terms of the models it used. The EXPOSITORY model through the initial part of its development was an alphabetical listing of the rules generated from DataEase. With the addition of later maintenance tools, this listing lost much of its role to the output of these tools. As the rules already existed in X10 and needed only to be translated into rules suitable to the rule-base of MICRO-X10, the listing of X10 stood as an EXEMPLARY model.

The Demise of MICRO-X10 as a Working System

MICRO-X10 was developed over a period of nine weeks. But by the end of this period it had been evident for some time that MICRO-X10 was not going to be usable as a replacement for X10. The rules being entered into the DataEase rule-base were tested by exporting them along with the rest of the rule-base into Prolog and then running MICRO-X10 with some fifty test orders which were amended for this purpose. As the rule-base grew in size, the time taken to process each order increased.

The first step taken was to replace the test orders with real orders taken from the central computer. This made little difference to the processing time. By the time the rule-base was complete the average process time was some thirty seconds per order. A trace of the Prolog processing showed that there was no unexpected processing being carried out by the Prolog.

Given this situation, both the Prolog and the rules were amended to increase the speed of the system. These amendments included calling decodes and order data through rules in the rule-base specifically written for each affected attribute rather than general clauses in the `eq' relationship. Also holidays and wide weeks were written directly into the rule-base and the special files ignored. The `eq' relationship was removed from the output of the DataEase rule-base and used only to allow subsidiary results to be recorded to prevent them being recalculated.

The amendments greatly increased the speed of MICRO-X10. The average time for processing an order dropped to six seconds. This however remained too slow. The number of new and updated orders requiring checking by MICRO-X10 was some three thousand giving a run time of five hours! This was clearly impractical.

The surprise was not that the system was so slow. Rather the surprise was that the problem with the speed had not been considered during the requirements analysis stage. The size of the rule-base had been known reasonably accurately. The number of orders requiring checking was also known. It was also known that to expect a 350-rule rule-base to answer an enquiry involving the majority of the rules in less than one second using the computers and software available was asking too much.

The only explanation for this situation is that the speed of an expert system was not considered as a limiting constraint on the technology and that the ability to estimate the speed of an enquiry from the number of rules in the rule-base and the `breadth' of the search within these rules had not been made explicit. Such learning is clearly important for PP&C systems which analyse individual orders in a large orderbook.

7.4 TESTS CARRIED OUT ON MICRO-X10

The demise of MICRO-X10 as a working system did not prevent the system being used to identify the amount of change which would have affected it through time. The first sub-section presents the finding from this work.

Testing the success of MICRO-X10 in providing an understandable and maintainable programming environment for the X10 programme was affected by its failure as a working system. The allocators were by this time fully aware that MICRO-X10 was incapable of replacing X10. However they were still willing to be involved in testing the understandability of the finished prototype system. This work is presented in the second sub-section.

The Scope of the Maintenance Task of X10

X10 had undergone a major alteration a year prior to the building of MICRO-X10 and in the following twelve months had undergone further minor alterations.

At the end of this period, X10 was examined to identify the effect of these minor alterations on the rule-base of MICRO-X10. To update MICRO-X10 so as to incorporate these alterations a total of thirteen rules and one membership record required deletion while 23 rules, two membership records and one decode record required amendment. As well as this, a further 24 new rules were required. In total, this represented change in almost a quarter of the rule-base.

The causes of the change were mainly due to redefinition of the leadtime rules themselves. There were three other causes which together accounted for one third of the change. These three causes were amendments to the codes used in the central computer order files, changes in the production facilities at the works and changes in the output required of X10.

The allocator who maintained X10 was not surprised by the level of change. He described the changes in the leadtime rules over the year as "greater than normal". On reflection however he agreed that the change that had occurred was more easy to cope with than was normal. The type of change which was less evident than usual over the 12 months and was normally more difficult to cope with was change in the production facilities at the works.

MICRO-X10 was easily capable of absorbing all the alterations. Of the amended rules, about one quarter involved the addition or deletion of clauses. Each amendment was thus contained within a single rule. About half the amendments were somewhat more extensive and best tackled

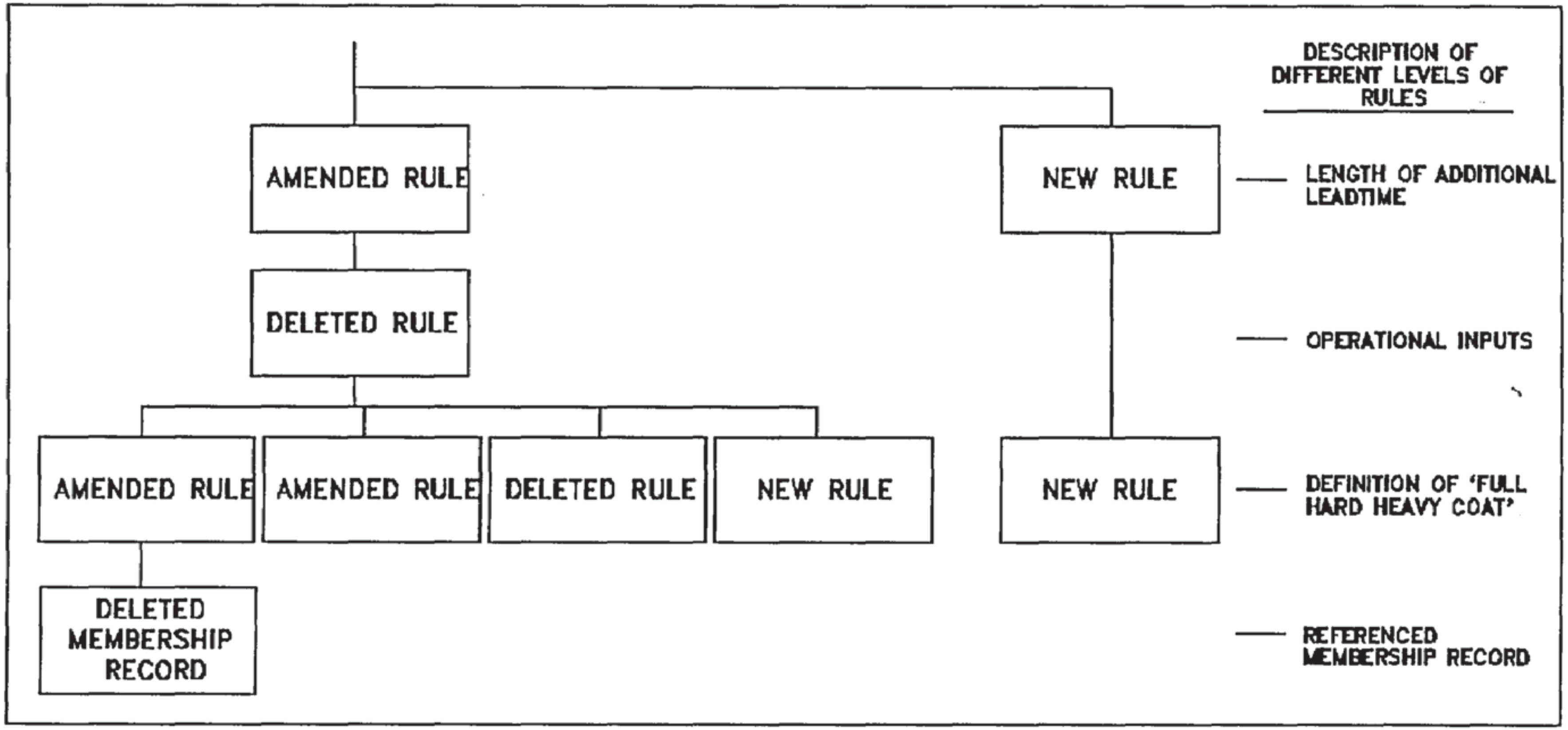


FIGURE 7.9 AN ILLUSTRATION OF RULE AMENDMENT WITHIN X10 SHOWING THE EXTENT OF AN AMENDMENT ALTERING THE EXTENSION IN LEADTIME AFFECTING 'FULL HARD HEAVY COATED' STRIP

by rewriting the rule from scratch. The majority of such amended rules were part of sets of similar rules some or all of which were also subject to amendment.

The final quarter of the amended rules involved multiple sets of rules, and the addition and deletion of other rules. A typical example of this is illustrated in figure 7.9. It concerned `full hard and heavy coated' Zinc coated strip and involved five rules and one membership record. Of these items, three were deleted, three were amended and three new rules were added. This example represented one seventh of the change within X10 during the twelve month period.

The reason for the change in the `full hard and heavy coated' rules within X10 was prompted by a change in the leadtime rules. However, once the change was considered, changes in the use of codes in the central computer order file were also incorporated. This extra requirement resulted in the deletion of one rule and the amendment of another.

The deleting of rules when carried out in isolation was a straightforward alteration. Half of the new rules added to the rule-base involved linking the new rule into different levels of the rule-base rather than the new rule simply being added to an existing set of rules.

The amendments in X10 over the twelve month period were easily encoded in MICRO-X10. This then demonstrates an instance of an expert system coping with the maintenance requirements imposed by a PP&C system. The abilities of MICRO-X10 to cope with new production facilities at the works had not been demonstrated.

However this has not demonstrated was whether or not MICRO-X10 would have been understood well enough by personnel within the situation to cope with the maintenance task required by MICRO-X10.

<u>The Understandability of MICRO-X10</u>

The presentation of MICRO-X10 to the allocators attempted to investigate two basic areas. The first area was the abilities of the allocators to cope with the maintenance requirements. The second was their acceptance of MICRO-X10 as a replacement to X10. This second area concerned only two allocators, and concerned mainly only the allocator who carried out the major amendments for X10.

Generally the allocators demonstrated that they were able to cope with required amendments in MICRO-X10, both those familiar and those unfamiliar with the workings for X10. It must be remembered that although most allocators were unable to amend X10, they were capable of amending other C.A.EARL programmes and were conversant with the terminology of the central computer file. The complaint of these allocators with regard to X10 was that they could not be sure that any change made by them would not have undetectable side effects in the operation of the programme. This complaint was not made about MICRO-X10. The allocators were confident that the use of any rule within the rule-base could be identified. Thus any rule could be amended with the knowledge that all uses of that rule had been considered.

Despite this apparent acceptance of MICRO-X10, all but one of the allocators remained uneasy with the logic of the rule-base and its lack of a complete procedural listing of its operation. This unease was never fully dispelled but it was admitted that with time the rule-base would prove itself (or otherwise).

The one exception was the allocator who carried out the major amendments in X10. His unease about MICRO-X10 was more based on the abilities of such a rule-base to duplicate the functions of X10. This unease did appear to be dispelled as the functionality of the rule-base became understood by him.

One area of deficiency with MICRO-X10 was found to be the support given in identifying the rules relevant to a required amendment. A DataEase `report' had been provided to search for the relevant word within any of the fields of a particular file. However, these reports often failed to find the correct rules due to differences in terminology or spelling. The multiplicity of jargon used within Order Allocation exacerbated this situation.

As an example of this, one allocator attempted to identify the rules affecting UNANNEALED strip. His initial search failed as he was using the term HARD_IRON which means UNANNEALED as does BLACK_IRON. (In this case the term used would depend on one's origins within BSSP.)

The other utility within MICRO-X10 expected to help in such circumstances was the `list' files. If the `word' being sought were contained in a `value' field rather than an `attribute' field, the `list' files proved of little help.

This then identified the need for a new function in which equivalent and similar terms could be identified. Work developing such a function was not initiated due to a lack of project time coupled with the demise of MICRO-X10.

To test the usefulness of rule descriptions in the rule-base, descriptions for some rules were written on paper and presented to the allocators when they began amending these rules. The rule presented in figures 7.3 and 7.5 was described as follows:-

> GENERAL DESCRIPTION - Narrow strip is slit from wider coil. This generally adds to the leadtime of HR or CR strip.
> CLAUSE 1 -The widest slit coil for the whole group is 710mm so this rule prevents further processing for most orders.
> CLAUSE 2 -QYDC2 identifies coil requiring extra leadtime if slit (basically CRCOIL and HRCOIL).

CLAUSE 3 & 4 - HRSHEET which is not reshaped_resheared requires an extra week in the leadtime. CRSHEET does not require any extra leadtime.
CLAUSE 5 - The minimum unslitted width varies by product and by site, and is returned by SITE_WIDTH.

Writing such descriptions into a rule-base was a discipline which was not tested. Separate comment by three of the allocators regarding the function of rules being amended when descriptions were provided stressed the need for a description of attributes rather than rules. This extended far beyond any description possible through extending the length of attribute names. It was apparent that the allocators were asking for a means of providing discipline within the rule-base. An attribute is normally defined by the rules which identify its existence. The allocators were effectively asking that these definitions be made explicit and fixed. Such descriptions could have been easily provided within the various attribute files of MICRO-X10.

It should be noted that descriptions within rule-bases in general usually attempt to assist the user of a rule-base rather than the maintainer. Thus such discussion of attribute descriptions is not found in expert system literature.

The second area of investigation, the acceptance of MICRO-X10 as a replacement for X10 was affected more by the failure of MICRO-X10 as a working system than the investigation of the allocators' abilities to cope with amendments in MICRO-X10. The client, that is the allocator who carried out major amendments to X10, knew only too well the limitations of X10. Thus the extra functionality available in MICRO-X10 was seen as a great advantage, especially within the policies of more accurate delivery dates then being pursued within BSSP. His conclusion was that had MICRO-X10 been able to duplicate the results of X10 for a week and given a usable run time, he would have been confident enough with the working of MICRO-X10 to terminate the use of X10.

Despite this view being given honestly, the effect of knowing that MICRO-X10 was unusably slow may have coloured this view.

7.4 FINDINGS FROM THE MICRO-X10 DEVELOPMENT

This final section discusses the findings from the MICRO-X10 development. The findings are presented under the headings of the five reasons for developing MICRO-X10 as stated at the beginning of this chapter.

DataEase as a Front-End for an Expert System

MICRO-X10 has demonstrated the viability of using a proprietary database management system (DBMS) to provide the user and maintenance interfaces for an expert system. The expert system techniques were amply provided by Prolog which in no way restricted the solution.

As discussed in section 5.3.3, such use of a proprietary DBMS as a front-end for an expert system has not been advocated by other authors.

The choice of the actual software to use, LPA Prolog version 1.2 and DataEase version 2.5, was obviously dependent on the software available at the time of the decision. More powerful versions of Prolog (for instance Prolog++) which have become available since this decision, while not rigorously tested within the project, do not appear to alter the rationale behind the original choice of such a combination of AI language and DBMS which would also take account of more powerful DBMS which may have become available.

Implementation Issues in Expert PP&C Development

The failure of MICRO-X10 to become a working system implies that MICRO-X10 did not truly test the implementation issues within PP&C. However the failure of MICRO-X10 was not due to soft implementation problems. This then supports the contention that the constructs within the problem-driven approach used to cope with such problems, namely SSM and the list of actors, were successful. Being an extension of SSM, this work is itself action research. As such it is not possible to draw general conclusions from just one úse of these constructs.

Detailed Design Using the Problem Driven Constructs

The design of MICRO-X10, unlike UAAMS, was not assisted by the expansion of the initial structure of the system (figure 7.1) to provide detailed activity diagrams of the various functions of the system. This was unnecessary because the rules within the rulebase were easily extracted from the listing of the X10 programme. Thus the functionality required within the rulebase to encode the rules was easily identified and implemented. The functionality developed to support the maintenance of MICRO-X10 was identified during the analysis stage as consisting of straightforward and separate functions. Their development within DataEase required no support from activity diagrams.

MICRO-X10 thus provided no learning with regard to the use of the problem-driven constructs as an aid to detailed design within the approach.

The absence of significant interdependence between the different functions within MICRO-X10, while not requiring the support of activity diagrams during their design, allowed the prototype MICRO-X10 system to take the role of an EXPOSITORY model of the system. Had there been significant interdependence between the different functions, its use as an EXPOSITORY model would not have been possible, and the problem-driven nature of the development would have been drawn into question.

The MICRO-X10 development then provides a second form of problem-driven development of an expert system and emphasizes the role played by EXPOSITORY models within the problem-driven approach.

Maintenance Requirements of PP&C Systems

The MICRO-X10 development has provided an example of the maintenance requirements within the domain of PP&C. Over a twelve month period, maintenance resulted in changes to almost a quarter of the rulebase. The causes of this change and the scope of the task of implementing the change were described in section 7.4.

The rulebase within MICRO-X10 was well capable of absorbing this change and in this respect was a great improvement over the existing conventional programme, X10. The most difficult maintenance requirements, coping with changes in the production facilities at the works, were less evident than is normal over a twelve month period. As a result the abilities of MICRO-X10 to cope with such change were not fully tested.

The Usefulness of Expert Routing Systems

During the development of UAAMS, expert systems appeared as a useful means of encoding routing rules. Such rules often exist already within PP&C in an explicit form. The UAAMS development had also shown the knowledge encoded in such rules to be predominantly shallow in nature.

The MICRO-X10 development again identified routing knowledge as predominantly shallow in nature. One characteristic of the development was that this knowledge was already encoded within an existing conventional programme. Thus the development did not have to define the rules it used and thus did not encounter any `soft' problems resulting from such a task.

The most obvious finding from the MICRO-X10 development was the slow operation of Prolog which made the system unusable. It is more significant that this problem could have been identified prior to encoding the rules. Although the exact number of rules in the rulebase and the average proportion of these rules used to analyse an order were not known, the estimated numbers that were available would have identified the speed of the programme as a major problem. Such a problem could easily affect other routing systems required to analyse large numbers of orders. While the speed of MICRO-X10 is evidently dependent on both the software and the hardware used, the speed required to make MICRO-X10 usable would be some ten times greater than that achieved.

The other facet of this problem with the slow operation of expert systems is whether a situation could exist where the problem being tackled was important enough to merit an excessively long run-time. Certainly the task carried out by MICRO-X10 was important, but of course, X10 copes with the situation. It will be remembered that the problem being tackled by MICRO-X10 was the problem of maintaining X10. An expert system which also analyses an orderbook is R1/XCON (346). It can be seen here that the problem being tackled by R1/XCON is important enough to merit long run-times and the alternative solution, manual configuration, was unacceptable.

The conclusion with regard to the usefulness of expert systems for carrying out routing tasks is that the knowledge required by such systems is amenable to incorporation within an expert system but the resulting usefulness may be constrained by the slow operation of expert systems when required to analyse large numbers of orders.

CHAPTER 8
CONCLUSIONS

The first section of this chapter describes the scope of the research. The second section presents the conclusions relevant to the various disciplines involved in the research. The third section considers directions of future research.

8.1 THE SCOPE OF THE RESEARCH

The project was initiated by the desire to improve a problem situation within the domain of PP&C. The project thus has reviewed the `traditional' approaches and techniques for problem solving in PP&C from a `problem-driven' perspective. In this respect, some techniques were not in any way appropriate to the project task, for instance Just-In-Time and Economic Order Quantity, and were thus not reviewed fully.

The project was also driven by the desire to identify and harness the usefulness of expert systems within the problem situation. While it is expressed within the literature that expert systems are applicable to the domain of PP&C, the literature does not provide a track record to support such a proposition. Work in identifying the track record of expert systems beyond that recorded within the literature was not considered appropriate project work and is thus beyond the scope of this thesis.

The expert system literature was also used to review expert system methodology; that is, how to go about developing an expert system. The main-stream methodology was found to be deficient when viewed from a `problem-driven' perspective, particularly with respect to the domain of PP&C. The results of a trial development of an expert system using the exploratory approach (the Leadtime system) supported this view. As a result a problem-driven approach for expert system development was developed. The different constructs within the problem-driven approach were developed within a number of conceptual designs. As the rationale behind each of the constructs is described adequately outside the context of these conceptual designs and the usefulness of each construct is demonstrated by later work within the project, a detailed account of the conceptual design work is beyond the scope of this thesis.

Two systems were developed using the problem-driven approach, UAAMS and MICRO-X10. Neither system gained the distinction of becoming a usefully working system. However, their development does stand as a test of the practical usefulness of expert systems within the domain of PP&C. They also stand as demonstrations of the use of a problem-driven approach to expert system development.

The development of UAAMS and MICRO-X10, while they approach the use of expert systems and expert system techniques from a `problem-driven' perspective, cannot be isolated from the considerations usual within more laboratory-based research. Three points are important here, firstly the choice of expert system software, secondly the effect of the work being action research and finally the constraints imposed on the description of this research work.

The choice of software to use was initially made in late 1987 and the factors affecting this decision have been fully described. Neither at that time nor at the time of the termination of the UAAMS development in mid-1989 was all available expert system software exhaustively researched. Under a `problem-driven' perspective, the task of the research is not to identify the most suitable software for a particular development. Rather it is to identify the abilities of the technology using software which is not considered unsuitable to the task in hand.

The problem-driven approach developed was strongly based on the ideas of Soft Systems Methodology (SSM). The concepts of action research are thus easily relatable to the problem-driven approach. Not only is the real-world problem situation a major input into the outcome of such research, but the researcher must be considered as a part of that problem situation. Thus it is impossible to develop general conclusions from a single application of the problem-driven approach. This is also the case with the conclusions regarding the usefulness of expert systems within the domain of PP&C but to a much lesser extent. While the usefulness of expert systems remains the outcome of action research, it is possible to relate the factors within the problem situation to other descriptions of PP&C.

A final consideration concerns the complexity of the real-world situation and the limitations imposed on any description provided by this thesis of the situation itself or the project work carried out within that situation. The description of the problem situation is in many ways a simplification of the real situation. Similarly, the description of the project work could also be expanded in many areas. While the problem situation is described to provide suitable background understanding, the description of the project work attempts to cover all areas relevant to the research. Within this research, the problem-driven approach can be described as `self-constructing' (1), the starting point being SSM. The development of the two expert systems, UAAMS and MICRO-X10, represents the first true iteration around the `self-constructing' cycle. Factors which will be identified as important to the approach within future research may not appear significant at such an early stage of research as this. Given this, with little foregoing research being specifically relevant to the problem-driven approach, it is not possible to provide the correct description of the use of the problem-driven approach within the confines imposed on this thesis.

8.2 CONCLUSIONS

The conclusions relating to the various disciplines of this Inter-Disciplinary project are presented in successive sections.

8.2.1 THE USEFULNESS OF EXPERT SYSTEMS WITHIN PP&C

The conclusions as to the usefulness of expert systems within the domain of Production Planning and Control (PP&C) are presented in five sub-sections. Firstly, the problems involved in the maintenance of such systems are presented. The three following sub-sections cover the usefulness of expert systems for three distinct tasks within PP&C; scheduling, plan maintenance and routing. A final sub-section compares these conclusions with those of other authors.

The Problem of Maintenance

The leadtime system, described in section 5.2, highlighted the maintenance requirements for expert PP&C systems. An example of the levels of maintenance required by such systems was provided by MICRO-X10 (presented in section 7.4).

The implications of the maintenance requirements within PP&C are that they preclude the general use of the simpler currently-available expert system tools. The inflexibility of such tools led to incorporating implementation devices within the knowledge-base increasing the complexity of the maintenance task. The MICRO-X10 system provides an example of the extra functionality required to prevent implementation devices being incorporated within a rule-based PP&C system and the assistance required within the maintenance task. Tests carried out on MICRO-X10 showed that the resulting maintenance task was within the abilities of PP&C staff within the situation.

Expert Scheduling Systems

The UAAMS development described in chapter 6 found a major input into the design of an expert scheduling system to be the design of the `hard' scheduling constructs required by such systems. A major consideration in this regard was found to be the constructs in current use within the situation.

Further, the abilities of AI languages to cope with such `hard' constructs have been drawn into question. Specifically, the use of Prolog was shown to be inappropriate within UAAMS, although expert system literature gives no indication of any such limitations in the use of Prolog. The use of alternative software to develop expert scheduling systems and capable of coping with `hard' scheduling constructs remains an area requiring further research.

Expert Systems for Plan Maintenance

Plan maintenance was the subject of analysis within the UAAMS development. The findings of this analysis were that, while plan maintenance within British Steel Strip Products is less complex than the scheduling task with regard to arithmetical manipulation, it is less well defined and more subject to `soft' problems encompassing conflicting objectives than the other tasks within UAAMS. The project, however, gave no opportunity to test such findings by developing expert systems for plan maintenance. Thus these findings must remain speculative.

Expert Routing Systems

The development of UAAMS identified routing knowledge to be predominantly shallow in nature and apparently well suited for the application of expert system technology. The development of MICRO-X10 described in chapter 7 confirmed the generally shallow nature of routing knowledge, usefully encoding such knowledge within a simple rule-base (although certain aspects of MICRO-X10 were encoded in extensions to this rule-base). The exceptions to this general rule regarding the shallow nature of routing knowledge included all temporal considerations within MICRO-X10.

The one problem encountered with the development of MICRO-X10 was the slow speed of the completed prototype system, a problem which prevented its adoption as a working system. The implications of the slow speed of expert systems on the usefulness of routing systems can be significant where the size of a company's order book is large or the analysis of the individual orders is involved.

Relating the Conclusions to the Literature

The track record of usefully working expert systems within PP&C described in section 4.3 consisted of two routing systems and fourteen scheduling systems. However it was argued (see section 5.1.1) that these systems were inadequately described within the literature to demonstrate the usefulness of expert systems within the domain. The track record was found to show more sophisticated software being used within PP&C than other domains (with the exception of process control systems). This was argued in section 5.1.1 to be due to the maintenance requirements of PP&C.

Views expressed by other authors concerning the usefulness of expert systems within PP&C were reviewed in section 4.4. These authors consider either the scheduling task or the domain of PP&C generally. The most detailed of these accounts relate the capabilities of expert systems to the failings of traditional techniques and identify expert systems as being capable of coping with the complexity and adaptability required within PP&C systems. The author's conclusions, based on the development of UAAMS and MICRO-X10, generally agree with these views but provide a far more detailed reasoning behind such conclusions.

8.2.2 CONCLUSIONS REGARDING EXPERT SYSTEMS GENERALLY

The thesis, in viewing expert system technology from a `problem-driven' perspective, has identified the need for a universally accepted list of usefully working expert systems. Problems in identifying such a list are discussed in section 4.2. In circumventing the absence of such a list, a personal view of such a list was presented and conclusions as to the usefulness of expert systems derived. These conclusions, presented in section 4.3, categorize the great majority of systems as fault diagnosis systems, process control systems, `electronic rulebooks' or `configurers'. Such conclusions were shown not to contradict the conclusions of other authors. However, this list remains a personal view. Future research may reveal inaccuracies or incompleteness within the list.

In section 4.5 expert system methodology was reviewed. A spectrum of methodologies was developed and the conclusion reached that the main-stream expert system methodology was exploratory in nature, although the literature provides little evidence of the actual development process used to develop expert systems. That the generally advocated approach to expert system development is exploratory is not generally acknowledged by other authors.

The UAAMS and MICRO-X10 developments provide two examples of expert systems which failed to gain the status of working systems. Such examples have previously been elusive (237).

8.2.3 CONCLUSIONS REGARDING THE PROBLEM-DRIVEN APPROACH

The problem-driven approach can be described as a domain-specific development of Soft Systems Methodology (SSM). As such the development of UAAMS and MICRO-X10 provides a demonstration of the usefulness of the approach, but as this work was action research, cannot provide general conclusions.

Of the five requirements of a problem-driven approach, described in section 5.3.1, two were demonstrated to be satisfied. `Soft' problems were not a cause for concern for either of the two developments. Also the identification of the prospective tasks and roles of expert systems within the design process was never found deficient.

Of the other three requirements, activity diagrams were found to be an appropriate format for the systems' specifications and designs at a broad level. The two developments did not allow full testing of this format at a detailed level of design.

Another requirement, choosing software which will not constrain the development, was not satisfied. The limitations of Prolog encountered were, however, not of a form anticipated during the choice of the software. (See section 5.3.3.) The limitations considered at that time did not restrict either development.

The final requirement was to be able to understand the use of models and prototypes within the approach. The two uses of the approach followed significantly different courses in the manner in which the design was expanded. The different uses of models and prototypes defined in section 5.3.3 showed the significance of the role played by what were termed EXPOSITORY models in differentiating a problem-driven approach from an exploratory approach.

8.2.4 CONCLUSIONS RELATING TO SOFT SYSTEMS METHODOLOGY

The problem-driven approach to the development of expert systems, described in chapter 5, can be seen as a development of the main-stream use of SSM. It thus stands as a domain-specific use of SSM. In a similar manner, the uses of SSM for identifying organizational information requirements developed by Wilson (26) can also be described as domain-specific.

In section 3.3.2, the review of SSM as a means of tackling PP&C problems identified the deficiency in the broadest description of SSM to be that it provides a generalized description of SSM rather than an all-embracing description encompassing all uses of SSM. Previous criticism of the description of SSM has been less specific in identifying this deficiency. Ignoring the one account of Mode 1 and Mode 2 SSM given by Checkland (101), domain-specific uses of SSM currently remain outside the broadest description of SSM.

Within the problem-driven approach is the construct termed the Intelligent Agent (described in section 5.3.3). This model was originally developed as a conceptual model. The re-use of conceptual models is not considered appropriate within main-stream SSM and thus the term `semi-precious' model was coined to describe such models as the Intelligent Agent. The concept of `semi-precious' model appears a useful construct within the domain-specific application of SSM and as such their contentious nature requires to be reconciled within SSM.

Section 3.3.3 identified many different descriptions of the PP&C task as well as differences in the importance ascribed to them by their authors. Also in section 3.3.3, a model was developed of the minimum necessary activities within the PP&C task. This use of the SSM concept of minimum necessary activities has not been explicitly used to structure the task of PP&C previously. Indeed, the application of SSM to the structuring of disciplines such as PP&C remains a totally unexplored perspective for the use of SSM.

8.3 DIRECTIONS OF FUTURE RESEARCH

This thesis has demonstrated the usefulness of a problem-driven approach to the development of expert systems within the domain of PP&C. As mentioned in section 5.3.2, the constructs within the approach are not considered as being only relevant to PP&C. Future research will be required to identify domains of a similar nature to PP&C in which a problem-driven approach may also be useful.

The conclusions reached by this thesis regarding the usefulness of expert systems within the domain of PP&C catagorized expert systems into three different catagories; scheduling, plan maintaining and routing. Future research will be required to confirm these conclusions both regarding the usefulness of expert systems and the applicability of the catagorization.

The general usefulness of the problem-driven approach and each of its constructs cannot be determined from a single piece of action research. Future research is required, using either the approach developed here or constructs fulfilling a similar function, to allow such general conclusions to be made.

Finally, the abilities of expert system software were twice found deficient for their application to real-world problems. This points to a need for more research into the comparative abilities of expert system software.

REFERENCES

1 P.B. Checkland (1981) Systems Thinking Systems Practice. John Wiley, Chichester.

2 J.B. Pond (1985) Iron Ages's 1984 Top 50 World Steel Producers. Iron Age 228, May 3rd, 1985, pp.72-82.

3 Anon. (1988) British Steel - Showing its Mettle. The Economist, July 2nd, 1988, pp.73-74.

4 British Steel Corporation (1987) Annual Report and Accounts 1986/87.

5 K.Warren (1970) The British Iron and Steel Industry since 1840, Bell, London.

6 I. Okinaka (1986) Problems in the Production Management System of the Steel Industry. In Towards Interactive and Intelligent Decision Support Systems, Volume 1 (Y. SAWARAGI, K. INOUE and H. NAKAYAMA, Ed.), pp.353-362, Springer-Verlag, Berlin.

7 Organization for Economic Co-Operation and Development (1974) Forecasting Steel Consumption. O.E.C. and D., Paris.

8 G.W. Plossl (1988) Throughput Time Control. Int.J.Prod.Res. 26, No.3, pp.493-499.

9 D.H. Kelley (1976) Production Planning and Control Systems in the British Steel Corporation. British Steel, Port Talbot.

10 F. Fitzgerald (1988) Energy, High Technology and Economics in Modern Steelmaking. Technical Update 1, pp.7-19.

11 W. Jack (1987) Expert Systems in Maintenance: A Case Study on Continuous Casting Plant.Steel Times 215, No.2, pp.70-72.

12 M.A. Rodger (1985) The Kemutec Project: An Intervention in the Production Function of a Manufacturing Company, Master's Dissertation, Dept of Systems, University of Lancaster.

13 G.W. Plossl (1985) Production and Inventory Control: Principles and Techniques, Prentice-Hall, NJ.

14 D.K. Corke (1987) Production Control Engineering. Edward Arnold, London.

15 V. Prabhu and M. Baker (1986) Production Management and Control, McGraw-Hill, London.

16 D.T. Koening (1987) Manufacturing Engineering - Principles for Optimization, Springer-Verlag, Berlin.

17 O.W. Wight (1981) M.R.P. II: Unlocking America's Productivity Potential Oliver Wight Ltd Publications, Williston VT.

18 J.A. Tompkins (1986) Presentation Outline Computer Integrated Manufacturing Update. Presented at the 1986 Fall Industrial Engineering Conference December 1986, Boston MA.

19 M.S. Fox (1983) The Intelligent Management System: An overview. In Processes and Tools for Decision Support (H.G. SOL, Ed.), North-Holland, Amsterdam.

20 K.D. Barber and R.H.Hollier (1986) The Effects of Computer-Aided Production-Control Systems on Defined Company Types, Int.J.Prod. Res. 24, No.2, pp.311-327.

21 P.E. Galgut (1982) Adaptive Loading and Real Time Scheduling Procedures. Engineering Costs and Production Economics 6, pp.81-87.

22 P.F. Bestwick and K. Lockyer (1982) Quantitative Production Management. Pitman, London.

23 H. Mather (1982) Making It All Work. Production and Inventory Management 23, No.3, pp.67-73.

24 A.C. Hax and G.R. Bitran (1979) Hierarchical Planning Systems - A Production Application. In Disaggregation: Problems in

Manufacturing and Service Organizations (L.P. RITZMAN, L.J. KRAJEWSKI, W.L.BERRY, S.H.GOODMAN, S.T. HARDY and L.D. VITT, Ed.), pp.63-93, Martinus Njhoff, Boston MA.
25 J.W.M. Bertrand and J. Wijngaard (1986) The Structuring of Production Control Systems, Int.J.Operations and Production Management 6, No.2, pp.5-20.
26 B. Wilson (1984) Systems: Concepts Methodology and Applications John Wiley, Chichester.
27 G.D. Scudder and T.R. Hoffmann (1986) The Use of Value-Based Rules to Avoid Shop Disruption. OMEGA 14, No.3, pp.259-264.
28 H. Ingham (1971) Balancing Sales and Production - Models of Typical Business Policies. Management Publications Ltd, London.
29 R. van Deirdonck and J.G. Miller (1981) Designing Production Planning and Control Systems. Production and Inventory Management 22, No.4, pp.59-71.
30 M.K. Starr (1971) Systems Management of Operations, Prentice-Hall, London.
31 M.A. Hegseth (1984) The Challenge of Operation Yield. Production and Inventory Management 25, No.1, pp.4-9.
32 M.S. Fox, B. Allen, G. Strohm (1982) Job-Shop Scheduling: An Investigation in Constraint-Directed Reasoning. Presented at AAAI-82: The National Conference on Artificial Intelligence, Pittsburgh, August 1982.
33 T.J. Grant (1986) Lessons for O.R. from A.I.: A Scheduling Case Study. J.Opl.Res.Soc, 37, No.1, pp.41-57.
34 J.N. Gupta (1979) A Review of Flowshop Scheduling Research. In Disaggregation: Problems in Manufacturing and Service Organizations (L.P. RITZMAN, L.J. KRAJEWSKI, W.L.BERRY, S.H.GOODMAN, S.T. HARDY and L.D. VITT, Ed.), pp.363-388, Martinus Njhoff, Boston MA.
35 E.S. Buffa and R.K. Sarin (1987) Modern Production/Operations Management. Wiley, New York.
36 C.J. Constable and C.C. New (1976) Operations Management. Wiley, London.
37 T. Hill (1985) Manufacturing Strategy Macmillan, London.
38 C.A. Voss (1982) United Kingdom Manufacturing Performance. Analysis and Implications. Engineering Costs and Production Economics 6, pp.247-258.
39 W.B. Miller (1981) Total Operations Management. Production and Inventory Management 22, No.4, pp.90-98.
40 B.G. Dale and D. Russell (1983) Production Control Systems for Small Group Production. OMEGA 11, No.2, pp.175-185.
41 W.R. Ashby (1956) An Introduction to Cybernetics. Methuen, London.
42 A.S. Marucheck and M.K. McClelland (1986) Strategic Issues in Make-To-Order Manufacturing, Production and Inventory Management 27, No.2, pp.82-95.
43 T.G. Schmitt, T. Klastorin and A. Shtub (1985) Production Classification System: Concepts, Models and Strategies, Int.J.Prod.Res. 23, No.3, pp.563-578.
44 A.W. Todd (1987) MRP: Whole Answer, Part Answer or No Answer? In Handbook of Production and Inventory Control (N.V. REINFIELD, Ed.), pp.344-360. Prentice-Hall, NJ.
45 S.M. Seward, S.G. Taylor and S.F. Bolander (1985) Progress in Integrating and Optimizing Production Plans and Schedules, Int.J.Prod.Res. 23, No.3, pp.609-624.
46 T.J. Hill (1987) Teaching and Research Directions in Production/Operations Management: The Manufacturing Sector, Int.J. Operations and Prod. Mgt 7, No.4, pp.5-12.

47 F.L. Sawyer, A. Charlesby, T.E. Easterfield and E.E. Treadwell (1989) Reminiscences of Operational Research in World War II by Some of its Practitioners. J.Opl.Res.Soc. 40, No.2, pp.155-136.
48 J. Rosenhead (1989) Operational Research at the Crossroads: Cecil Gordon and the Development of Post-War OR. J.Opl.Res.Soc. 40, No.1, pp.3-28.
49 J.C. Ranyard (1988) A History of OR and Computing. J.Opl.Res.Soc. 39, No.12, pp.1073-1086.
50 R.L.Ackoff (1979) The Future of Operational Research is Past. J.Opl.Res.Soc. 30, No.2, pp.93-104.
51 R.H. Colcutt (1965) The First 20 Years O.R. British Iron and Steel Research Association.
52 S. Bonder (1979) Changing the Future of Operational Research. In Operational Research '78: Proceedings of the Eighth IFORS International Conference on Operational Research (K.B. HALEY, Ed.), pp.62-79, North Holland, Amsterdam.
53 P.Sadler (1978) O.R. and the Transition to a Post-Industrial Society. J.Opl.Res.Soc. 29, No.1, pp.1-7.
54 R.E.D. Woolsey (1979) Pragmatism Triumphant or Past Sophistication and Future Elegance. In Operational Research '78: Proceedings of the Eighth IFORS International Conference on Operational Research (K.B. HALEY, Ed.), pp.80-86, North Holland, Amsterdam.
55 A. Jackson (1989) Knowledge-Based Systems Applied to Scheduling. Presented at the Seminar on Expert Systems in Scheduling, March 1989, Teeside Laboratories, British Steel.
56 Council of the OR Society (1986) Report of the Commission on the Future Practice of Operational Research. J.Opl.Res.Soc. 37, No.9, pp.829-886.
57 H.M. Weingartner (1987) The Changing Character of Management Science. OMEGA 15, No.3, pp.257-262.
58 S. Eilon (1980) The Role of Management Science, J.Opl.Res.Soc 31, No.1, pp.17-28.
59 N.R. Tobin, K. Rapley and W. Teather (1980) The Changing Role of O.R., J.Opl.Res.Soc. 31, No.4, pp279-288.
60 C. Eden, S. Jones and D. Sims (1983) Messing About in Problems. Pergamon Press, Oxford.
61 P.G. Bennett and C.S. Huxham (1982) Hypergames and What They Do: a `Soft' OR Approach. J.Opl.Res.Soc. 33, No.1, pp41-50.
62 R.O. Mason and I.I. Mitroff (1981) Challenging Strategic Planning Assumptions, Wiley, New York.
63 P. Morse (1953) Trends in Operations Research, Opns. Res. 1 pp.159-165.
64 M.R. Dando and R.G. Sharp (1978) Operational Research in the UK in 1977: The Causes and Consequences of a Myth? J.Opl.Res.Soc 29, No.10, pp.939-949.
65 R.N. Woolley and M. Pidd (1981) Problem Structuring - A Literature Review, J.Opl.Res.Soc. 32, No.3, pp197-206.
66 C. Schwenk and H. Thomas (1983) Formulating the Mess: The Role of Decision Aids in Problem Formulation. OMEGA 11, No.3, pp.239-252.
67 J. Tait (1988) The Role of Values in Quantitative Decision-Making. J.Opl.Res.Soc. 39, No.7, pp.669-674.
68 S.C. Ward (1989) Arguments for Constructively Simple Models. J.Opl.Res.Soc. 40, No.2, pp.141-153.
69 M.C. Jackson (1987) Present Positions and Future Prospects in Management Science. OMEGA 15, No.6, pp.455-466.
70 M.C. Jackson and P. Keys (1984) Towards a System of Systems Methodologies. J.Opl.Res.Soc 35, No.6, pp.473-486.

71 P. Robins (1985) The Methodology of Operational Research. In Managing with Operational Research (J. KIDD, Ed.), pp.17-31, Philip Allan, Oxford.
72 K.B. Haley (1984) Technique Maketh O.R. J.Opl.Res.Soc 35, No.3, pp.191-194.
73 S.Eilon (1975) Five Approaches to Aggregate Production Planning. AIIE Transactions 7, No.2, pp.118-131.
74 S.M. Alexander (1987) An Expert System for the Selection of Scheduling Rules in a Job Shop, Computers ind. Engng 12, No.3, pp.167-171.
75 A.P. Muhlemann, A.G. Lockett and C.K. Farn (1982) Job Shop Scheduling Heuristics and Frequency of Scheduling. Int.J.Prod. Res. 20, No.2, pp.227-241.
76 B.R. Fox and K.G. Kempf (1985) Complexity, Uncertainty and Opportunistic Scheduling. Presented at The Second Conference of Artificial Intelligence Applications, Miami, December 1985.
77 B.G. Kingsman (1983) The Effect of Payment Rules on Ordering and Stockholding in Purchasing, J.Opl.Res.Soc. 34, pp.1085-1098.
78 L.P. Ritzman, B.E. King and L.J. Krajewski (1984) Manufacturing Performance - Pulling the Right Levers. Harvard Business Review 62, March 1984, pp.143-152.
79 E.S. Buffa (1981) Research in Operations Management. Production and Inventory Management 22, No.3, pp.57-69.
80 S. Eilon (1987) O.R. is Not Mathematics. OMEGA 15, No.2, pp.87-92.
81 S. Eilon (1985) Management Assertions and Aversions. Pergamon Press, Oxford.
82 M.C. Jackson (1987) New Directions in Management Science. In New Directions in Management Science (M.C. JACKSON and P. KEYS, Ed.), pp.133-164. Gower, Aldershot.
83 H. Mintzberg (1979) Beyond Implementation: An Analysis of the Resistance to Policy Analysis. In Operational Research '78: Proceedings of the Eighth IFORS International Conference on Operational Research (K.B. HALEY, Ed.), pp.106-162, North Holland, Amsterdam.
84 G.R. Lindsey (1979) Looking Back Over the Development and Progress of Operational Research. In Operational Research '78: Proceedings of the Eighth IFORS International Conference on Operational Research (K.B. HALEY, Ed.), pp.13-31, North Holland, Amsterdam.
85 J.S. Oakland and A. Sohal (1987) Production Management Techniques: A Proposed Methodology for Overcoming Barriers to Acceptance, Int.J.Operations and Prod. Mgt 7, No.4, pp23-34.
86 R. Tomlinson (1984) Rethinking the Process of Systems Analysis and Operational Research: From Practice to Precept - and Back Again. In Rethinking the Process of Operational Research and Systems Analysis (R. TOMLINSON and I. KISS, Ed.), pp205-222, Pergamon Press, Oxford.
87 S. Eilon (1977) Who Says Production Control is Dead? OMEGA 5, No.2, pp.107-113
88 S. Eilon (1979) Production scheduling. In Operational Research '78: Proceedings of the Eighth IFORS International Conference on Operational Research (K.B. HALEY, Ed.), pp.237-266, North Holland, Amsterdam.
89 J.L. Coker (1985) Analysing Production Switching Heuristics for Aggregate Planning Models via an Application. Production and Inventory Management 26, No.4, pp.1-13.

90 P. Duchessi (1987) The Conceptual Design for a Knowledge-Based System as Applied to the Production Planning Process. In Expert Systems for Business (B.G. SILVERMAN, Ed.), pp.163-194, Addison-Wesley, Reading MA.
91 V. Gilgeous (1987) Aggregate Planning in UK Manufacturing Companies. Int.J. Operations and Production Management 7, No.1, pp.30-61.
92 J.S. Oakland and A. Sohal (1987) Production Management Techniques in UK Manufacturing Industry: Usage and Barriers to Acceptance. Int.J. Operations and Production Management 7, No.1, pp.8-37.
93 E.L. Heard (1982) Bridging the Gap Between Theory and Practice in Production and Inventory Control. Engineering Costs and Production Economics 6, pp.119-129.
94 C. Eden (1987) Problem-Solving or Problem-Finishing? In New Directions in Management Science (M.C. JACKSON and P. KEYS, Ed.), pp.97-107. Gower, Aldershot.
95 P.B. Checkland (1983) Operational Research and the Systems Movement. J. Opl Res Soc 34 pp.661-675.
96 N.J. Butterworth (1989) Giving up `The Smoke': A Major Institution Investigates Alternatives to being Sited in the City. J.Opl.Res.Soc. 40, No.8, pp.711-717.
97 J. Rosenhead (1989) Diverse Unity: the Principles and Prospects for Problem Structuring. In Rational Analysis for a Problematic World (J. ROSENHEAD Ed.) pp.341-358. John Wiley, Chichester.
98 R.L. Ackoff (1979) Resurrecting the Future of Operational Research. J.Opl.Res.Soc. 30, No.3, pp.189-199.
99 C. Eden and C. Huxham (1988) Action-Oriented Strategic Management. J.Opl.Res.Soc. 39, No.10, pp.889-899.
100 C. Eden (1989) Using Cognitive Mapping for Strategic Options Development and Analysis. In Rational Analysis for a Problematic World (J. ROSENHEAD Ed.) pp.21-42. John Wiley, Chichester.
101 P.B. Checkland (1988) Future Directions for the Development of Systems Ideas. Presented at the Systems Conference Lancaster, July 1988.
102 P.B. Checkland (1989) Soft Systems Methodology. In Rational Analysis for a Problematic World (J. ROSENHEAD Ed.) pp.71-100. John Wiley, Chichester.
103 P.B. Checkland (1988) Soft Systems Methodology: An Overview. Journal of Applied Systems Analysis 15, pp.27-30.
104 P.B. Checkland (1989) An Application of Soft Systems Methodology. In Rational Analysis for a Problematic World (J. ROSENHEAD Ed.) pp.101-120. John Wiley, Chichester.
105 I. von Bulow (1989) The Bounding of a Problem Situation and the Concept of a System's Boundary in Soft Systems Methodology. Journal of Applied Systems Analysis 16, pp.35-41.
106 E.M. Goldratt (1988) Computerized Shop Floor Scheduling. Int.J. Prod.Res. 26, No.3, pp.443-455.
107 N.V. Reinfield (1987) Computers. In Handbook of Production and Inventory Control (N.V. REINFIELD, Ed.), pp.14-21, Prentice-Hall, NJ.
108 S.A. Melnyk (1985) MRP II: The Early Returns Are In. Production and Inventory Management 26, No.1, pp.124-137.
109 J. Orlicky (1975) Material Requirements Planning, McGraw-Hill, New York.
110 E.A. Haglund (1981) Primary Metals - We're Different. Production and Inventory Management 22, No.3, pp.21-26.

111 C.R. Fiora and J.L. Pitzer (1986) An Inventory Management Tool for a Basic Industry, Production and Inventory Management 27, No.1, pp.108-114.
112 E.C. Etienne (1983) MRP May Not Be Right for You: At Least Not Yet. Production and Inventory Management 24, No.3, pp.33-46.
113 G. Plenert and T.D. Best (1986) MRP, JIT and OPT: What's "Best"? Production and Inventory Management 27, No.2, pp.22-29.
114 T.E. Vollmann (1986) OPT as an Enhancement to MRP II, Production and Inventory Management 27, No.2, pp.38-47.
115 H. Andersson, S. Axsater and H. Jonsson (1981) Hierarchical Material Requirements Planning. Int.J.Prod.Res. 19, No.1, pp.45-47.
116 P.J. Billington, J.O. McClain and L.J. Thomas (1983) Mathematical Programming Approaches to Capacity-Constrained MRP Systems: Review, Formulation and Problem Reduction. Management Science 29, No.10, pp.1126-1141.
117 L.J. Krajewski and L.P. Ritzman (1977) Disaggregation in Manufacturing and Service Organizations: Survey of the Problems and Research. Decision Sciences 8, No.2, pp.1-17.
118 D. Swann (1986) Using MRP for Optimized Schedules (Emulating OPT), Production and Inventory Management 27, No.2, pp.30-37.
119 J.A. Leahy (1986) Integrating Just-In-Time and Optimized Production Technology. Presented at the 1986 Fall Industrial Engineering Conference, December 1986, Boston MA.
120 J. Browne (1988) Production Activity Control - A Key Aspect of Production Control. Int.J.Prod.Res. 26, No.3, pp.415-427.
121 C.A. Voss (1984) Production/Operations Management - A Key Discipline and Area for Research. OMEGA 12, No.3, pp.309-319.
122 H. Tsubone and M Sugawara (1987) A Hierarchical Production Planning System in the Motor Industry, OMEGA 15, No.2, pp113-120.
123 S.G. Taylor, S.M. Seward, S.F. Bolander and R.C. Heard (1981) Process Industry Production and Inventory Planning Framework: A Summary.Production and Inventory Management 22, No.1, pp.15-33.
124 K.D. Barber and R.H. Hollier (1986) The use of numerical analysis to classify companies according to production control complexity. Int.J.Prod.Res. 24, No.1, pp203-222.
125 D.J Bennett (1981) Operations Planning and Control. In Operations Management in Practice (C.D. LEWIS, Ed,), Philip Allan, Oxford.
126 D.K. Corke (1977) Production Control in Engineering. Edward Arnold, London
127 R. Stewart (1971) How Computers Effect Management. Macmillan, London.
128 A. Rao (1985) Manufacturing Systems Implementation: Agenda for Top Management. Production and Inventory Management 26, No.1, pp.88-101.
129 W.N. Norquist (1987) Comment on Computers. In Handbook of Production and Inventory Control (N.V. REINFIELD, Ed.), pp.21-22, Prentice-Hall, NJ.
130 R.J. Boaden and B.G.Dale (1986) What is Computer-Integrated Manufacturing? Int.J.Operations and Production Management 6, No.3, pp.30-37.
131 L.G. Chalmet (1987) Increasing Your Competitive Advantage Through CIM. In Computer Integrated Manufacturing (K. RATHMILL and P MACCONAILL,Ed.), pp.49-57, Springer Verlag, Berlin.
132 R.J. Boaden and B.G. Dale (1986) Development of a Model for Use in Planning for Computer-Integrated Manufacturing, Int.J. Operations and Production Management 6, No.5, pp.8-19.

133 A. DeMeyer and K. Ferdows (1985) Integration of Information Systems in Manufacturing. Int.J.Operations and Production Management 5, No.2, pp.5-12.
134 A. DeMeyer (1987) The Integration of Information Systems in Manufacturing. OMEGA 15, No.3, pp.229-238.
135 P.A. Newman (1988) Scheduling in CIM Systems. In Artificial Intelligence: Implications for CIM (A. KUSIAK, Ed.), pp.361-402, Springer-Verlag, Berlin.
136 M.S. Fox (1987) Constraint-Directed Search: A Case Study of Job-Shop Scheduling, Pitman, London.
137 H. van Dyke-Parunak, B.W. Irish, J. Kindrick and P.W. Loso (1985) Fractal Actors for Distributed Manufacturing Control. Presented at The Second Conference of Artificial Intelligence Applications, Miami, December 1985.
138 P.Burke (1988) Scheduling in a Dynamic Environment. Technical Report AISL38, Dept of Computer Science, Strathclyde University.
139 I. Buchanan, P. Burke, J. Costello and P. Prosser (1988) A Distributed Asynchronous Scheduler. Technical Report AISL36, Dept of Computer Science, Strathclyde University.
140 I. Buchanan (1989) A Distributed Asynchronous Scheduler and its Application on an Aluminium Mill. Presented at the Seminar on Expert Systems in Scheduling, March 1989, Teeside Laboratories, British Steel.
141 N.R. Greenwood (1987) CIM: What will be its Impact During the Next Few Years. In Computer Integrated Manufacturing (K. RATHMILL and P MACCONAILL,Ed.), pp.1-7, Springer Verlag, Berlin.
142 J.R. King (1976) The Theory-Practice Gap in Job-Shop Scheduling. The Production Engineer 55, No.3, pp138-143.
143 J.R. King (1979) Scheduling and the Problem of Computational Complexity. OMEGA 7, No.3, pp.233-240.
144 M.P. Meleton (1986) OPT - Fantasy or Breakthrough? Production and Inventory Management 27, No.2, pp.13-21.
145 P.J. O'Grady and M.A. Azoza (1987) An Adaptive Approach to Shop Loading. OMEGA 15, No.2, pp.121-128.
146 S. Shimoyashiro, K. Isoda and H. Awane (1984) Input Scheduling and Load Balance Control for a Job Shop, Int.J.Prod.Res. 22, No.4, pp.597-605.
147 W. Bechte (1988) Theory and Practice of Load-Oriented Manufacturing Control. Int.J.Prod.Res. 26, No.3, pp.375-395.
148 H.C. Meal (1984) Putting Production Decisions Where They Belong. Harvard Business Review 62, March, 1984, pp.103-111.
149 J.H. Blackburn (1984) Considerations in Choosing Shop Scheduling Algorithms. Production and Inventory Management 25, No.4, pp.107-125.
150 J.N. Njus (1987) Resource Requirements Planning - It Works!. In Handbook of Production and Inventory Control (N.V. REINFIELD, Ed.), pp.153-164. Prentice-Hall, NJ.
151 R.W. White (1986) Is it Time for a Physical Exam of your Material Requirement Planning System? Production and Inventory Management 27, No.3, pp.24-29.
152 D.A. Latham (1981) Are You Among MRP's Walking Wounded? Production and Inventory Management 22, No.3, pp.33-41.
153 R.S. Lee (1983) Material Requirements Planning and the Organization. Production and Inventory Management 24, No.3, pp.103-114.
154 P.C. Galvin (1986) Visions and Realities: MRP as "System", Production and Inventory Management 27, No.3, pp.91-95.

155 J.A.G. Krupp (1986) MRP Re-Implementation, Production and Inventory Management 27, No.4, pp.73-81.
156 M.H. Safizadeh and F Raafat (1986) Formal/Informal Systems and MRP Implementation, Production and Inventory Management 27, No.1, pp.115-121.
157 S.D. Deaton (1984) Back to Basics, Production and Inventory Management 25, No.1, pp.55-60.
158 L.P. Kneppelt (1981) Implementing Manufacturing Resource Planning: Difficulty of the Task. Production and Inventory Management 22, No.2, pp.59-77.
159 J.R. Minifie and R.A. Davis (1986) Survey of MRP Nervousness Issues, Production and Inventory Management 27, No.3, pp.111-119.
160 J.F. Proud (1981) Controlling the Master Schedule, Production and Inventory Management 21, No.2, pp.78-90.
161 P.L. Abad and D.J. Sweeney (1982) Decentralized Planning with an Independent Marketing-Production System OMEGA 10, No.4, pp.353-359.
162 A.C. Hax (1976) The Design of Large Scale Logistics Systems: A Survey and an Approach. In Modern Trends in Logistics Research (W.H. MARLOW, Ed.), pp.59-96, MIT Press, Cambridge MA.
163 G.R. Bitran and A.C. Hax (1977) On the Design of Hierarchical Production Planning Systems. Decision Sciences 8, No.2, pp.28-55.
164 C-H. Chung and L.J. Krajewski (1987) Interfacing Aggregate Plans and Master Production Schedules via a Rolling Horizon Feedback Procedure. OMEGA 15, No.5, pp.401-409.
165 B. Dodin (1987) The Use of Demand Information in Selecting Production Plans and Schedules. OMEGA 15, No.6, pp.477-488.
166 J. Wijngaard (1982) On Aggregation in Production Planning. Engineering Costs and Production Economics 6, pp.259-265.
167 R.W. Haesslar (1979) The Disaggregation Problem in the Paper Industry. In Disaggregation: Problems in Manufacturing and Service Organizations (L.P. RITZMAN, L.J. KRAJEWSKI, W.L.BERRY, S.H.GOODMAN, S.T. HARDY and L.D. VITT, Ed.), pp.257-270, Martinus Njhoff, Boston MA.
168 A.C. Hax and H.C. Meal (1975) Hierarchical Integration of Production Planning and Scheduling. In Logistics (M.A. GEISLER, Ed.), North-Holland, Amsterdam.
169 S.P. Bradley, A.C. Hax and T.L. Magnanti (1977) Applied Mathematical Programming, Addison-Wesley, Reading MA.
170 G.G. Stephenson (1970) A Hierarchy of Models for Planning in a Division of I.C.I. Opl.Res.Quarterly 21, No.2, pp.221-245.
171 E.S. Buffa and J.S. Dyer (1981) Management Science/Operational Research: Model Formulation and Solution Methods. John Wiley, New York.
172 C. Perry and W.J. Preston (1986) Production Planning with Linear Programming: From Textbook to Factory. OMEGA 14, No.3, pp.233-238.
173 S.K. Jain, K.L. Stoot and E.G. Vasold (1978) Orderbook Balancing Using a Combination of Linear Programming and Heuristic Techniques. INTERFACES 9, No.1, pp52-67.
174 R. Khorramshahgol and H.M. Steiner (1988) Resource Analysis in Project Evaluation: A Multicriteria Approach. J.Opl.Res.Soc. 39, No.9, pp.795-803.
175 R.E. Box and D.G. Herbe (1988) A Scheduling Model for LTV Steel's Cleveland Works' Twin Strand Continuous Slab Caster, INTERFACES 18, No.1, pp.42-56.
176 R.L. Ackoff (1955) Production and Inventory Control in a Chemical Process. In Operational Research Society of America 3, pp.319-338.

177 K.D. Tocher and A.P. Amiry (1963) New Developments in Simulation. Presented at the Third International Conference on Operational Research, July 1963, Oslo.
178 J.A. Farmer and R.H. Collcutt (1965) Experience of Digital Simulations in a Large O.R. Group. In Digital Simulation in Operational Research (S.H. HOLLINGDALE, Ed.), pp.166-180, American Elsevier, New York.
179 A.P. Amiry (1965) The Simulation of Information Flow in a Steelmaking Plant. In Digital Simulation in Operational Research (S.H. HOLLINGDALE, Ed.), pp.157-165, American Elsevier, New York.
180 F. Dewhurst and B.G. Dale (1983) A Microcomputer Simulation of an Automated Non Ferrous Sand Casting Foundry. OMEGA 11, No.6, pp.599-606.
181 K.M. Thinnes and V. Kachitvichyanukul (1989) Simulation of Printed Circuit-Board Manufacturing, J.Opl.Res.Soc. 40, No.7, pp.643-647.
182 S. Economides and F. Cunningham (1987) Microcomputer Simulation Improves Production Scheduling for Microelectronics Manufacturers, Journal of Manufacturing Systems 6, No.4, pp.267-275.
183 L.F. Gelders and F.V. van Steelandt (1980) Design and Implementation of a Production Planning System in a Rolling Mill: A Case Study. AIIE Transactions 12, No.1, pp.54-58.
184 R.M. Narchal (1988) A Simulation Model for Corporate Planning in a Steel Plant. Eur. J. Opl. Res 34, pp.282-296.
185 J.Reitman (1981) Computer Simulation Applications: Discrete-Event Simulation for Synthesis and Analysis of Complex Systems, Robert E. Krieger, Malabar, Florida.
186 A. Hulme (1985) Simulation: Why, When and How. In Managing with Operational Research (J. KIDD, Ed.), pp.214-230, Philip Allan, Oxford.
187 M. Robinson (1990) Simulating a Steel Plant - Experiences in the Development and Use of a VIS Model. Presented at the 6th Young OR Conference, Warwick, March 1990.
188 P. Klahr and W.S. Faught (1980) Knowledge-Based Simulation. Presented at AAAI-80: The First National Conference on Artificial Intelligence, Stanford University, August 1980.
189 D.H. Goodman, D.W. Balmer and G.I. Doukidis (1987) Interfacing Expert Systems and Simulation for Job-Shop Production Scheduling. Presented at the Third Expert System Conference October 1987, London.
190 K. Tangen and U Wretling (1987) Intelligent Front Ends to Numerical Simulation Programs. In Research and Development in Expert Systems 3 (M.A. BRAMER, Ed.), pp.254-265, University Press, Cambridge.
191 R.A. Herrod and J. Rickel (1986) Knowledge-Based Simulation of a Glass Annealing Process: An AI Application in the Glass Industry. Presented at AAAI-86, the Fifth National Conference on Artificial Intelligent, Philadelphia, August 1986.
192 M.A. Bramer (1988) Expert Systems in Business: A British Perspective, Expert Systems 5, No.2, pp104-117.
193 C.B. Stabell (1983) A Decision-Oriented Approach to Building DSS. In Building Decision Support Systems (J.L. BENNETT, Ed.), pp.221-260, Addison-Wesley, Reading MA.
194 P. Holroyd, G. Mallory, D.H.R. Price and J.A. Sharp (1985) Developing Expert Systems for Management Applications, OMEGA 13, No.1, pp1-11.
195 P.H. Winston (1987) Artificial Intelligence: A Perspective. In AI in the 1980s and Beyond - An MIT Survey (W.E.L. GRIMSON and R.S. PATIL, Ed.), pp.1-11, The MIT Press, Cambridge MA.

196 J.Bader, J.Edwards, C.Harris-Jones and D.Hannaford (1988) Practical Engineering of Knowledge-Based Systems. Information and Software Technology 30, 266-277.
197 E. Feigenbaum, P. McCorduck and H.P. Nii (1988) The Rise of the Expert Company. Macmillan, London.
198 D.G. Bobrow, S. Mittal and M.J. Stefik (1986) Expert Systems: Perils and Promise, Comms. of the ACM 29, No.9, pp.880-894.
199 T.R. Addis, (1987) The Role of "Explanation" in Knowledge Elicitation, Int.J.Systems Research and Info.Science 2, pp.101-110.
200 K.L. Modesitt (1987) Experts: Human and Otherwise. Presented at the Third Expert System Conference October 1987, London.
201 J.A. Campbell (1984) Three Uncertainties of AI. In Artificial Intelligence: Human Effects (M.YAZDANI and A. NARAYANAN, Ed.), pp.249-273, Ellis Horwood, Chichester.
202 P.H. Winston (1987) The Commercial Debut of Artificial Intelligence. In Application of Expert Systems (J.R. QUINLAN, Ed.), pp.3-20, Addison-Wesley, Sidney.
203 S.A. Hayward (1985) Is a Decision Tree an Expert System? In Research and Development in Expert Systems (M.A. BRAMER, Ed.), Cambridge University Press, Cambridge.
204 B.G. Buchanan (1986) Expert Systems: Working Systems and the Research Literature, Expert Systems 3, No.1, pp.32-51.
205 D.A.Waterman (1986) A Guide to Expert Systems. Addison-Wesley, Reading MA.
206 A.S. Pollitt (1986) Reducing Complexity by Rejecting the Consultation Model as a Basis for the Design of Expert Systems, Expert Systems 3, No.4, pp.234-238.
207 R.T. Hartley (1981) How Expert Should Expert Systems Be? Presented at IJCAI-7, the Seventh Joint Conference on Artificial Intelligence, August 1981, Vancouver.
208 R. Davis (1987) Knowledge-Based Systems: The View in 1986. In AI in the 1980s and Beyond - An MIT Survey (W.E.L. GRIMSON and R.S. PATIL, Ed.), pp.13-41, The MIT Press, Cambridge MA.
209 P. Szolovits (1987) Expert System Tools and Techniques: Past, Present and Future. In AI in the 1980s and Beyond - An MIT Survey (W.E.L. GRIMSON and R.S. PATIL, Ed.), pp.43-74, The MIT Press, Cambridge MA.
210 R. Worden (1988) Processes of Knowledge and Software. Presented at the BCS Specialist Group on Expert systems, Dec 1988, Brighton.
211 A.C. Beerel (1987) Expert Systems: Strategic Implications and Applications. Ellis Horwood, Chichester.
212 J. Debenham (1987) Expert Systems: An Information Processing Perspective. In Application of Expert Systems (J.R. QUINLAN, Ed.), pp.200-217, Addison-Wesley, Sidney.
213 W. Little, H.W. Fowler, J. Coulson and C.T. Onions (1972) The Shorter Oxford English Dictionary. Oxford University Press, London.
214 G. Ostler (1948) The Little Oxford Dictionary of Current English. Clarendon Press, London.
215 B. Chandrasekaran (1983) Expert Systems: Matching Techniques to Tasks. In Artificial Intelligence Applications for Business (W. REITMAN, Ed.), pp.41-64, Ablex Publishing Corp., Norwood NJ.
216 W.B. Rauch-Hindin (1985) Artificial Intelligence in Business Science and Industry. Volume Two - Applications. Prentice-Hall, Englewood Cliffs NJ.
217 M.R. Genesereth (1983) An Overview of Meta-Level Architecture. Presented at AAAI-83: The National Conference on Artificial Intelligence, Washington, August 1983.

218 W. Reitman (1983) Managing the Acquisition of AI capability: Some Observations Suggestions and Conclusions. In Artificial Intelligence Applications for Business (W. REITMAN, Ed.), pp.325-330, Ablex Publishing Corp., Norwood NJ.
219 D. McDermott, M.M. Waldrop, R.Schank, B. Chandrasekaran and J. McDermott (1984) The Dark Ages of AI: A Panel Discussion at AAAI-84, The AI Magazine 5, No.3, pp.122-134.
220 A.W. Smith (1987) The ICL Expert Mainframe Configurer (S39XC). Presented at the Third Expert System Conference October 1987, London.
221 P. Harmon (1988) Expert Systems in Use. In The Rise of the Expert Company (E. FEIGENBAUM, P. MCCORDUCK and H.P. NII, Ed.), pp.273-315, Macmillan, London.
222 J. Mingers and J. Adlam (1989) Where are the "Real" Expert Systems? OR Insight 2, No.3, pp.6-9.
223 Various (1989) Letters to the Editor: Where are the "Real" Expert Systems? OR Insight 2, No.4, pp.23-25.
224 J.J. Kanet and H.H. Adelsberger (1987) Expert Systems in Production Scheduling, Euro.J.Opl.Res. 29, pp51-59.
225 M.S. Steffen (1986) A Survey of Artificial Intelligence-Based Scheduling Systems. Presented at the 1986 Fall Industrial Engineering Conference December 1986, Boston MA.
226 A. Kusiak and M. Chen (1988) Expert Systems for Planning and Scheduling Manufacturing Systems, Euro.J. of Opl.Res. 34, pp.133-130.
227 A. Kraft (1987) Artificial Intelligence: Next Generation Solutions. In Intelligent Knowledge-Based Systems: An Introduction (T, O'SHEA, J. SELF and G. THOMAS, Ed.), pp.110-121. Harper & Row, London.
228 E.M. Fisher (1986) Building AI Behind Closed Doors, Datamation 32, No.15, pp46-50.
229 R. Mayer (1985) Artificial Intelligence Applied to Manufacturing, CIM Review, Fall 1985, pp.25-29.
230 S.M. Alexander (1985) Knowledge-Based Expert Systems: Their Application to Production Management, Production and Inventory Management, No.4, pp.109-113.
231 R.W. Blanning (1987) A Survey of Issues in Expert Systems for Management. In Expert Systems for Business (B.G. SILVERMAN, Ed.), pp.24-39, Addison-Wesley, Reading MA.
232 J.T. Edmundson, D.P. Jenkins and J, Mortimer (1986) Multiple Burner Stoichiometry Control and Efficiency Optimization by Means of a Teleogical, Heuristic On-Line Expert System, IChemE SYMPOSIUM SERIES No.105, pp.47-60.
233 A.L. Kidd and M.B. Cooper (1985) Man-Machine Interface Issues in the Construction and Use of Expert Systems, Int.J. Man-Machine Studies 22, pp.91-102.
234 L.Steels (1986) Second Generation Expert Systems. In Research and Development in Expert Systems 3 (M.A. BRAMER, Ed.), pp.175-183., University Press, Cambridge.
235 R. Khanna and R.L. Moore (1986) Expert Systems Involving Dynamic Data for Decisions. Presented at Second Expert System Conference, London, October 1986.
236 P. Harmon, R. Maus and W. Morrissey (1988) Expert Systems: Tools and Applications. John Wiley, New York.
237 M. O'Neill and A. Morris (1989) Expert Systems in the United Kingdom: An Evaluation of Development Methodologies, Expert Systems 6, No.2, pp.90-99.

238 L.Steels and J.A. Campbell (1985) Areas of Application. In Progress in Artificial Intelligence (L. STEELS and J.A. CAMPBELL, Ed.), pp.255-256, Ellis Horwood, Chichester.
239 J. Hewett (1987) Commercial Expert Systems in North America. Presented at the Third Expert System Conference October 1987, London.
240 M. Schindler (1984) Artificial Intelligence Begins to Pay Off with Expert Systems for Engineering, Electronic Design, 9th August, 1984, pp.106-146.
241 A. d'Agapeyeff and C.J.B. Hawkins (1987) Report to the Alvey Directorate on the Second Short Survey of Expert Systems in UK Business, Alvey News, August 1987, Whole Issue.
242 A. Goodall (1985) The Guide to Expert Systems. Learned Information Systems (Europe) Ltd, Oxford.
243 J. Hewett (1987) Commercial Progress in Europe. Presented at the Third Expert System Conference October 1987, London.
244 W. Reitman (1982) Applying Artificial Intelligence to Decision Support: Where do the Good Alternatives Come From? In Decision Support Systems: Proceedings of the NYU Symposium on DSS (M.J. GINZBERG, W. REITMAN and E.A. STOHR, Ed.), pp.155-174, North-Holland, Amsterdam.
245 D. Bedford (1989) Traditional Techniques and Expert Systems - A Comparison. Presented at the Seminar on Expert Systems in Scheduling, March 1989, Teeside Laboratories, British Steel.
246 M. Merry (1985) Expert Systems - Some Problems and Opportunities. In ES 85. Proceedings of th Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems (M.MERRY, Ed.), Cambridge University Press, Cambridge.
247 M.H. Richie (1986) An Evaluation of Expert System Development Tools, Expert Systems 3, No.3, pp.166-182.
248 S.M. Weiss and C.A. Kulikowski (1983) A Practical Guide to Designing Expert Systems. Chapman and Hall, London.
249 B.G. Buchanan (1982) New Research on Expert Systems. In Machine Intelligence 10 (J.E. HAYES, D. MICHIE and Y-H. PAO, Ed.), pp.269-299. Ellis Horwood, Chichester.
250 D.C. Berry and D.E. Broadbent (1987) Expert Systems and the Man-Machine Interface Part Two: The User Interface, Expert Systems 4, No.1, pp.18-27.
251 A. Basden (1983) On the Application of Expert Systems, Int.J.Man-Machine Studies 19, pp.461-477.
252 M. Schinder (1985) Expert Systems: Technical Forecast 1985, Electronic Design, Jan 10th, 1985, pp.111-134.
253 H. Takenouchi and Y Iwashita (1987) An Integrated Knowledge Representation Scheme for Expert Systems, Expert Systems 4, No.1, pp.31-43.
254 R.T. Hartley (1985) Representation of Procedural Knowledge for Expert Systems. Presented at The Second Conference of Artificial Intelligence Applications, Miami, December 1985.
255 B. Chandrasekaran (1986) Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design, IEEE Expert, Fall, 1986, pp.23-30.
256 N.A.D. Connell (1986) Expert Systems: Where are the Business Applications? Presented at 28th Annual Conference of the OR Society, Sept 1986.
257 A.Sloman (1986) Epistemology and Artificial Intelligence. In Expert Systems in the Micro-Electronic Age (D. MICHIE, Ed.), pp.235-241, Edinburgh University Press, Edinburgh.

258 N. Davies (1988) Expert Systems in Finance: A Personal View. Presented at the Fifth Young OR Conference, March 1988, Warwick.
259 G.R. Martins (1984) The Overselling of Expert Systems, Datamation 30, No.18, pp76-80.
260 D. Partridge (1988) To Add AI, or Not to Add AI? Presented at the BCS Specialist Group on Expert systems, Dec 1988, Brighton.
261 R.M. O'Keefe (1985) Expert Systems and Operational Research - Mutual Benefits, J.Opl.Res.Soc. 36, No.2, pp.125-129.
262 R.M. O'Keefe, V. Belton and T. Ball (1986) Experiences with Using Expert Systems in O.R. J.Opl.Res.Soc. 37, No.7, pp.657-668.
263 A.W. Pearson (1985) Speculations on the Future of Knowledge Engineering in Europe II. In Artificial Intelligence: Towards Practical Applications (T.BERNOLD and G.ABLERS, Ed.), pp.185-187, Elsevier Science (North Holland).
264 A. d'Agapeyeff (1989) The Experience Gained from Expert System Applications in Business. Seminar at Aston University, Feb 1989.
265 A. Bundy (1987) How to Improve the Reliability of Expert Systems. In Research and Development in Expert Systems IV, (D.S. MORALEE, Ed.), pp.3-19, Cambridge University Press, Cambridge.
266 Anon. (1987) Artificial Intelligence Is Finding Its way in KSC's Steel Mills/Offices at an Accelerated Pace. Kawasaki Steel Corp Newsletter, Nov-Dec 1987, pp.3-4.
267 B. Chandrasekaran (1985) Generic Tasks in Knowledge-Based Reasoning: Characterizing and Designing Expert Systems at the "Right" Level of Abstraction. Presented at The Second Conference of Artificial Intelligence Applications, Miami, December 1985.
268 M. Coombs and J. Alty (1984) Expert Systems: An Alternative Paradigm, Int.J.Man-Machine Studies 20, pp.21-43.
269 J. Cohen (1985) Describing Prolog by its Interpretation and Compilation, Comms. of the ACM 28, No.12, pp.1311-1324.
270 J. McDermott (1982) R1: A Rule-Based Configurer of Computer Systems, Artificial Intelligence 19, pp.39-88.
271 J. Bachant and J. McDermott (1984) R1 Revisited: Four Years in the Trenches, The AI Magazine 5, No.3, pp.21-32.
272 E. Soloway, J. Bachant and K. Jensen (1987) Assessing the Maintainability of XCON-in-RIME: Coping with the Problems of a VERY Large Rule-Base. Presented at AAAI-87: The Sixth National Conference on Artificial Intelligence, Seattle, July 1987.
273 M.A. Bramer (1986) Expert Systems in Britain: Progress and Prospects. In Research and Development in Expert Systems 3 (M.A. BRAMER, Ed.), pp.1-12, University Press, Cambridge.
274 F. Kluzniak and S. Szpakowicz (1984) Prolog - A Panacea? In Implementations of Prolog (J.A. CAMPBELL, Ed.), pp.71-84, Ellis Horwood, Chichester.
275 R. Kowalski (1987) Logic Programming in Artificial Intelligence and Software Engineering. In Intelligent Knowledge-Based Systems: An Introduction (T, O'SHEA, J. SELF and G. THOMAS, Ed.), pp.83-96, Harper & Row, London.
276 M.A. Bramer (1985) Expert Systems: The Vision and the Reality. In Research and Development in Expert Systems (M.A. BRAMER, Ed.), pp.1-12, Cambridge University Press, Cambridge.
277 F. Hayes-Roth, D.A. Waterman and D.B. Lenat (1983) An Overview of Expert Systems. In Building Expert Systems (F. HAYES-ROTH, D.A. WATERMAN and D.B. LENAT, Ed.), pp3-29, Addison-Weseley, London.
278 S. Hayward (1985) Diagnostic Expertise and its Use in Commercially Viable Expert Systems. In Advances in Artificial Intelligence (T. O'SHEA, Ed.), pp.71-75, Elsevier Science (North Holland).

279 S. Moralee (1987) Expert Systems - Some User Experience. In Intelligent Knowledge-Based Systems: An Introduction (T, O'SHEA, J. SELF and G. THOMAS, Ed.), pp.97-109, Harper & Row, London.
280 J. Efstathiou, V. Rajkovic and M. Bohanec (1986) Expert Systems and Rule Based Decision Support Systems. In Computer Assisted Decision Support (G. MITRA, Ed.), pp.165-174, Elsevier Science (North Holland).
281 R,M,G. Perkins, M. Pitt and A.E. Price (1986) Expert Systems: Practical Applications in a Traditional Industry. In Research and Development in Expert Systems 3 (M.A. BRAMER, Ed.), pp.231-241, University Press, Cambridge.
282 Anon. (1988) Luxembourg Steel Maker Cuts Wastage and Retains Its Experts' Knowledge with Diagnostic System, Intelligence: Int.J. on A.I. - Texas Instruments No.13, pp.1-3.
283 M. Philips, J. Loup-Gelard and A. Picca (1989) Knowledge Based Systems Applied to Scheduling. Presented at the Seminar on Expert Systems in Scheduling, March 1989, Teeside Laboratories, British Steel.
284 Anon. (1989) Spanish Power Station Diagnoses Disturbances to its Electrical Network in Real Time, Intelligence: Int.J. on A.I. - Texas Instruments No.16, pp.1-2.
285 P.J. Brew and J. Catlett (1987) SA: An Expert System for Trobleshooting a Smelter. In Application of Expert Systems (J.R. QUINLAN, Ed.), pp.174-183, Addison-Wesley, Sidney.
286 D. Leonard-Barton and J.J. Sviokla (1988) Putting Expert Systems to Work, Harvard Business Review 66, No.2, pp.91-99.
287 L.Sweet (1985) Artificial Intelligence in General Electric, The AI Magazine 6, No.3, pp.220-226.
288 S.M. Alexander (1987) The Application of Expert Systems to Manufacturing Process Control, Computers ind.Engng 12, No.4, pp.307-314.
289 S. Nishitomo, S. Hori, Y. Tokunaga and T. Masuda (1985) Recent Computer Control Technology in the Steel Industry, Hitachi Review 34, No.4, pp.194-200.
290 P.J. King and E.H. Mamdani (1977) The Application of Fuzzy Control Systems to Industrial Processes, Automatica 12, pp301-308.
291 B. Knight, R. Endersby and V.R. Voller (1984) The Use of Expert Systems in Industrial Control, Measurement and Control 17, Dec/Jan 1984/85, pp.409-413.
292 Anon. (1988) Expert System in Control at NKK Fukuyama Works Blast Furnace, Steel Times 66, April 1988, p207.
293 M.A. Rodger and J.S. Edwards (1989) A Problem-driven Approach to Expert System Development, J.Opl.Res.Soc. 40, No.12, pp.1069-1077.
294 Anon. (1987) Artificial Intelligence Applied to Production Control Systems. NKK News, June 1987, pp.2-4.
295 Anon. (1988) Manufacturing Printers, Magnetic Design, Train Scheduling, Hypermedia Systems and More, Intelligence: Int.J. on A.I. - Texas Instruments No.13, p7.
296 E. Payling (1988) Monitoring Organizational Performance using Expert Systems. Presented at the Fifth Young OR Conference, March 1988, Warwick.
297 D. Guerillot and E. Petiot (1986) Another Use of Inference Engines: Interactive User's Guide for Large Numerical Software (Aiglon). Presented at CIIAM 86: the Second Conference on Artificial Intelligence Marseille, Dec 1986.
298 R.G. Smith (1984) On the Development of Commercial Expert Systems, The AI Magazine 5, No.3, pp61-73.

299 M. Litt, J.C.H. Chung, D.C. Bond, G.G. Lininger and J.Hall (1988) A Scheduling and Planning Expert System for Multiple Furnaces, Eng.Appli. of AI 1, No.1, pp.16-21.
300 T. Lenau and L.Alting (1986) XPLAN - An Expert Process Planning System. Presented at Second Expert System Conference, London, October 1986.
301 J. Thorpe (1988) Using Expert Systems to Monitor an Automatic Stock Control System. Presented at the Fifth Young OR Conference, March 1988, Warwick.
302 W.F. Kaemmerer and J.R. Allard (1987) An Automated Reasoning Technique for Providing Moment-By-Moment Advice Concerning the Operation of a Process. Presented at AAAI-87: The Sixth National Conference on Artificial Intelligence, Seattle, July 1987.
303 P. Harmon and D. King (1985) Expert Systems: Artificial Intelligence in Business. John Wiley, New York.
304 J.M. Sepulveda and E.L. DePorter (1987) IPSS: An Approach Towards Automated Decisions in CIM Systems, Computers ind.Engng 13, Nos.1-4, pp.11-14.
305 T. Wittig (1988) Distributed Knowledge Based Systems for CIM. Presented at the Fourth CIM Europe Conference, May 1988, Madrid.
306 A.W. Smith and D.R. Hughes (1986) The Control of Manufacture by Information. Presented at Second Expert System Conference, London, October 1986.
307 D. Ben-Arieh and C.L. Moodie (1987) Knowledge Based Routing and Sequencing for Discrete Part Production, Journal of Manufacturing Systems 6, No.4, pp.287-297.
308 G. Bruno, A. Elia and P. Laface (1986) A Rule-Based System to Schedule Production, Computer 17, No.7. pp.32-39.
309 M.Z. Bell (1985) Why Expert Systems Fail, J.Opl.Res.Soc. 36, No.7, pp.613-619.
310 S.R.T. Kumara, S. Joshi, R.L. Kashyap, C.L. Moodie and T.C. Chang (1986) Expert Systems in Industrial Engineering, Int.J.Prod.Res. 24, No.5, pp.1107-1125.
311 D. Partridge and Y. Wilks (1987) Does AI have a Methodology which is Different from Software Engineering? Artificial Intelligence Review 1, pp111-120.
312 E.C.P. Portman (1987) The Alvey DHSS Large Demonstrator Project, ICL Technical Journal, May 1987, pp.371-375.
313 P.Rixhon (1986) Intelligent Knowledge-Based Systems: A Tool for Manufacturing Planning and Scheduling. In Artificial Intelligence in Economics and Management (L.F. PAU, Ed.), pp283-288, Elsivier Science (North Holland).
314 A. Dionisi-Vici, F. Malbocchia and L. Sisto (1986) A Methodology for Automating Expert Diagnosis in Expert Systems. Presented at CIIAM 86: the Second Conference on Artificial Intelligence, Marseilles, Dec 1986.
315 P. Hammond (1987) Logic-Based Tools for Building Expert and Knowledge-Based Systems: Successes and Failures of Transferring the Technology. In Knowledge-Based Expert Systems in Industry (J.KRIZ, Ed.), pp.69-90, Ellis Horwood, Chichester.
316 E. Pollitzer and J.Jenkins (1985) Expert Knowledge, Expert Systems and Commercial Interests, OMEGA 13, No.5, pp.407-418.
317 R. Stow, S. Lumm and P. Slatter (1986) How to Identify Business Applications of Expert Systems. Presented at the Second Expert System Conference October 1986, London.
318 B. Sheil (1983) Power Tools for Programmers, Datamation 29, Feb 1983, pp.131-144.

319 G.Peters and J.Naughton (1984) Complexity, Management and Change: Applying a Systems Approach - Comparing Systems Approaches. The Open University Press, Milton Keynes.
320 M.L.G.Shaw and B.R.Gaines (1987) An Interactive Knowledge-Elicitation Technique Using Personal Construct Technology. In Knowledge Acquisition for Expert Systems: a Practical Handbook (A.L.KIDD, Ed.), pp.109-136. Plenum, New York.
321 R.G.Moll (1986) A Method for Analysing Requirements for Possible Expert System Implementations. Presented at Second Expert System Conference, London, October 1986.
322 E. Gudes, A. Shafrir and S. Gerlitz (1986) Software and Knowledge Engineering Methods in Building Expert Systems. Presented at Second Expert System Conference, London, October 1986.
323 A.E.M. Oliver (1986) Control of Rapid Prototyping in Expert Systems Development. Presented at Second Expert System Conference, London, October 1986.
324 M. Thomas (1987) Structured Techniques for the Development of Expert Systems. In Research and Development in Expert Systems IV, (D.S. MORALEE, Ed.), pp.258-266, Cambridge University Press, Cambridge.
325 R. Budde, K. Kuhlenkamp, L. Mathiassen and H. Zullighoven (Ed.) (1984) Approaches to Prototyping. Springer Verlag, Berlin.
326 F. Prabst (1984) Experience with Prototyping in an IBM-Based Installation. In Approaches to Prototyping. (R. BUDDE, K. KUHLENKAMP, L. MATHIASSEN and H. ZULLIGHOVEN Ed.), pp.152-164, Springer Verlag, Berlin.
327 W.E. Riddle (1984) Advancing the State of the Art in Software System Prototyping. In Approaches to Prototyping. (R. BUDDE, K. KUHLENKAMP, L. MATHIASSEN and H. ZULLIGHOVEN Ed.), pp.19-26, Springer Verlag, Berlin.
328 L. Ford (1987) Artificial Intelligence and Software Engineering: A Tutorial Introduction to their Relationship, Artificial Intelligence Review 1, pp.255-273.
329 D.A. DeSalvo, A.E. Glamm and J. Liebowitz (1987) Structured Design of an Expert System Prototype at the National Archives. In Expert Systems for Business (B.G. SILVERMAN, Ed.), pp.40-77, Addison-Wesley, Reading MA.
330 M Stefik, J. Aikins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth and E. Sacerdoti (1982) The Organization of Expert Systems: A Tutorial, Artificial Intelligence 18, pp.135-173.
331 H. Reichgelt and F. van Harmelen (1985) Relevant Criteria for Choosing an Inference Engine in Expert Systems. In ES 85. Proceedings of th Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems (M.MERRY, Ed.), Cambridge University Press, Cambridge.
332 P.J.Kline and S.B. Dolins (1986) Problem Features that Influence the Design of Expert Systems. Presented at AAAI-86, the Fifth National Conference on Artificial Intelligent, Philadelphia, August 1986.
333 A.L. Kidd and W.P. Sharpe (1987) Goals for Expert Systems Research: An Analysis of Tasks and Domains. In Research and Development in Expert Systems IV, (D.S. MORALEE, Ed.), pp.146-152, Cambridge University Press, Cambridge.
334 D.S. Prerau (1985) Selection of an Appropriate Domain for an Expert System, The AI Magazine 6, No.2, pp.26-30.
335 W.J. Clancey (1985) Heuristic Classification, Artificial Intelligence 27, pp.289-350.

336 V. Vemuri (1978) Modelling of Complex Systems: An Introduction. Academic Press, London.
337 R.S. Pressman (1987) Software Engineering: A Practitioner's Approach, McGraw-Hill, New York.
338 A. Barton (1987) Experiences in Expert Systems. J.Opl.Res.Soc. 38, No.10, pp.965-974.
339 M.J. Ginzberg and E.A. Stohr (1982) Decision Support Systems: Issues and Perspectives. In Decision Support Systems: Proceedings of the NYU Symposium on DSS (M.J. GINZBERG, W. REITMAN and E.A. STOHR, Ed.), pp.9-31, North-Holland, Amsterdam.
340 E.G. Hurst, D.N. Ness, T.J. Gambino and T.H. Johnson (1983) Growing DSS: A Flexible Evolutionary Approach. In Building Decision Support Systems (J.L. BENNETT, Ed.), pp.111-132, Addison-Weseley, Reading MA.
341 P.G.W. Keen (1986) Decision Support Systems: The Next Decade. In Decision Support Systems: A Decade in Perspective (E.R. MCLEAN and H.G. SOL, Ed.), pp.221-237, Elsevier Science, North Holland.
342 R.H. Sprague (1989)A Framework for the Development of Decision Support Systems. In Decision Support Systems: Putting Theory into Practice (R.H. SPRAGUE and H.J. WATSON, Ed.), pp.9-35, Prentice-Hall, London.
343 M. Alavi and H.A. Napier (1989) An Experiment in Applying the Adaptive Design Approach to DSS Development. In Decision Support Systems: Putting Theory into Practice (R.H. SPRAGUE and H.J. WATSON, Ed.), pp.82-92, Prentice-Hall, London.
344 R.C. Henderson and R.S. Ingraham (1982) Prototyping for DSS: A Critical Appraisal. In Decision Support Systems: Proceedings of the NYU Symposium on DSS (M.J. GINZBERG, W. REITMAN and E.A. STOHR, Ed.), pp.79-96, North-Holland, Amsterdam.
345 P.W.G. Keen and T.J. Gambino (1983) Building a Decision Support System: The Mythical Man-Month Revisited. In Building Decision Support Systems (J.L. BENNETT, Ed.), pp.133-172, Addison-Weseley, Reading MA.
346 J. McDermott (1981) R1: The Formative Years. The AI Magazine 2, No.2, pp.21-29.
347 A. van de Brug, J. Bachant and J. McDermott (1986) The Taming of R1. IEEE Expert 1, No.3, pp.33-38.
348 W.P. Sharpe (1985) Logic Programming for the Law. In Research and Development in Expert Systems (M.A. BRAMER, Ed.), pp.217-228, Cambridge University Press, Cambridge.
349 D. Partridge (1986) Artificial Intelligence: Applications in the Future of Software Engineering, Ellis Horwood, CHichester.
350 J. Costello (1988) A Common Knowledge Base for Multiple Applications. Technical Report AISL35, Dept of Computer Science, Strathclyde University.
351 D.A.J. Swinkels, L.Lock-Lee and M. Saunders (1986) Comparison of Four Expert System Development Packages. Presented at Expert System Conference, London, October 1986.
352 Anon. (1986) Expertech Xi: The Expert System - Concepts and Facilities, Expertech Ltd, Slough.
353 S.P Ennis (1982) Expert Systems: A User's Perspective of Some Current Tools. Presented at AAAI-82: The National Conference on Artificial Intelligence, Pittsburgh, August 1982.
354 B.G. Buchanan (1979) Issues of Representation in Conveying the Scope and Limitations of Intelligent Assistant Programs. In Machine Intelligence 9 (J.E. HAYES, D. MICHIE and L.I. MIKULICH, Ed.), pp.407-425, Ellis Horwood, Chichester.

355 B.W. Boehm (1988) A Spiral Model of Software Development and Enhancement. Computer 21, No.5, pp.61-78.
356 Anon. (1988) Food, Petrochemical and Glass Producers Combine to Create Planning/Scheduling Package, Intelligence: Int.J. on A.I. - Texas Instruments No.13, p.5.
357 M. Dowson (1984) A Note on Microplanner. In Implementations of Prolog (J.A. CAMPBELL, Ed.), pp.19-27, Ellis Horwood, Chichester.
358 J.H. Alexander, M.J. Freiling, S.J. Shulman, J.L. Staley, S. Rehfuss and S.L. Messick (1986) Knowledge Level Engineering: Ontological Analysis. Presented at AAAI-86, the Fifth National Conference on Artificial Intelligence, Philadelphia, August 1986.
359 W. Swartout and R. Balzer (1982) On the Inevitable Intertwining of Specification and Implementation. Comms of the ACM 25, No.7, pp.438-440.
360 M. Berry (1988) I.S. Requirements: A Tale of Chickens and Eggs. Presented at the Systems Conference Lancaster, July 1988.
361 A.T. Wood-Harper, L. Antill and D.E. Avison (1985) Information System Definition: The Multiview Approach, Blackwell Scientific, Oxford.
362 D.T. Ross (1977) Reflections on Requirements. IEEE Trans. Software Engineering SE-3, No.1, pp.2-5.
363 D.T. Ross and K.E. Schoman (1977) Structured Analysis for Requirements Definition. IEEE Trans. Software Engineering SE-3, No.1, pp.6-15.
364 D.T. Ross (1977) Structured Analysis (SA): A Language for Communicating Ideas. IEEE Trans. Software Engineering SE-3, No.1, pp.16-34.
365 D.T. Ross (1985) Applications and Extensions of SADT. Computer 6, No.4, pp.25-34.
366 P.B. Checkland (1988) Personal Communication.
367 M.A. Rodger (1988) Production Planning and Control. Presented at the Systems Conference Lancaster, July 1988.
368 T.T. Carey and R.E.A. Mason (1983) Information System Prototyping: Techniques, Tools and Methodologies. INFOR 21, No.3, pp.177-191.
369 C. Floyd (1984) A Systematic Look at Prototyping. In Approaches to Prototyping (R. BUDDE, K. KUHLENKAMP, L. MATHIASSEN and H. ZULLIGHOVEN, Ed.), pp.1-17, Springer Verlag, Berlin.
370 S.T. Murdoch (1987) A Review of the Intersection of Expert Systems and Database Systems: Expert/Database Systems. Int.J. Systems Research and Info. Science 2, pp.111-119.
371 R. Worden (1986) Blackboard Systems. Computer Assisted Decision Making (G. MITRA, Ed.), pp.95-106, Elsiveir Science, North Holland.
372 C. Mellish and S. Hardy (1984) Integrating PROLOG in the POPLOG Environment. In Implementations of Prolog (J.A. CAMPBELL, Ed.), pp.147-162, Ellis Horwood, Chichester.
373 K. Lyytinen (1988) Stakeholders, Information System FAilures and Soft Systems Methodology. J. of Applied Systems Analysis 15, pp.61-81.
374 P.E. Hart (1981) Directions for AI in the Eighties. Sigart Newsletter 79, pp.11-16.

APPENDIX A
USEFULLY WORKING EXPERT SYSTEMS

Section 4.3 of this thesis presented a personal view of the track record of expert systems. The systems considered as usefully working in that discussion are listed here within the categories presented within section 4.3 above.

There are a total of one hundred and sixty systems listed; fifty-nine Fault Diagnosis systems, fifteen Process Control systems, nineteen `Electronic Rulebooks', forty-four Configurers, fourteen Schedulers, seven General Diagnosis systems and two systems otherwise unclassified. As described in section 4.3, there is no clear cut divide between these different categories.

The list provides the name of the system if any, the user or the vendor if the system is for sale, a description of the system and if known the language in which the system implemented.

The references refer to the the evidence that the system is working. It will be noted that for almost three quarters of the systems this evidence is found exclusively within two references, Feigenbaum et al (197) and Harmon (221). Further, the author has included systems such as Northrop's ESP, the actual status of which the author described in section 4.2.2 as unclear. The list then represents a personal, and optimistic, list of usefully working systems. Given the reasons discussed in section 4.2.2, many systems may be missing from the list while for systems in the list, it may be incorrect to include a number of them.

Expert Systems Categorized as Fault Diagnosis Systems

Sinter Advisor 1 Sulphide Corp, Australia. Diagnoses moisture problems in Lead-Zinc sintering. Written in Fortran. Rule induction was used during its development. (285)

British Steel. Diagnoses maintenance problems in the concast plant at Ravenscraig. Reported as working in a stand-alone consultative mode only. Written using a specially designed diagnostic shell called Sesame. (11)

AI-STOP-GO, Kawasaki Steel. Diagnoses problems in a blast furnace. Taking the output from a conventional computer system which identifies imminent problems, the system identifies the cause and action to correct it. (266)

Polapres, MMRA, Luxembourg. Diagnoses causes of problems in steel rolling process. Written in Personal Consultant Plus. (282)

SHEARER, British Coal. Diagnoses problems with coal cutting equipment. Written in Savoir. (281)

CATS-1, General Electric. Diesel-Electric Locomotive `trouble-shooter'. (Also known as DELTA and GEN-X.) Developed in Lisp. Re-written in Forth. (236,240,287)

Turbomac, Hartford. Diagnoses the causes of turbomachinery vibration. Written in Rulemaster. (221)

Rotary Equipment Vibration Advisor, sold by Stone and Webster Engineering. Aids the interpretation of vibration measurement in rotary equipment. Written in Exsys. (221)

Vibration Diagnosis Expert System, GM. Aids technicians diagnose car vibration and noise problems. Written in S.1. (221)

Ford. Car engine diagnosis system. (197)

Renault. Diagnosis of automatic gearbox problems. (197)

Charley, GM. Car engine diagnosis system. (197)

Atrex, Toyota. Assist experts in diagnosing car motor electronics faults. (197)

DOC, Prime Computers Inc. Electronic field service engineer assistant, diagnosing the causes of common computer system software problems. Written in Prolog (240)

Expert Probe, Unisys. Printed circuit board quality control fault diagnosis. Written in KEE. (221)

Hewlett-Packard. Diagnoses disc drive faults. Written in HP-RL. (221)

DEFT, IBM. Diagnosing quality control test defects in computer disc drives. Written in ESE. (197)

RBEST, Hewlett-Packard. Analyses results of disk memory tests. Written in Lisp. (221)

AI-SPEAR, Digital. Diagnoses faults in tape drives through analysis of logged error messages. (221,227)

Permaid, Honeywell. Assists in diagnosis and provides preventive maintenance plans for large disc drives. Written in LOOPS. (221)

Page-1, Honeywell-Bull. Troubleshoots high-speed page printing system. Written in Lisp and LOOPS. (221)

DASD, Bool & Babbage. Analyses data to identify and diagnose problems with data storage devices. Written in Aion. (221)

Diag 8100, Travelers Corp. Diagnoses problems with IBM 8100 computers. Written in M.1. (221)

IDT, Digital. Diagnoses faults found during production at their Burlington factory. (227)

TIMM-Tuner, Sold by General Research Corp. Assists in tuning VAX/VMS operating systems. (221)

Mindover, Applied Data Research Inc. Analyses a computer system's log to facilitate performance management & capacity planning. (221)

VPA, Digital. Analyses a computer system's log and recommends improvements. (221)

Expert-Tek, Motorola. Aids diagnosis and repair of computer systems and helps plan customer service needs. Written in S.1. (221)

Sold by ICL. Assists in the capacity management of computer systems. (197)

Seguide, Fujitsu. Analyses output from computer load prediction package and offers solution. Written in EShell. (221)

ESPm, sold by NCR. Analyses computer maintenance logs to identify possible future problems. Written in S.1. (221)

Mask, Norcom. Assists helpline staff diagnose callers' problems with Norcom software package. Written in 1st Class. (221)

BDS, Lockheed. Signal switching system fault diagnosis. Written in LES. (221)

NTC, Digital. Assists in diagnosing problems with crashed computer networks. (227)

GEMS-TTA, AT&T. Diagnoses faults on phone trunk lines. Written in OPS83. (221)

IDEA, sold by Pacific Bell. Assists diagnosis of problems with particular telephone switch gear. Written in Exsys. (221,236)

Herald Terminal Fault Diagnosis, British Telecom. Assists in the repair of Herald terminals. Written in Tracker, a specially developed tool. (241)

COMPASS, GTE. Analyses "often voluminous" reports containing phone switch gear operating data to recommend switch gear maintenance plans. Written in KEE. (221,236)

ACE, AT&T. Analyses daily operating and repair data of telephone system, identifying areas of preventive maintenance and further repair for telephone lines. Developed in OPS4. (221,236,240)

Iberduero, Spain. Identifies disturbances in electricity supply, determines the cause and provides solutions. Written in Lisp. (284)

TurbinAID, Westinghouse. On-line monitoring of steam turbines. Operating data is transmitted to a central Westinghouse control office where the diagnosis is made by the system. It is then transmitted back to the turbine site as well as to the Westinghouse central experts who also monitor the on-line turbine data. System also known as PDS (197,221,286)

GenAID, Westinghouse. On-line monitoring of electricity generators. (221)

ChemAID, Westinghouse. On-line monitoring of boiler and steam systems. (221)

Du Pont. Provides solutions to slurry blockage problems in pipelines. (197)

Du Pont. Determine locations of water inputs in large and complex chemical process. (197)

Campbells Soup. Diagnoses unusual operating problems in different `cookers' used to sterilize soup prior to canning. The cookers vary by make, model, age, etc, and are located world wide. Previously, one expert would deal with the problems, by phone or sometimes a site visit. The system handles 95% of the expert's work. It is written in Personal Consultant. (236)

GOI, Fuji Electric Co. Diagnoses transformer oil abnormalities using data from an automatic gas analyser. Written in COMDEX, a Fuji developed shell. (221)

TOGA, Hartford. Analyses insulation oil to diagnose faults in large utility transformers. Written in Rulemaster. (221)

Texas Instruments. Diagnoses faults in optical coating machines used in microchip manufacture. (197)

Photolithography Advisor, Hewlett-Packard. Diagnoses errors in photolithography process used in integrated circuit manufacture. Written in Prolog. (221)

TQMSTUNE, Lawrence Livermore Labs. Assists in tuning triple quadrupole mass spectrometers. Written in KEE. (221)

Advanced Coater Plant Advisor, ICI. Assists in diagnosing and rectifying faults in a plastic sheet treatment plant. Written in Savoir. (241)

IMP Texas Instruments. Diagnoses problems with epitaxial reactors used in microchip manufacture. Written in Personal Consultant. (197,283)

Reis Service Expert, Reis Machines Inc. Helps identify and solve problems with Robostar factory robots. Written in KDS. (221)

ASEA Maintenance Assistant, Ford. Assists in diagnosis of robot problems. Used mainly for training. Written in Personal Consultant. (221)

Hotline Helper Texas Instruments. Assists customer service to diagnose problems with Omni printers. Written in Personal Consultant. (221)

Mentor, Honeywell. Aids field technicians diagnose problems in air conditioning units. Written in Lisp. (221)

Hoist Diagnoser, Oxko. Isolates faults in hoisting equipment used in a plating process, recommending repairs. Written in Insight. (221)

Sold by Novacast. Casting defect analyser. (197)

MUDMAN, Sold by N.L. Baroid. Identifies problems with oil well drilling mud. Written in OPS5. (221,286)

Expert Systems Categorized as Process Control Systems

BAISYS, NKK. Monitors a blast furnace, identifying furnace abnormalities. Written in a version of EShell adapted for real-time processing. (197,221,292)

Nippon Steel. Blast furnace monitoring system. (197)

Kawasaki Steel. Used within the control process of a sinter plant. Stated as required to "optimize" the control process, it uses Fuzzy Logic. (266)

Nippon Steel. Monitoring and identification of problems in a continuous smelting process. (197)

Kawasaki Steel. The system is part of the computer control system of a slab reheating furnace in a hot strip mill. The system minimizes fuel usage. The system is relayed any changed process variables. It firstly identifies if the change is significant. If it is, the system accesses a library of linear programmes to calculate the required corrections which are sent back to the conventional process control system. (266)

Kawasaki Steel. Automated monitoring and control system of a billet mill finishing line. The system replaced Fortran programmes which took a great deal to update. Written in C. (266,289)

LINKman, Blue Circle. Cement production control. The system uses fuzzy logic and is said to have reduced energy costs. Written in C. (221,356)

Phosphorous Burden Advisor, FMC. Monitors and advises actions in a phosphorous refining process. FMC developed a shell which was used by the system. (197)

Turbine Generator Controller, Tohoku Oil. Monitors the requirement for power and steam needed by an oil refinery, and defines optimum turbine operating levels. Variables considered include operations plan, the weather and the cost of electricity. Written using a tool developed by Fuji Electronic. (221)

Component Impact Analysis System, Argonne National Laboratory. Advises nuclear reactors operators on the proper valve and switch settings. Written in Prolog. (221)

Corning Glass. The system assists in the control of a glass annealing oven or `Lehr'. The system comprises an expert `simulation' and an expert planner. The `simulation' provides a temperature curve through the oven given the control settings. The expert planner provides the control settings to obtain a particular temperature curve. Written in Lisp. (191)

RBT, J.H. Jansen Co. Simulates boilers to aid operator training. Written in Fortran and C. (221)

Du Pont. Provides advice concerning impurity purging for the operators of a distillation column. (197)

Sold by Hitachi. Control of train brake operation to provide a smooth ride. (197)

Mover, Digital. Co-ordinates amd drives two robot warehouse transporters. (221)

Expert Systems Categorized as `Electronic Rulebook' Systems

CASES, IBM. Guides an IBM employee through the company's capital asset transfer procedures. Written in ESE. (197)

Capital Expert System, Texas Instruments. Analyses capital investment proposals within TI. Written in Personal Consultant Plus. (221,283,286)

Underwriter's Aid, Nippon Life. Assess insurance risk to the company of new policies involving diabetes. (Fourteen further diseases will be added in the future.) The expertise was based on a manual of insurance-related medical knowledge. Written in KEE. (197,221)

Authorizer's Assistant, American Express. Calculates a credit limit and so advises on acceptance of a debt. Written in ART. (197,286)

Loan Approval Advisor, Kredietbank, Belgium. A existing system, the Numeric Credit Judgement system, obtains information about a loan request and outputs a scoring list. If the loan cannot be allowed given this list, the expert system is used. The system asks further questions and outputs advise to staff. Written in SAGE. (241)

BEST MIX, Sanwa Bank. Advises on large personal investment portfolios. Considers tax law, maturity dates as well as interest rates. Written in Brains. (197,221)

ExperTAX, Coopers & Lybrand. Evaluates the affect of US tax law on companies. Replaced a large written questionnaire which was analysed by an expert. Written in Lisp. (221,286)

Sold by AD/PR Software. Gives advice concerning Californian tax law with regard to financial transactions with advertising agencies. Written in 1st Class. (221)

Sold by Helix Expert Systems Ltd. Helps companies comply with the UK Data Protection Act. Written with Expert Edge. (221)

Sold by Expertech. System helps determine if an employee is covered by two UK employment acts. Written in Xi Plus. (221)

National Health Service. UK pension rights advisor used by NHS personnel to answer enquiries. (197)

PTE, sold by Computer Law Systems. Analyses employee benefit transactions governed by Employment Retirement Income Security Act. Written in Personal Consultant Plus. (221)

Sold by Buyers Casgrain. Provides legal advice concerning trading between USA and Canada. Written in Guru. (221)

Sold by Novacast. Postal rule analyser identifying the cheapest route and required documentation. (197)

Technical Specifications Advisor, sold by Stone & Webster, The system provides safety and regulatory advice leading from the proposed maintenance and operation of manufacturing plants, power plants and chemical plants. Written in Exsys. (221)

Fire Planning Equipment System, Shell. Determine the fire fighting equipment required by petroleum tank storage depots. Written in Crystal. (241)

Sold by Stone & Webster Engineering. Identifies welding qualifications of personnel required for major US construction projects. Written in Exsys. (221)

Hazardous Chemical Advisor, Air Products & Chemicals. Provides advice on how to handle, label and ship hazardous chemicals. Written in 1st Class. (221)

Gas Turbine Test Selector, Rolls-Royce. Advises choice of tests required for new turbine designs. The function was previously guided by a manual. Written in Xi Plus. (241)

Expert Systems Categorized as Configurers Systems

CSF, IBM. Assists in the estimation of hardware relocation costs. Written in ESE. (221)

Intelligent Software Configurer, Honeywell-Bull. Helps configures software for data processing customers. Written in Loops. (221)

XSEL, Digital. Assists salesmen in configuring their computer requirements. Written in OPS5. (197,227)

Dragon, System Designers. Assists salesmen to configure ICL 39 computers. Written in Envisage. (221)

BEACON, Unisys. A sales aid used to ensure correct and complete configuration of a computer systems. It provides input to an inventory system. Written with KNET, a Unisys software tool. (221)

CONAD, Nixdorf. Used by salesmen to configure Nixdorf's 8864 computer system as well as the configuration of technical and financial supporting services. Written in Twaice. (221)

SYCSON, Honeywell. Configures the detailed design of DPS 90 mainframes. Written in OPS5. (221)

Ocean, NCR. Part of the customer ordering system, checking the engineering implications of NCR Computer installations. (221)

SNAP, Infomart. Assesses personal computer requirements. Written in Personal Consultant. (221)

CDES/S, Fujitsu. Aids inexperienced engineers to configure the detailed design of S series computers. Written in EShell. (221)

XCON, Digital. Configures the detailed designs of a number of Digital computer types. Written in OPS5. (227,270,271,346)

Hitachi. The design of computer room layout from the equipment housed and the dimensions of the room. Written in Lisp. (221)

DBA Assistant, sold by Knowledge-Based Engineering Inc. Assists in the design and evaluation of problems with database structures within an IDMS/R environment. Written in KES II. (221)

Cabling Configurer, Ferranti. Determines efficient configurations for the cabling connecting circuit boards together. Written in ART. (221)

Hail-I, Hazeltine Group. Analyses printed circuit board designs with regard to component insertion. Written in OPS5. (240)

WES. Identifies wirability of computer chip designs. Written in KEE. (221)

StrateGene, Intelligenetics. Using molecule and cloning information, the system guides a researcher through complex cloning experiments. Written in KEE. (221)

Tokyo Electric Power. Supports the design of large-capacity electric substations. Cost, safety, maintenance and construction constraints are considered. (197,221)

Brush Designer, GM. Assists in the design of electric motor brushes and springs used in the motor industry. Written in S.1. (216,221)

Schlumberger. Aids in the design of specialist transformers used in bore hole sensors. Written in STROBE. (197,221)

British Petroleum. Aids the design of gas/oil separators. (197)

Optex, Canon. Aids the design of telephoto lenses. The system configures a design from user inputs then generates the input for a conventional system which evaluates the design. Written in Lisp. (197,221)

UFEL, British Coal. Diagnosis mine methane risk in projected mine shafts. Written in SAGE. (241,281)

CORPS, British Gas. Predicts corrosion for gas well tubes. Written in SAGE. (241)

Dustpro, US Bureau of Mines. Provides advice concerning dust control and ventilation techniques in mines. Written in Insight 2 Plus. (221)

Methpro, US Bureau of Mines. Advises on the control of methane gas in coal mines. Written in Insight 2 plus. (221)

ACEKIT, Mitsubishi Electric Co. Costs large electric motors. Written in Prolog. (221)

Kajima. Assesses the probability of liquification at different depths on a building site based on bore hole data and employing a number of different approaches which can give different results. (197)

Predicte, Lend Lease Corp. Estimates the length of construction time of a high rise building from basic design information. (197,221)

BEST, Sekisui Kagaku. Aids in the decomposition of design and the selection of parts in prefab house designs. Written in Prolog. (221)

LOKE. Selects appropriate oil field drill bits. Written in M.1 and C. (221)

MESA-1, Ishikawajima. Selection of an appropriate merchant ship engine based on a number of criteria. Written in EShell. (221)

Boeing. Selection of the correct tools and materials for assembling electrical connectors. Written in OPS5. (221)

Weld Selector, sold by American Welding Institute. Welding electrode selection. Written in Personal Consultant Plus. (221)

Navistar. Configures specialist trucks within manufacturing limitations. Written in KEE. (197)

AALPS, US Army. Configures air cargo shipping loads. Written in Prolog. (221)

Sold by Purdue University. Assists in determining selling or storing strategies for grain farmers based on futures market prices. Written in Personal Consultant Plus. (221)

Sold by Palladian software. Assists in evaluating complex business proposals. Written in Lisp. (236)

ONCOCIN, Stanford University. Assists in managing clinical data of patients undergoing experimental protocols. Within this Oncocin provides advise concerning drug toxicity and the ordering of certain tests. Written in Lisp. (221)

Sold by Expert Tech. Provides layouts for yellow pages. Written with tool based on cartesian geometry. (221)

Sold by Rockwell. Configures the layout of printing plates given a newspaper format. (221)

Sold by Crossfield CSI. Assists in laying out newspaper pages and the planning of printing runs. Written in ART. (221)

ESP, Northrop. Routing of aircraft component manufacture. Written in KEE. (197)

CDS, Digital. Sourcing of computer components. Unlike the superseded conventional system, CDS requires no manual intervention. Written in OPS5. (221)

Expert Systems Categorized as Scheduling Systems

FACOM-M780, Kawasaki Steel. Assists in scheduling the berthing of ships and dock facilities in a steel works plate mill. As well as assigning ships and cranes to particular orders, the scheduler also has to cope with late arrival of ships, of plates for shipment and possible late delivery to the customer. The system provides schedules of ship berthing, cargo loading sequences and crane allocations. The expert operates the system, ammending the knowledge-base to suit the situation. (266)

NKK. Schedules `rounds' in a hot strip mill. The orderbook files are pre-analysed by a conventional system and the result input into the expert system. The expert system attempts to minimize the slab yard inventory while providing realistic `rounds'. The scheduler amends the inputs into the expert system to allow for wider considerations. (294)

NKK. Schedules the flow of material and products, assigning workers to tasks.(197)

Texas Instruments. Schedules printer production at TI's Temple works. Developed using an early version of an shell specially developed by TI for scheduling tasks and known as MASS. (283,295)

IBM Japan. Assigning workers to tasks on computer printer assembly line considering parts availability, orders and assembly line organization. (221)

LMS, IBM. Rescheduling advisor within a microprocessor factory. The system obtains Work-In-Progress data from a conventional computer system. From this it advises operators and managers on the priorities in queues ahead of stations and the rerouting required if a problem developes. Written using XEN, a tool especially developed for this application. (197,221)

Dispatcher, Digital. Routes and releases works orders. The system models the state of the factory to determine the best release timing and route. (221)

COCOMO1, sold by Level Five Research. Used to aid the scheduling of software projects. The system predicts cost and labour requirements in software projects forecasting the productivity of the project. Written in Insight 2 and Pascal. (221,236)

CASS/X, Fujitsu. Schedules the batch processing of a computer system within a Fujitsu site although the system is over-ruled for rush jobs. Written in EShell. (221)

CA/ISS 3, sold by Computer Associates Inc. Assists in managing computing resources and identifying cost-effective upgrades to the system. System operating data is pre-processed then analysed by CA/ISS 3. Written in C. (221)

SIMLAB, Boeing. Schedules a flight simulator. Written in OPS5. (221)

FRESH, US Navy. Advises on the deployment of ships in the US Pacific fleet. (197)

ESTRAC-II, Japan. Traffic control and scheduling of railway operations. Written in Lisp. (222)

Southland. Involved in scheduling a fleet of supermarket delivery lorries. It amends a schedule produced by a conventional system to take account of details such as frozen products and empty boxes which otherwise required manual alterations. (197)

Expert Systems Categorized as General Diagnosis Systems

Manager's Broker Monitoring System, Bear Sterns & Co. The system presents a profile of the brokers' discretionary activity and is able to identify items of interest to a broker's manager. Developed in Goldworks. Implemented in Lisp and C. (221)

Performance Indicator, DHSS. The system identifies Health Authority performances and reasons for extremes. This was previously a laborious manual system. Written in Crystal. (241,296)

P4765A Electrocardiograph, Hewlett-Packard. Aids in diagnosing heart disease by interpreting ECG readings. Written using an ECL shell. (221)

Pulmonary Consultant, sold by Medical Graphics Corp. Analyses pulmonary-function data to assist in diagnosing lung disease. Written in Pascal. (221)

IPE-CAC, Maryland Poison Centre. Diagnoses poisoning, in particular, human antihistamines and decongestant ingestion. Written in Personal Consultant with some Lisp. (221)

Wheat Councillor, ICI. Diagnoses requirements for disease-control in wheat and suggests chemical control treatment. Written in Savoir. (197,221,236)

Metals Analyst, General Electric. Allows a novice to identify types of metal. Written in Exsys. (221)

Uncategorized Usefully Working Expert Systems

Dipmeter Advisor, Schlumberger. Determining geology from drilling core samples. The exact function carried out by this system is not given. The system did however have a protracted development path. Written in Lisp. (197)

AIGLON. System acts as a `front-end' to a finite element analysis package. Written in S.1. (297)

APPENDIX B

THE LISTING OF THE "RESITE" MODULE CODING

The listing below is the PROLOG code developed to carry out the RESITE function within the UAAMS system.

```
***** CALL CLAUSE *****
((TEST RUN)
  (RESITE (HR 935)))

***** PROCEDURE FOR RE_SITING ONE PROCESS.WEEK *****
((RESITE (_process|_week))
  (RES_GET (_process|_week) _ftaload)
  (RES_SET MINMAX)
  (OR ((EQ 0 _ftaload)) ((RES_LOAD _ftaload 0 0)))
  (RES_PUT (_process|_week))
  / )

**** SUB_ROUTINE: GET DATA FROM SCHED_DATA ****
((RES_GET (_process|_week) _ftaload)
  (SCHED_DATA (_process|_week) (_ftaload|_X) _sitedata)
  (RES_GET1 _sitedata))
((RES_GET1 (_headsite|_tailsite))
  (OR ((EQ _tailsite ())) ((RES_GET1 _tailsite)))
  (EQ _headsite (_site _plan _cum-plan _insistload _ftamax _ftaload))
  (ADDCL ((RES_DATA UNLOADED (_site _plan _insistload _ftamax 0)))))

**** SUB_ROUTINE: CALCULATE MINIMUM & MAXIMUM PROPORTIONS ****
((RES_SET MINMAX)
  (FORALL ((RES_DATA UNLOADED (_site _plan _insist _ftamax 0))
           (NOT EQ _ftamax 0))
          ((SUM _insist _ftamax _maxload)
           (TIMES _plan _propinsist _insist)
           (TIMES _plan _propmaxload _maxload)
           (ADDCL ((RES_PROP START _site _plan _propinsist)))
           (ADDCL ((RES_PROP END _site _plan _propmaxload))))))

**** SUB_ROUTINE: LOAD FTA_ORDERS ONTO SITES ****
((RES_LOAD _ftaload _startprop _affectedplans)
  (ISALL _proploadlist _propload (RES_PROP _flag _site _plan
                                                              _propload))
  (LEAST _proploadlist _minprop)
  (RES_PROP _flag _site _plan _minprop)
  (SUM _startprop _propload _minprop)
  (TIMES _propload _affectedplans _loadsize)
  (OR ((LESS _ftaload _loadsize) (EQ END _loop)
       (TIMES _actualprop _affectedplans _ftaload))
     ((EQ _propload _actualprop) (EQ YES _loop)))
  (FORALL ((RES_DATA LOADING (_s _p _i _fmax _fload)))
          ((TIMES _actualprop _p _extraload)
           (SUM _fload _extraload _newfload)
           (DELCL ((RES_DATA LOADING (_s _p _i _fmax _fload))))
           (ADDCL ((RES_DATA TEMP (_s _p _i _fmax _newfload))))))
  (FORALL ((RES_DATA TEMP (_s _p _i _fmax _fload)))
          ((DELCL ((RES_DATA TEMP (_s _p _i _fmax _fload))))
           (ADDCL ((RES_DATA LOADING (_s _p _i _fmax _fload))))))
  (SUM _newftaload _loadsize _ftaload)
```

```
  (RES_LOAD1 _flag _site _affectedchange)
  (SUM _affectedplans _affectedchange _newaffectedplans)
  (DELCL ((RES_PROP _flag _site _plan _minprop)))
  (OR ((EQ _loop END))
      ((RES_LOAD _newftaload _minprop _newaffectedplans))))
((RES_LOAD1 START _site _plan)
  (DELCL ((RES_DATA UNLOADED (_site _plan _insist _ftamax _ftaload))))
  (ADDCL ((RES_DATA LOADING (_site _plan _insist _ftamax 0)))))
((RES_LOAD1 END _site _affectedchange)
  (DELCL ((RES_DATA LOADING (_site _plan _insist _ftamax _ftaload))))
  (ADDCL ((RES_DATA LOADED (_site _plan _insist _ftamax _ftaload))))
  (SUM _plan _affectedchange 0))

**** SUB_ROUTINE: SAVE FTA_LOADS IN SCHED_DATA ****
((RES_PUT (_process|_week))
  (SCHED_DATA (_process|_week) _totals _sitedata)
  (RES_PUT1 _sitedata _free _overload _newsitedata)
  (EQ _totals (_ftatotal _something _oldoverload _oldfree))
  (EQ _newtotals (_ftatotal _something _overload _free))
  (DELCL ((SCHED_DATA (_process|_week) _totals _sidedata)))
  (ADDCL ((SCHED_DATA (_process|_week) _newtotals _newsitedata)))
  (KILL RES_DATA)
  (KILL RES_PROP))
((RES_PUT1 () 0 0 ()))
((RES_PUT1 (_firstsite|_tail) _free _overload (_newsite|_newtail))
  (RES_PUT1 _tail _tailfree _tailoverload _newtail)
  (EQ _firstsite (_site _plan _cumplan _insist _ftamax _fta))
  (RES_DATA _state (_site _plan _insist _ftamax _ftaload))
  (EQ _newsite (_site _plan _cumplan _insist _ftamax _ftaload))
  (SUM _insist _ftaload _load)
  (SUM _load _inbalance _plan)
  (OR ((LESS _inbalance 0) (EQ _tailfree _free)
       (SUM _overload _inbalance _tailoverload))
      ((EQ _tailoverload _overload) (SUM _tailfree _inbalance
                                              _free))))

***** UTILITY CLAUSES *****
((TOTAL () 0))
((TOTAL (_head|_tail) _total)
  (TOTAL _tail _totaltail)
  (SUM _head _totaltail _total))

((LEAST _value _value)
  (VAR _value))
((LEAST (_value|_tail) _least)
  (LEAST _tail _least)
  (OR ((LESS _least _value)) ((EQ _least _value))))
((LEAST (_value|_tail) _value))

((MOST _value _value)
  (VAR _value))
((MOST (_value|_tail) _most)
  (MOST _tail _most)
  (OR ((LESS _value _most)) ((EQ _value _most))))
((MOST (_value|_tail) _value))
```

***** TEST DATA *****

```
((SCHED_DATA (HR 931) (11795 10250 22759.998 820.0001)
            ((LLANWERN 6650 6650 11795 11795 8614.9998)
             (LACKENBY 3600 3600 1600 1180 1179.9999)
             (PORT_TALBOT 2000 2000 10800 300 0)
             (RAVENSCRAIG 6650 6650 4850 2000 1999.9994))))
((SCHED_DATA (HR 932) (7155 9050 1810 0)
            ((LLANWERN 6050 12700 1575 7155 5685)
             (LACKENBY 3000 6600 2130 7155 1470))))
((SCHED_DATA (HR 933) (0 11850 0 10150)
            ((LLANWERN 7750 20450 1700 0 0)
             (LACKENBY 4100 10700 0 0 0))))
((SCHED_DATA (HR 934) (3475 11380 12840 0)
            ((LLANWERN 7890 28340 390 3475 2546.2169)
             (LACKENBY 3490 14190 370 3475 928.7827))))
((SCHED_DATA (HR 935) (0 11550 0 11550)
            ((LLANWERN 8100 36440 0 0 0)
             (LACKENBY 3450 17640 0 0 0))))
```

APPENDIX C

MODELS USED DURING THE ANALYSIS CARRIED OUT TO STRUCTURE THE MICRO-X10 SYSTEM

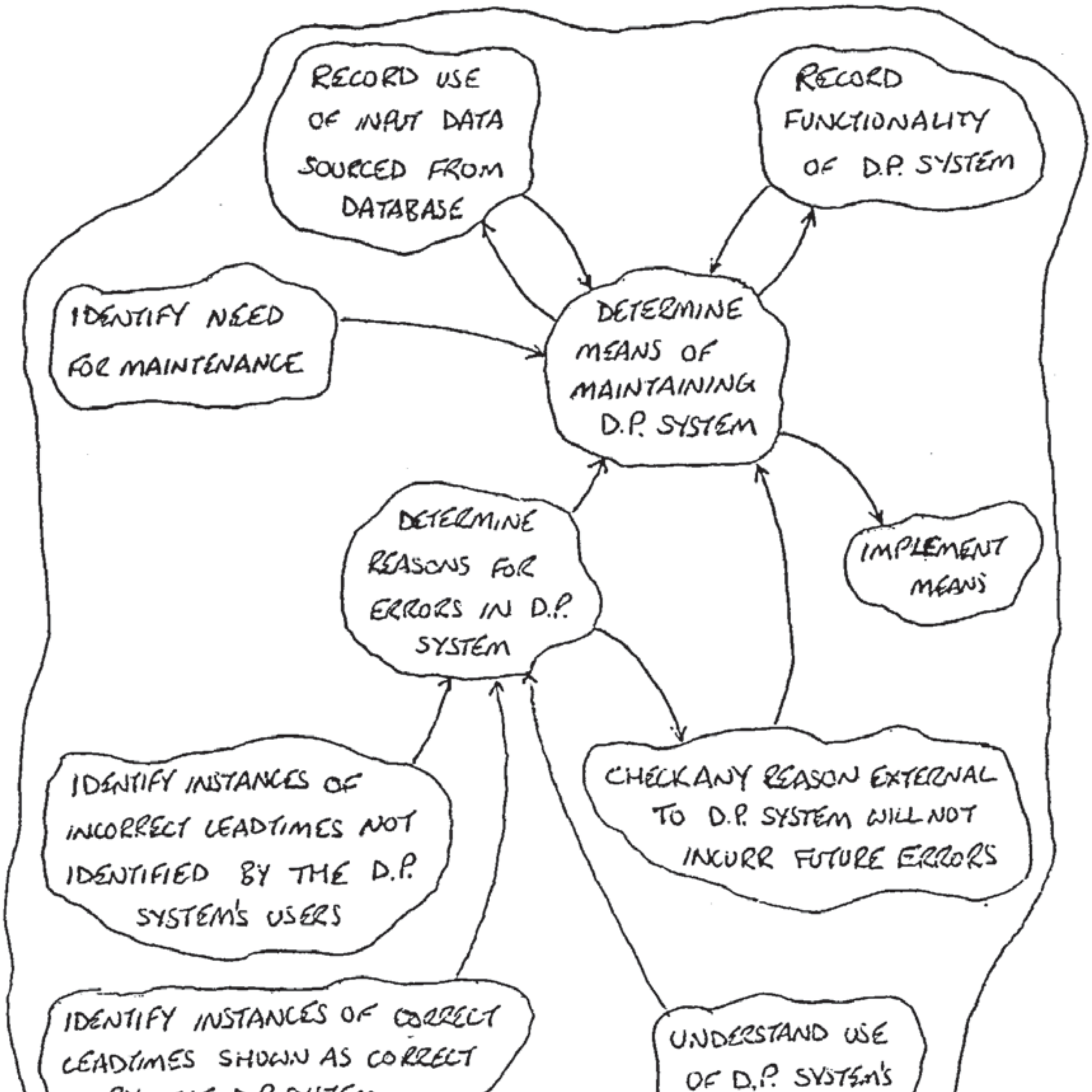
The following six models were used during the initial structuring of the MICRO-X10 system.

The first model, a conceptual model, was based on a direct statement of the problem but was found to be too broad to be greatly useful. The second model was devéloped to overcome this problem. No formal comparison table was developed using the conceptual models. The outcome of the informal comparison which was carried out is described in the second sub-section of section 7.3.1. This work allowed the development of MICRO-X10 to be justified an also provided an initial (although simple) structure for the system (figure 7.1).

The final four models were used to analyse two of the activities in the initial structure of MICRO-X10 as described in the third sub-section of section 7.3.1.

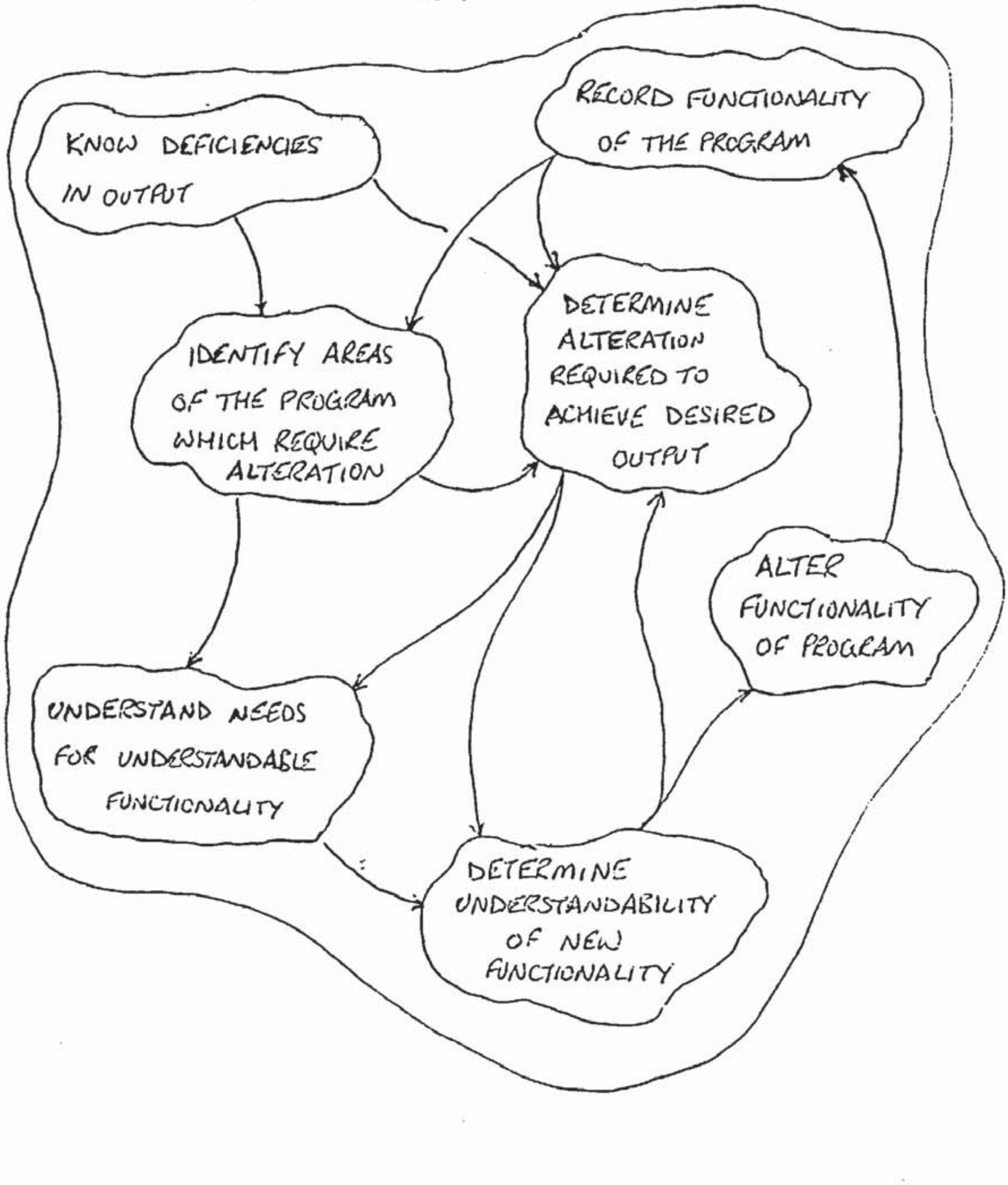
MODEL ONE

A SYSTEM TO MAINTAIN A COMPLEX DATA PROCESSING SYSTEM WHICH CONVERTS INPUT DATA SOURCED IN A DATABASE INTO AN OUTPUT USED TO IDENTIFY INCORRECT LEADTIMES



MODEL TWO

A SYSTEM WHICH ALTERS THE FUNCTIONALITY OF A PROGRAM GIVEN A KNOWN DEFICIENCY IN THE OUTPUT OF THE PROGRAM BUT MINDFUL OF THE NEED FOR FUTURE ALTERATIONS WHEN RECORDING THE FUNCTIONALITY OF THE PROGRAM

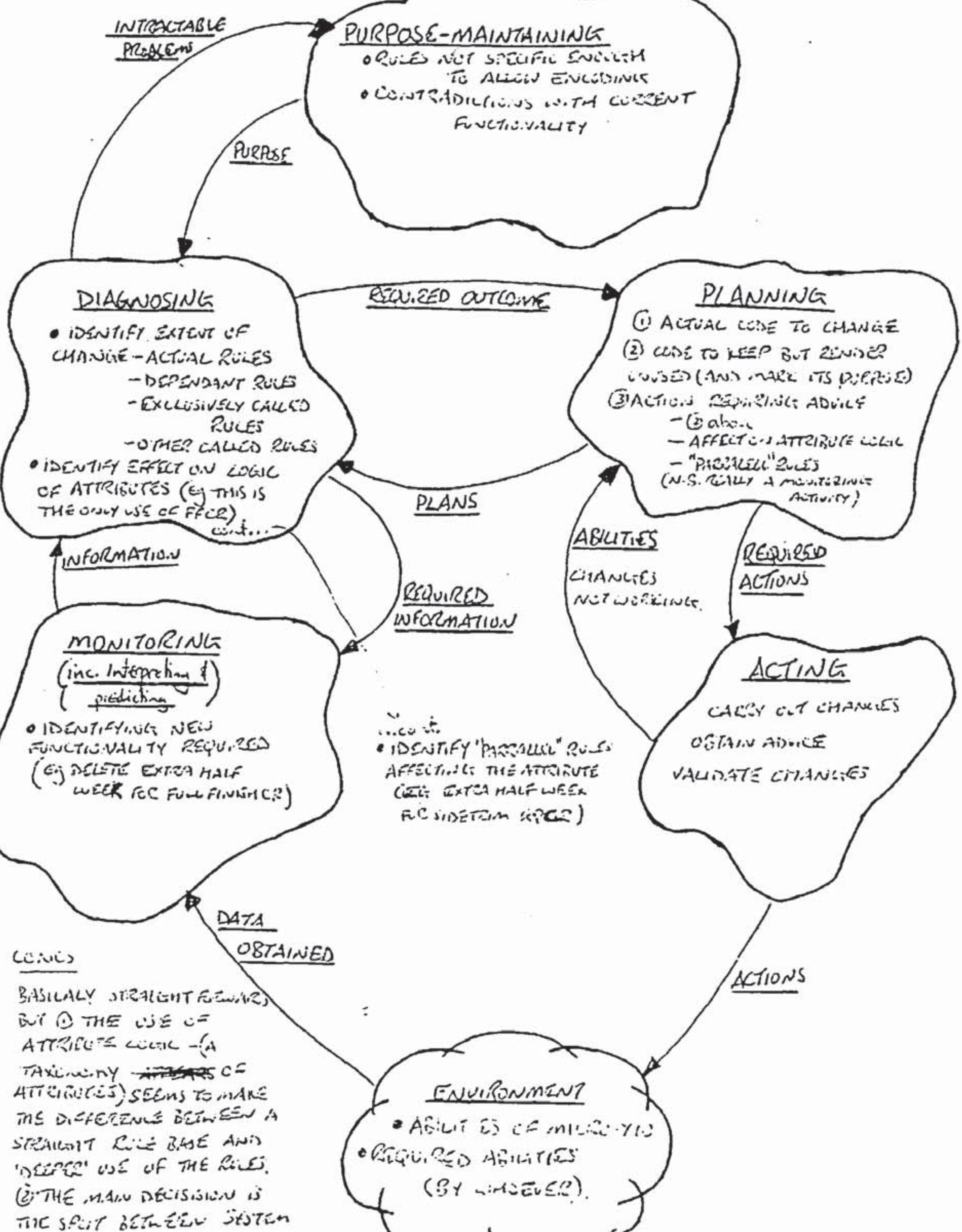


MODEL THREE

UNDERSTANDING OF MICRO-XIB FUNCTIONALITY [illegible]

⇒ ACTING = ACTION OF ANALYSIS WITH [illegible]
BEING THE OUTPUT (?)

⇒ MODEL = CHANGING THE FUNCTIONALITY OF ~~XIB~~ MICRO-XIB



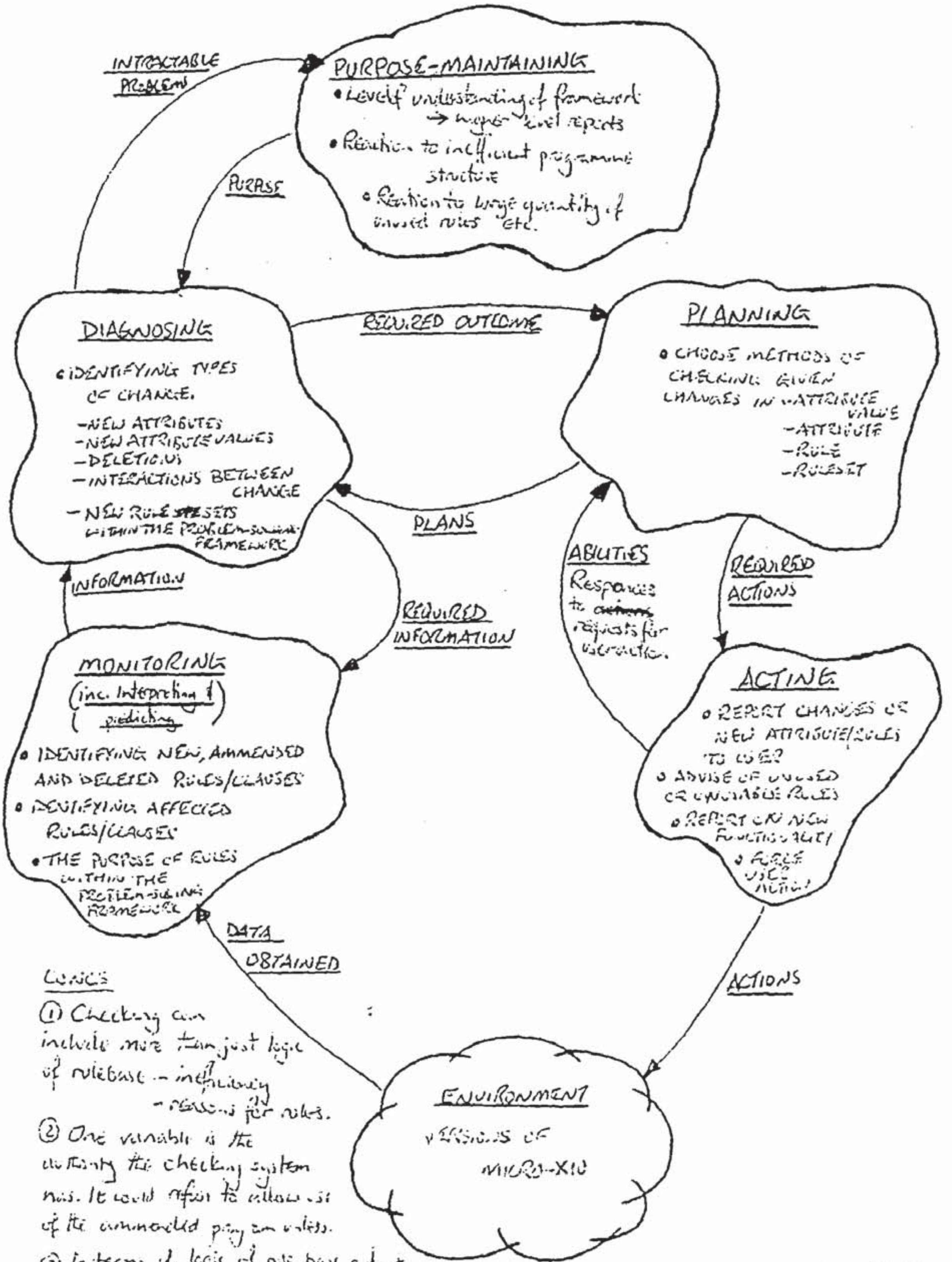
CONCS

BASICALY STRAIGHTFORWARD BUT ① THE USE OF ATTRIBUTE LOGIC - (A TAXONOMY ~~[illegible]~~ OF ATTRIBUTES) SEEMS TO MAKE THE DIFFERENCE BETWEEN A STRAIGHT RULE BASE AND 'DEEPER' USE OF THE RULES.

② THE MAIN DECISION IS THE SPLIT BETWEEN SYSTEM AND USER (AS PER PLANNING).

③ THE UNDERLYING COMMENT IS THE ASSUMPTION THAT A RULEBASE IS NOT UNDERSTANDABLE ON ITS OWN. THE 'PURPOSE-MAINTAINING' COULD BE SEEN AS THE AUTHORITY CHECKING THAT THE RULEBASE IS UNDERSTANDABLE AND THAT UPDATING IT IS NOT TOO MUCH OF A TASK GIVEN RULES/HELP/DESCRIPTION IN THE RULEBASE.

MODEL FOUR



LINKS

① Checking can include more than just logic of rulebase – inefficiency – reasons for rules.

② One variable is the authority the checking system has. It could refuse to allow use of the ammended program unless.

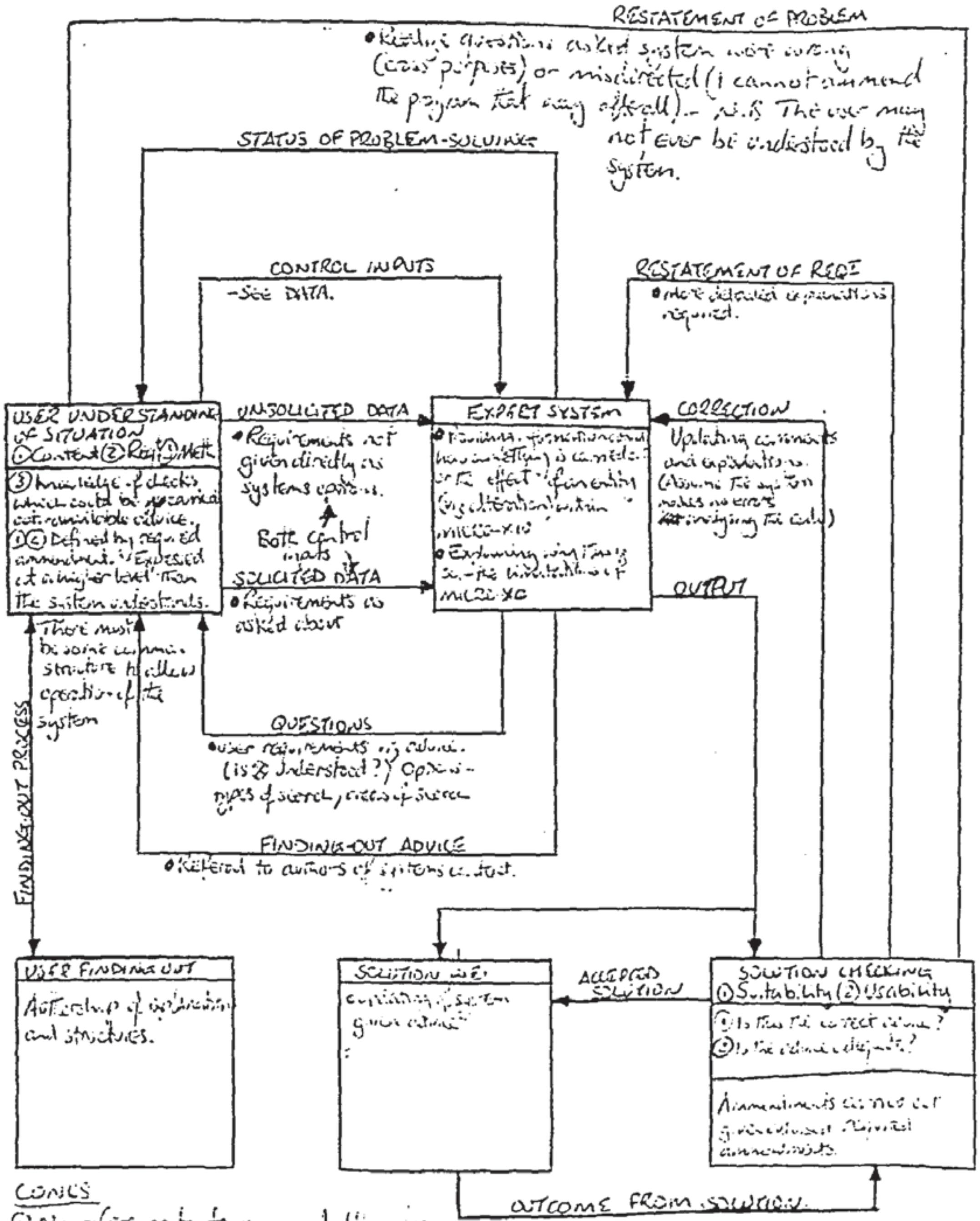
③ In terms of logic of rule base output can include – unused rules, unusable rules, new rules/attributes/values, programme structure (at different levels of detail), changes in exclusivity in categories or null groups (no attribute value), new levels in the "taxonomy" of attributes.

④ Actions can vary from nothing, to advice, to forced/unilateral action

⑤ The "taxonomy" of attributes' the exception in XIO can only be abstract but is, but so it can only make sense of it in terms of one ammendment at a time.

MODEL FIVE

PROVIDING UNDERSTANDING OF THE FUNCTION OF MICRO-XIC



CONICS

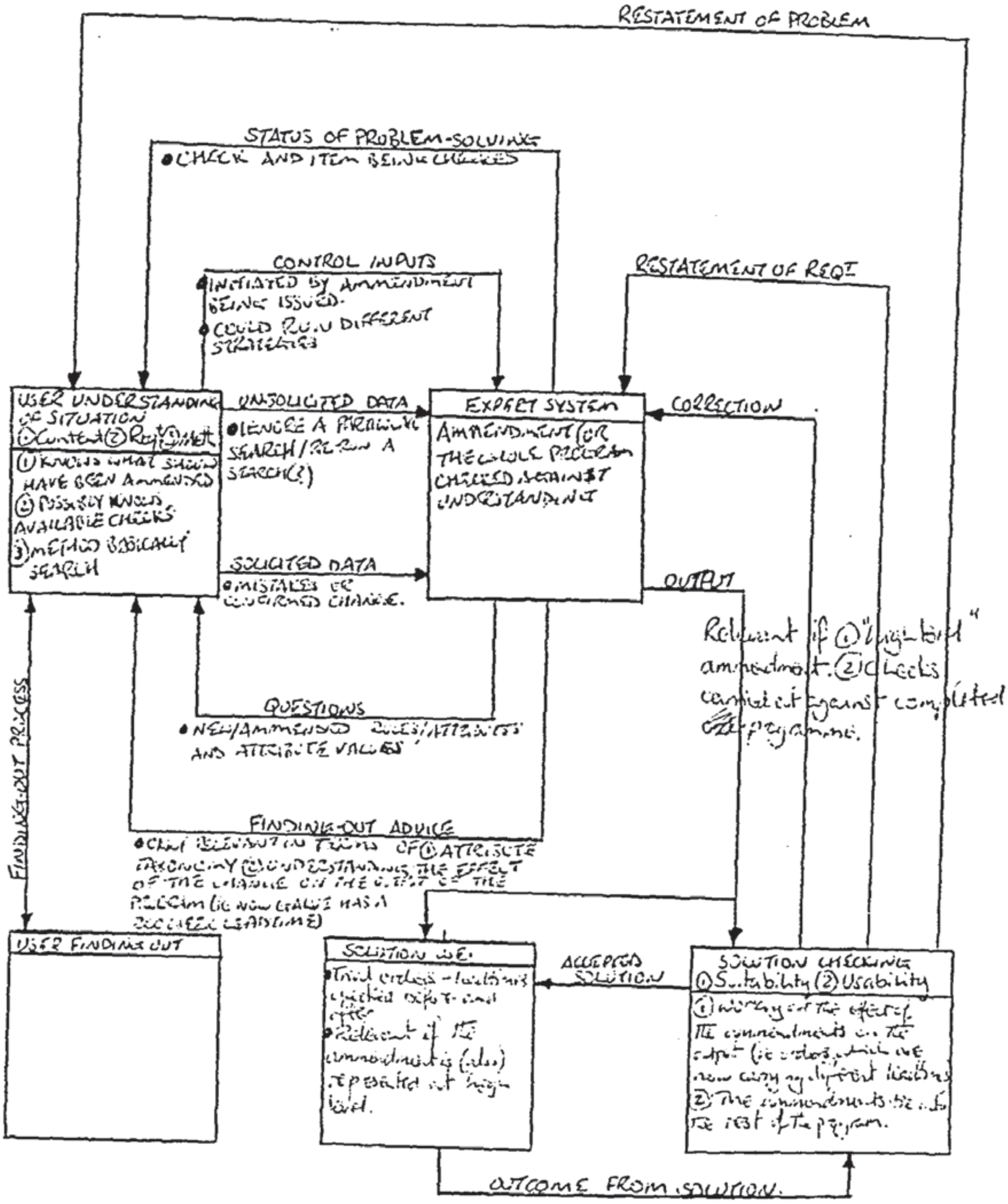
(1) No reference to forming of attributes.
But heavy reference to a common structure - ie the program structure and attributes & attribute values are possibly imputed to the user.

(2) The user, once he is capable of using this structure, is heavily in control of the system. Developing the structure (or available structures)

(3) The methods of providing understanding appear discrete, lists of attributes, uses of attributes, values of attributes, etc. — Certainly in terms of computing & providing a system.

MODEL SIX

CHECKING OF MICRO-XLC - THE OUTPUT BEING A CHECKED AMMENDMENT TO MICRO-XLC.



CONCS - The options available to expand/contract the boundary of the system are many - (confirming the IA).

- Possibility of checking resultant ammendment against trial orders (or monitoring actual use) suggests the composer may not understand but works at a higher level and engages in trial and error when ammending the programme - !